Control Engineering

Ashish Tewari

Automatic Control of Atmospheric and Space Flight Vehicles

Design and Analysis with MATLAB® and Simulink®

Birkhäuser

Control Engineering

Series Editor

Editorial Advisory Board

For further volumes:
http://www.springer.com/series/4988

Ashish Tewari

Automatic Control of Atmospheric and Space Flight Vehicles

Design and Analysis with MATLAB® and Simulink®

Birkhäuser

Ashish Tewari
Department of Aerospace Engineering
Indian Institute of Technology
Kanpur 208016, U.P.
India
ashtew@iitk.ac.in

ISBN 978-0-8176-4863-3 e-ISBN 978-0-8176-4864-0
DOI 10.1007/978-0-8176-4864-0
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011932916

Mathematics Subject Classification (2010): 70Qxx, 93Bxx

Printed on acid-free paper

www.birkhauser-science.com

To the Designer of the most exquisite control system

Preface

The main purpose of writing this book is to present a unified approach for automatic control of atmospheric and space flight vehicles. Such an outlook has become more necessary now than ever, with the advent of aerospace vehicles whose single mission covers operation as aircraft, rocket, and spacecraft at various instants. The automatic control system for such a craft must therefore have the entire gamut of flight control techniques in its repertoire.

This book is primarily designed as a textbook for senior undergraduates as well as graduate students in aerospace engineering/aeronautics and astronautics departments. As an acquaintance with control theory is not necessary, the book can be used as a first course in flight control systems. However, a familiarity with the basic mathematical concepts of calculus, linear algebra, and Laplace transform is required. The contents have evolved from the lecture notes of several 3rd–4th year undergraduate, and graduate-level courses I have taught in my career. The material in the book has been especially selected to be useful in modern courses on flight control systems, where the artificial distinctions among aircraft, rockets, and spacecraft are removed. Suggestions for the usage of the material by course instructors is given later. The chapters and sections are designed to follow in a sequence such that their concepts evolve logically, and are introduced in an easy-to-read manner, while retaining mathematical rigor.

It is easy for a reader to be lost in the theorems and proofs commonly found in textbooks on control systems, without realizing their practical significance. For this reason, the basic control theory concepts are covered in a manner accessible to a beginner in the topic. At the same time, understanding of relevant flight dynamic principles is highlighted while designing a flight control system. More emphasis is placed on realistic examples and exercises that require some programming, rather than those of the analytical type.

Each chapter begins with a list of clearly defined aims and objectives. At the end of each chapter, short summaries and a limited number of exercises are provided for consolidation. A primary feature of the book is the ready and extensive use of

MATLAB® and Simulink®.[1] The codes are supplied with the examples in order to readily illustrate control design and analysis. MATLAB/Simulink is standard, easy-to-use software that most engineering students learn in the first year of their curriculum. In giving the reader a hands-on experience with practical problems, the book is useful for a practicing engineer, apart from being an introductory text for the beginner. The use of the supplied MATLAB/Simulink codes as instructional tools (rather than as "black-boxes") is encouraged. The reader is required to write his/her own codes for solving many of the problems contained as exercises. All the codes used in the book are available for free downloading at the following web-site: http://home.iitk.ac.in/~ashtew.

The coverage of flight control topics in the book is basic rather than exhaustive, with a greater emphasis on single-variable control for a fundamental understanding of the relevant concepts. However, some advanced topics – such as linear, optimal control, nonlinear orbit plane control, and two-point boundary value problem solution for de-orbiting spacecraft – are included for imparting a flavor of the variety found in automatic flight control.

A reader is assumed to have taken basic undergraduate courses in mathematics and physics – particularly calculus, complex variables, linear algebra, and fundamental dynamics – and is encouraged to review these concepts at several places in the text. I will now briefly discuss the organization and highlights of the topics covered in each chapter in order to provide a ready guide to the reader and the classroom instructor.

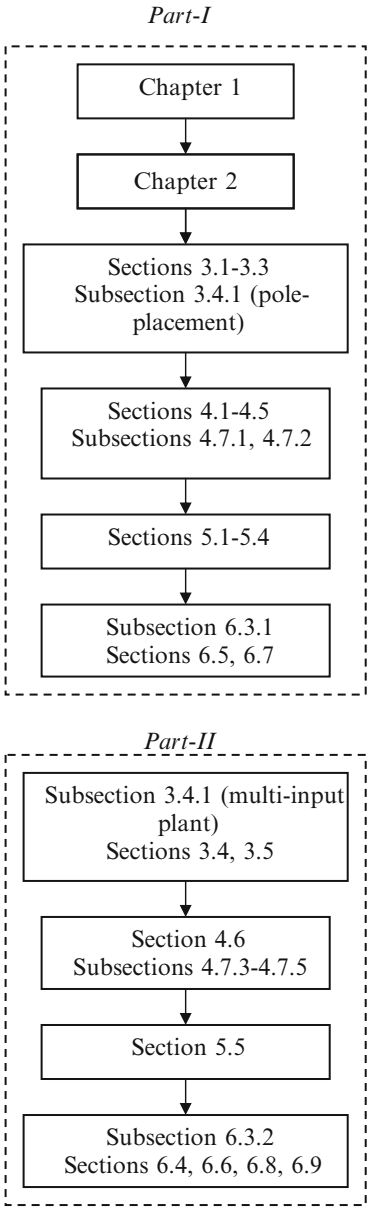
The undergraduate curriculum in an aerospace engineering department traditionally includes separate courses on introduction to automatic flight control of aircraft and spacecraft, depending upon the interest of students to pursue either atmospheric or space flight in further study and research. These basic courses are supplemented by advanced elective courses in the same respective areas. I think the time has come to move away from this tradition, and allow courses that cover aircraft, rockets, and spacecraft control systems at both introductory and advanced levels. It is envisaged that the contents of this book are followed in a single introductory course taught in the senior year. The fourth-year undergraduate courses, (a) 16.30 titled *Estimation and Control of Aerospace Systems*, taught by the Department of Aeronautics and Astronautics at Massachussetts Institute of Technology (MIT), and (b) AE-620 *Aircraft and Spacecraft Automatic Control Systems I*, of the Ohio State University (OSU) are likely examples. The following is a suggested coverage of material by course instructors.

The first part of the course would cover basics of control systems, with focus on single-variable control design for aircraft, rockets, and spacecraft. This part should cover material given in Chaps. 1 and 2, Sects. 3.1–3.3, 3.4.1 (only pole-placement), 4.1–4.5, 4.7.1, 4.7.2, 5.1–5.4, 6.3.1, 6.5, and 6.7, as shown in the accompanying

[1] MATLAB/Simulink® are registered products of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, U.S.A. http://www.mathworks.com.

flowchart. Such a range of topics includes basic stability and control of flight vehicles, thus also replacing the traditional aircraft/spacecraft stability and control courses (e.g., 16.333/16.335 of MIT and AE-720 of OSU).

The second (and shorter) part of the course would begin probably after the second midsemester examination, covering multivariable and nonlinear flight



control systems for aircraft, rockets, and spacecraft. This includes the design of regulators and observers by eigenstructure assignment and linear, optimal control (LQR) methods. The coverage can be selected from Sects. 3.4 (including 3.4.1 for multi-input plant), 3.5, 4.6, 4.7.3–4.7.5, 5.5, 6.3.2, 6.4, 6.6, 6.8, and 6.9 (see the accompanying flowchart). This material can also be used to supplement an advanced undergraduate course, such as (a) 16.31 *Feedback Control Systems* of MIT's Department of Aeronautics and Astronautics, and (b) AE-621 *Aircraft and Spacecraft Automatic Control Systems II* of Ohio State University.

I would like to thank the editorial and production staff of Birkhäuser Boston, especially Tom Grasso and Ben Cronin, for their constructive suggestions and valuable insights during the preparation of the manuscript. I am also grateful to The MathWorks, Inc. for providing the latest MATLAB/Simulink version, utilized in the examples throughout the book. This work would have been impossible without the unflinching support of my wife, Prachi, and frequent inputs from my daughter, Manya.

Indian Institute of Technology, Kanpur, India Ashish Tewari

Contents

Chapter 1
Introduction

1.1 Aims and Objectives

- To introduce control systems nomenclature and concepts
- To classify automatic control systems
- To briefly discuss linear systems theory

1.2 Control System

Control is the name given to the task of achieving a desired result. The object to be controlled (a flight vehicle) is referred to as *plant*, while the process that exercises the control is called the *controller*. Both the plant and the controller are *systems*, defined as self-contained sets of physical processes under study. A system is shown as a box (Fig. 1.1) connected by two arrows, one leading to, and the other away from, the box, called the *input* vector and the *output* vector, respectively. Such a representation of the system is called a *block-diagram*. The input and output vectors each consist of several scalar variables. Figure 1.1 shows a system with input vector, $\mathbf{u}(t)$, and output vector, $\mathbf{y}(t)$. In modeling a system, one must account for the relationship between the input and the output vectors. This relationship generally takes the form of a set of differential and algebraic equations if the system is governed by known physical laws. A system having known physical laws is said to be *deterministic*, whereas a system with unknown (or partially known) physical laws is called *nondeterministic*, or *stochastic*. Every system has certain unwanted external input variables – called *disturbance inputs* – that cannot be modeled physically and are thus treated as stochastic disturbances. The disturbances are generally of two types: (1) *process noise* that can arise either due to unwanted external inputs or due to uncertainty in modeling the system and (2) *measurement noise* that results from uncertainty in measuring the output vector.

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_1, © Springer Science+Business Media, LLC 2011

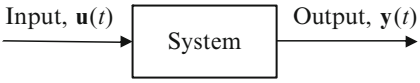


Fig. 1.1 Block-diagram of a system

A system consisting of the plant and the controller is called a *control system*. The controller exercises control over the plant through a *control input* vector, which is actually an input vector to the plant, but an output of the controller. In physical terms, this output can take the form of either a force or a torque (or both) applied to a flight vehicle. Often, only electrical (or mechanical) signals are generated by the controller through wires (or cables/hydraulic-lines), which must be converted into physical inputs for the plant by a separate subsystem called an *actuator*. Also, controllers generally require measurement of the output variables of the plant. Whenever a measurement of a variable is involved, it is necessary to model the dynamics of the measurement process as a separate subsystem called *sensor*. Generally, there are as many sensors and actuators as the number of measured scalar variables and the scalar control inputs, respectively. The sensors and actuators can be modeled as parts of either the plant or the controller. For our purposes, we shall model them as parts of the plant.

The condition, or *state*, of a system at a given time is specified by a set of scalar variables, called *state variables*, or in vector form, the *state vector*. The vector space spanned by the state vector is called a *state-space*, which can be visualized as a multidimensional space. The instantaneous state of the system is a point in the state space, whereas a plot of the state vector with time is referred to as the system's *trajectory*. The state of a system is defined as a collection of the smallest number of variables necessary to completely specify the system's evolution in time, in absence of external inputs. The number of state variables required to represent a system is called *order* of the system, because it is equal to the net order of differential equations governing the system. Thus, one can represent the system's behavior by a set of first-order differential equations in terms of the state variables, called *state equations*. A system's trajectory is then the solution of the state equations for given initial conditions and specified inputs. As the state variables can be either physically measurable quantities or abstract mathematical functions of time, one must have *output equations* that describe the dependence of the output variables on the state and input variables, in order to complete the system's physical description. (When both state and output equations are expressed in a vector format they are simply called the "state equation" and "output equation," respectively.)

While the size of the state-space (i.e., order of system) is unique, any given system can be described by infinitely many, alternative state-space representations. For example, a flight vehicle's state can be described by the position, $\mathbf{r}(t)$, velocity, $\mathbf{v}(t)$, angular velocity, $\boldsymbol{\omega}(t)$, and orientation, $\boldsymbol{\xi}(t)$, relative to a frame of reference. Thus, the state vector of a flight vehicle's motion is $\mathbf{x}(t) = \{\mathbf{r}(t), \mathbf{v}(t), \boldsymbol{\omega}(t), \boldsymbol{\xi}(t)\}^{\mathrm{T}}$. However, $\mathbf{x}(t)$ can be transformed into any number of different state vectors depending upon the choice of the reference frame.

Example 1.1. Consider a system with the following governing differential equations:

$$\frac{\mathrm{d}^2 y_1}{\mathrm{d}t^2} - 2\left(\frac{\mathrm{d}y_1}{\mathrm{d}t}\right)^3 - 3\sin(y_2) = u$$

$$\frac{\mathrm{d}y_2}{\mathrm{d}t} + 4y_2^2 = 0.$$

Here, the input applied to the system is $u(t)$ – a known function of time – while $y_1(t)$ and $y_2(t)$ are the measured output variables. Thus we have a single-input, two-output system. As the system is described by a second-order and a first-order differential equations, the order of the system is 3. Hence, we must select *three* state variables to describe the system's behavior. A possible choice of the state variables is the following:

$$x_1 = y_1, \qquad x_2 = \frac{\mathrm{d}y_1}{\mathrm{d}t}, \qquad x_3 = y_2,$$

which allows us to write the following state equations for the system:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = 2x_2^3 + 3\sin(x_3) + u$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = -4x_3^2,$$

or in a vector format, the following state equation:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, u),$$

where the state vector is $\mathbf{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ and

$$\mathbf{f}(\mathbf{x}, u) = \begin{Bmatrix} x_2 \\ 2x_2^3 - 3\sin(x_3) + u \\ -4x_3^2 \end{Bmatrix}.$$

The output equation expressed in terms of the state variables and the applied input is the following:

$$\mathbf{y} = \begin{Bmatrix} x_1 \\ x_3 \end{Bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}.$$

Notice that there is only an indirect dependence of the output vector on the applied input, $u(t)$, through the state equation.

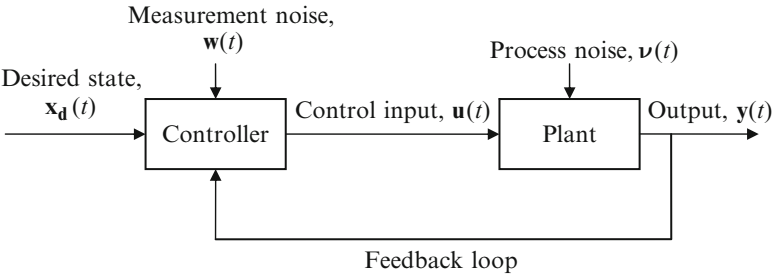


Fig. 1.2 A closed-loop control system

The most common task of a control system is bringing the plant to a desired state in the presence of disturbances, which can be achieved by either (a) an *open-loop* or (b) a *closed-loop* control system.

The controller of an open-loop control system works without any knowledge of the actual state of the plant at a given time, and the control is exercised based upon a model of the plant dynamics, as well as an estimate of its state at a previous time instant, called *initial condition*. Obviously, such a "blind" application of control can be successful in driving the plant to a desired state if and only if the plant model is exact, and the external disturbances are absent, which is seldom possible in practice. Therefore, a closed-loop control system is the more practical alternative, in which the actual state of the plant is provided to the controller through a *feedback loop* as shown in Fig. 1.2, so that the control input, $\mathbf{u}(t)$, can be appropriately adjusted to yield the desired state trajectory, $\mathbf{x_d}(t)$. In practice, the feedback consists of measurement of an output vector, $\mathbf{y}(t)$, through which an estimate of the plant's state can be obtained by the controller. Note the presence of unknown random inputs to the control system of Fig. 1.2 in the form of the process noise vector, $\boldsymbol{\nu}(t)$, and the measurement noise vector, $\mathbf{w}(t)$. If the feedback loop shown in Fig. 1.2 is removed, the control system becomes an open-loop type.

Example 1.2. Consider a feedback controller for the plant given in Example 1.1 with the following mathematical description of relationship between the two outputs and the input:

$$u = k_1 y_1 + k_2 \cos(y_2),$$

where k_1, k_2 are controller constants. The state equations of the overall closed-loop system are expressed as follows:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= x_2 \\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= 2x_2^3 + 3\sin(x_3) + k_1 x_1 + k_2 \cos(x_3) \\
\frac{\mathrm{d}x_3}{\mathrm{d}t} &= -4x_3^2,
\end{aligned}$$

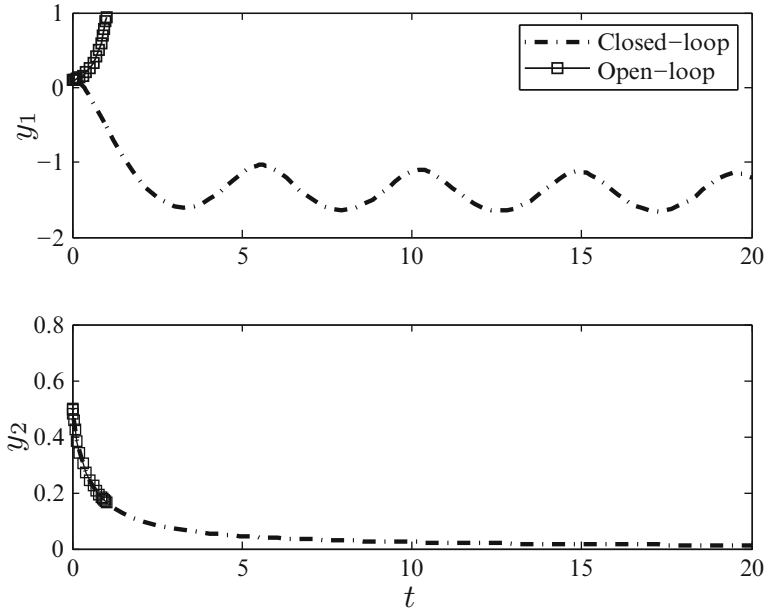


Fig. 1.3 Open-loop and closed-loop initial responses of the system of Example 1.2

If the system has an initial condition, $x_1(0) = 0.1, x_2(0) = -0.01, x_3(0) = 0.5$, then the response of the closed-loop system with $k_1 = -2, k_2 = -3$ is compared with that of the open-loop plant (i.e., $k_1 = k_2 = 0$) in Fig. 1.3, computed using MATLAB's *Runge-Kutta* solver [16, 19], *ode23.m*, by the statement

```
>>[t,x]=ode23('example1p2',[0 20],[0.1 -0.01 0.5]');
```

which calls the following function M-file, *example1p2.m*, that specifies the state equations of the closed-loop system:

```
function xdot=example1p2(t,x)
k1=-2;
k2=-3;
xdot(1,1)=x(2);
xdot(2,1)=2*x(2)^2+3*sin(x(3))+k1*x(1)+k2*cos(x(3));
xdot(3,1)=-4*x(3)^2;
```

Note that while the open-loop initial response, $y_1(t)$, continues to increase exponentially with time, the same output variable shows a bounded but oscillatory closed-loop behavior (in this case we have a nearly constant amplitude oscillation, called a *limit-cycle*). However, there is no change in the response, $y_2(t)$, when feedback control is applied. From this example, we can see that it is possible to modify a system's behavior by the use of feedback control. At the same time, the example shows that certain features of the original system (the open-loop plant) are unaffected by control.

1.3 Plant Model

Most plants are governed by nonlinear differential equations that can be expressed as a set of first-order, ordinary differential equations (state equations) such as the following:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\nu}(t), t], \tag{1.1}$$

where t denotes the time, $\mathbf{x}(t)$ the state vector of size $(n \times 1)$, $\mathbf{u}(t)$ is the control input vector of size $(m \times 1)$, and $\boldsymbol{\nu}(t)$ is the process noise vector of size $(p \times 1)$. The dimension n of the state vector is the order of the system. The nonlinear vector functional, $\mathbf{f}(.)$, is assumed to possess partial derivatives with respect to $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\boldsymbol{\nu}(t)$ in the neighborhood of $\mathbf{x_d}(t)$ that is a special solution of the state equation (called *nominal trajectory*). The nominal trajectory usually satisfies (1.1) for the unforced case, i.e., for $\mathbf{u}(t) = \mathbf{0},\ \boldsymbol{\nu}(t) = \mathbf{0}$:

$$\frac{\mathrm{d}\mathbf{x_d}}{\mathrm{d}t} = \mathbf{f}[\mathbf{x_d}(t), \mathbf{0}, \mathbf{0}, t], \quad t_\mathrm{i} \leq t \leq t_\mathrm{f}, \tag{1.2}$$

where $(t_\mathrm{i} \leq t \leq t_\mathrm{f})$ is called the *control interval* with initial time, t_i, and final time, t_f.

The output variables of a plant result from either direct or indirect measurements related to the state variables and control inputs through sensors. Certain errors due to sensor imperfections are invariably introduced in the measurement process leading to measurement noise. Therefore, the output vector, $\mathbf{y}(t)$ is related to the state vector, $\mathbf{x}(t)$, the control input vector, $\mathbf{u}(t)$, and the measurement noise vector, $\mathbf{w}(t)$, by an output equation given by:

$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t], \tag{1.3}$$

where $\mathbf{h}(.)$ is a vector functional and $\mathbf{w}(t)$ is generally of the same size as $\mathbf{y}(t)$.

We now take up an example to clarify the various concepts of system, state, and plant model.

Example 1.3. The level flight of an aircraft relative to a spherical, rotating Earth constitutes a fifth-order system with altitude, $h(t)$, airspeed, $v(t)$, azimuth, $\psi(t)$, latitude, $\delta(t)$, and longitude, $\lambda(t)$, as the state variables (Chap. 2). Our present objective is to derive a suitable plant that can be used for flying the aircraft over Earth's surface from an initial position, $(\delta_\mathrm{i}, \lambda_\mathrm{i})$, to a final position, $(\delta_\mathrm{f}, \lambda_\mathrm{f})$, while maintaining altitude and airspeed nearly constants in the presence of randomly changing wind speed and direction. Such a control system is called a *navigation* (or guidance) system. Control inputs to the level flight dynamics system are the thrust, $T(t)$, the lift, $\mathcal{L}(t)$, and the bank angle, $\sigma(t)$, measured from the local horizontal plane (local horizon). The inputs are applied through actuators either by a pilot or by an automatic control system (autopilot). Wind speed, $v_\mathrm{w}(t)$, and wind direction (azimuth), $\psi_\mathrm{w}(t)$, are stochastic (random) inputs, regarded as disturbances (process noise).

A constant altitude (i.e., level flight) is maintained by balancing the aircraft weight by the vertical component of the lift at all times. The lift is varied by rotating the aircraft about its pitch axis by the deflection, $\delta_E(t)$, of a control surface called elevator (Chap. 4). The variation of lift by pitching constitutes a second-order system, which increases the total order of the level flight system to 7. Furthermore, in order to have a nearly constant airspeed, the engine thrust must balance the aerodynamic drag, which is a function of the lift, and thus depends upon $\delta_E(t)$. The thrust is varied by changing the throttle valve angle, $\beta_T(t)$, of the engine, which is another second-order system. There are actuators called servomotors dedicated to rotating the elevator control surface by a required angle, $\delta_E(t)$, as well as for varying the throttle valve angle, $\beta_T(t)$. The elevator and throttle servomotors are second-order, feedback control systems driven by the autopilot through electrical signals, $u_1(t)$, and $u_2(t)$, respectively. Thus, the net order of the steady and level flight subsystem is 13.

The pure rolling (or banking) motion of the aircraft is a separate second-order system (Chap. 4) with the aileron deflection, $\delta_A(t)$, as the control input and bank angle, $\sigma(t)$, and roll rate, $P(t) = \dot{\sigma}$, as the state variables. An electrical signal, $u_3(t)$, is applied by the autopilot to an aileron servomotor – a second-order actuator – for changing the aileron angle by $\delta_A(t)$, which in turn, generates a torque in order to bank (or roll) the aircraft by angle $\sigma(t)$. However, when the ailerons are deflected, they also turn the aircraft slightly in the horizontal plane, but in the direction opposite to the desired turn (adverse yaw) (Chap. 4). In order to minimize drag, the adverse yaw must be avoided by coordinating the banking and turning (yawing) motions, which consists of pointing the fuselage in the desired flight direction. Therefore, control of a second-order yawing motion represented by the sideslip angle, $\beta(t)$, and yaw rate, $R(t) = \dot{\psi}(t) + \dot{\beta}(t)$, is necessary. The yawing motion is controlled by a control surface called rudder, whose deflection by angle $\delta_R(t)$ requires a rudder servomotor of order 2, and a driving electrical voltage, $u_4(t)$. Thus, the roll-yaw system is of order 8, thereby increasing the total aircraft plant order to 21.

The level flight dynamics is seen as the primary plant subsystem in Fig. 1.4, where Ω is the Earth's rotational rate. The velocity vector lies in the local horizon (the plane formed by North and East axes). The various actuator subsystems blocks representing the pitch, roll, and yaw dynamics, as well as elevator, throttle, aileron, and rudder actuators are shown in Fig. 1.5.

A likely choice of the state vector for the aircraft plant is the following:

$$\mathbf{x} = (h, v, \psi, \delta, \lambda, \alpha, \dot{\alpha}, T, \dot{T}, \delta_E, \dot{\delta}_E, \beta_T, \dot{\delta}_T, \sigma, P, \beta, R, \delta_A, \dot{\delta}_A, \delta_R, \dot{\delta}_R)^T, \qquad (1.4)$$

where overdot represents the time derivative. For a successful operation, the autopilot requires current knowledge of some state variables by either direct or indirect measurement. Hence, such variables constitute the output vector of the plant. While the current altitude, h, and airspeed, v, are measured by a pitot-static system, the aircraft's current geographic position, δ, λ, can be measured either by an inertial navigation system (INS) or a global positioning system (GPS). Furthermore, the angle-of-attack, α, and sideslip angle, β, can be measured by flow sensors, the

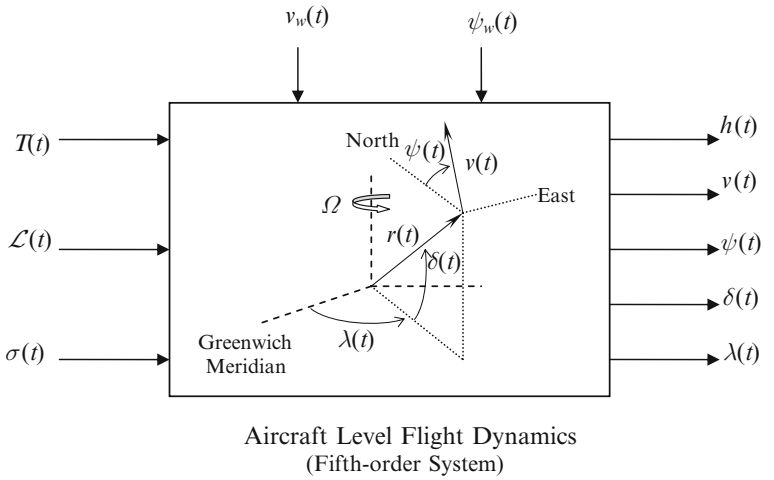


Fig. 1.4 Basic plant for aircraft's level flight

bank angle, σ, by a rate-integrating gyro, yaw rate, R, by a rate gyro (Chap. 2), and the control angles, $\delta_{\mathrm{E}}, \beta_{\mathrm{T}}, \delta_{\mathrm{A}}, \delta_{\mathrm{R}}$ picked-up by optical angle encoders. The likely output equation in this case is therefore the following:

$$\mathbf{y}(t) = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \end{pmatrix} \mathbf{x}(t) + \mathbf{w}(t), \qquad (1.5)$$

where $\mathbf{w}(t)$ is the stochastic measurement noise vector invariably associated with the sensors.

1.4 Properties of a System

A physical system is said to be *time-invariant* if its behavior does not change with time. Clearly, such a system is described by the same state and output equations at all times. If a system's behavior changes with time, the system is called a *time-varying*

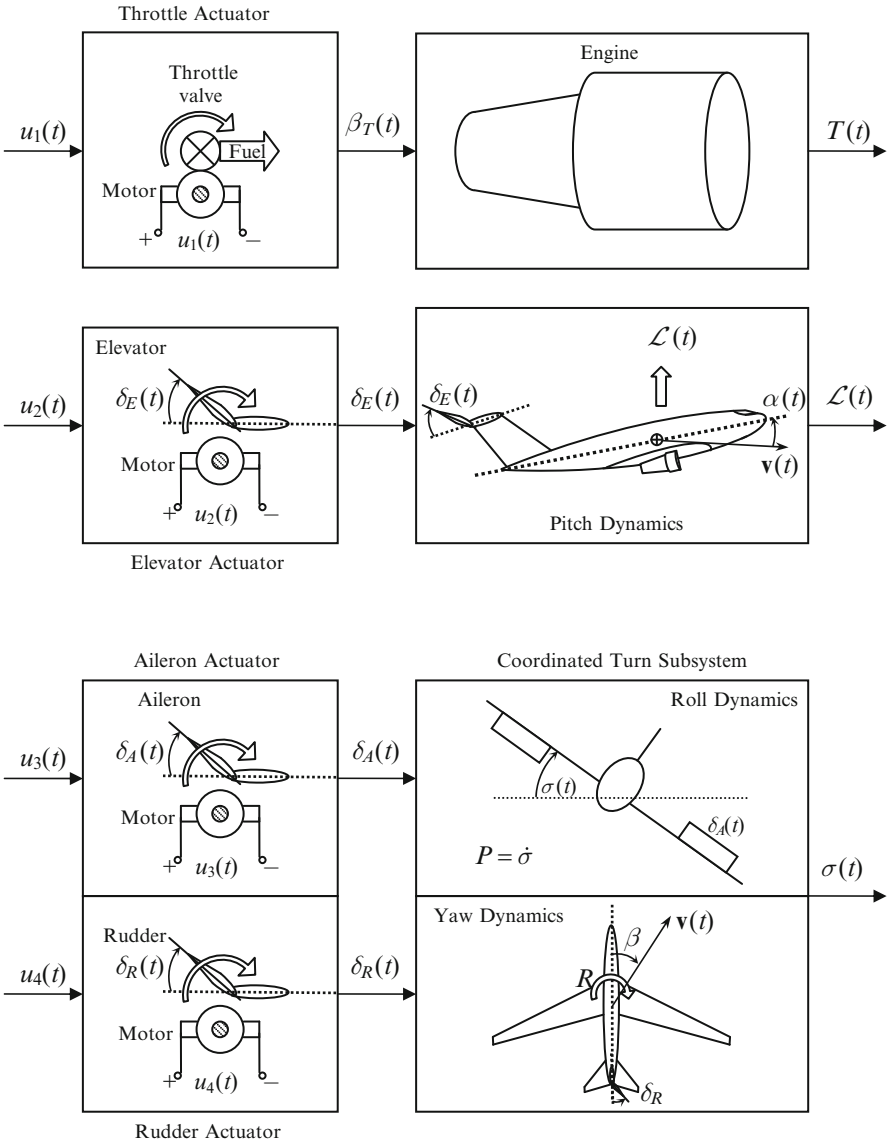


Fig. 1.5 Actuator subsystems for augmenting the level flight dynamics plant

system. An example of time-invariant systems is the rotational motion of an aircraft about its own center of mass, when flying at a constant speed and altitude. On the other hand, a rocket's flight is essentially a time-varying system due to a rapid change of mass and moments of inertia with time. For simplicity, we shall focus in this section on time-invariant systems. However, the concepts can be readily extended to time-varying systems, as explained later.

1.4.1 Stability in the Sense of Lyapunov

Every time-invariant system has certain states of rest, or *equilibria*, defined as the states that are unchanging with time. Such states are identified as fixed points in state-space, and are thus referred to as equilibrium points. By the very definition of an equilibrium point, a system remains in equilibrium forever unless displaced by an input. Clearly, all inputs of the system must vanish (i.e., we must have an *unforced* system) at equilibrium. An example of a system with multiple equilibria is a simple pendulum at its two vertical positions. However, by this very example we discover an important property of an equilibrium called *stability*, which is generally defined as the tendency to remain close to an equilibrium once displaced from it by a small disturbance. Of the two equilibria of a simple pendulum, one is clearly seen to be stable while the other is unstable by our everyday experience.

A practical method of analyzing the stability of an equilibrium state is by considering the initial response of the unforced system (i.e., $\mathbf{u} = \mathbf{0}$), beginning from an initial state very slightly displaced from the equilibrium at $t = 0$. Let the unforced state equation of a system be given by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1.6}$$

where $\mathbf{x}(t)$ is the state vector and $\mathbf{f}(.)$ is a vector functional assumed to possess continuous derivatives with respect to state variables at a given equilibrium point, $\mathbf{x}_e$, which must satisfy

$$\dot{\mathbf{x}}_e = \mathbf{f}(\mathbf{x}_e) = \mathbf{0}. \tag{1.7}$$

By defining the deviation (or perturbation) of the system's state from equilibrium as:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_e, \tag{1.8}$$

we write (1.6) as follows:

$$\dot{\mathbf{e}} = \mathbf{f}(\mathbf{e} + \mathbf{x}_e). \tag{1.9}$$

The term on the right-hand side of (1.9) is generally expanded in an infinite power series of the perturbation, $\mathbf{g}(\mathbf{e})$, from equilibrium,

$$\mathbf{f}(\mathbf{e} + \mathbf{x}_e) = \mathbf{f}(\mathbf{x}_e) + \mathbf{g}(\mathbf{e}) = \mathbf{g}(\mathbf{e}), \tag{1.10}$$

thereby enabling us to study the behavior of the system in the vicinity of the equilibrium point, $\mathbf{x}_e$.

There are many concepts of stability of an equilibrium point, all requiring a bounded response to a bounded perturbation. However, the precise definition of stability depends upon the system's behavior in the vicinity of the equilibrium point. The most widely accepted definition of stability is that in the sense of Lyapunov, according to which an equilibrium point is said to be stable if it is possible to make

the system's state, $\mathbf{x}(t)$, always remain arbitrarily close to the equilibrium point, $\mathbf{x}_e$, by starting sufficiently close to it. A mathematical statement of stability in the sense of Lyapunov is given as follows:

A system described by (1.6) and (1.7) is said to be stable about the equilibrium point, $\mathbf{x}_e$, in the sense of Lyapunov if for each real and positive number, ϵ, however small, there exists another real and positive number, δ, such that

$$|| \mathbf{e}(0) ||< \delta \tag{1.11}$$

implies that

$$|| \mathbf{e}(t) ||< \epsilon \; ; \qquad t \geq 0, \tag{1.12}$$

where

$$|| \mathbf{e} ||= \sqrt{\sum_{i}^{n} e_i^2} \tag{1.13}$$

denotes the *Euclidean norm* of $\mathbf{e} = (e_1, e_2, \ldots, e_n)^{\mathrm{T}}$ for an nth order system.

Example 1.4. For the closed-loop control system of Example 1.2 with $k_1 = -2$, $k_2 = -3$, it was observed that while $x_3(t)$ tends to zero at large times, the state variables, $x_1(t)$ and $x_2(t)$, stay away from the equilibrium point, $\mathbf{x}_e = \mathbf{0}$, when an initial perturbation is applied at $t = 0$. This behavior is seen to persist even when the initial perturbation is decreased to an arbitrary small value, such as $x_1(0) = 0.001, x_2(0) = -0.001, x_3(0) = 0.005$, as shown in Fig. 1.6. The departure of the system from equilibrium is clearly seen in the plot of $x_2(= \dot{x}_1)$ vs. x_1 (Fig. 1.7) called the *phase-plane plot* of the subsystem, where the response is attracted to a closed curve depicting a nearly constant amplitude oscillation, called a *limit-cycle*. As even a small perturbation makes the system depart from the vicinity of the equilibrium point, $\mathbf{x}_e = \mathbf{0}$, it is unstable in the sense of Lyapunov.

In Example 1.4, we are fortunate enough to have decoupled the system into separate subsystems by a proper choice of state variables, which is not always feasible. It is to be further noted from Example 1.4 that if a system has an unstable subsystem at equilibrium then the given equilibrium is unstable. Conversely, if all the subsystems of a system are stable then the equilibrium is stable.

1.4.2 Controllability and Observability

A system's *controllability* is an essential property by which it is possible to change the system's behavior by the application of control inputs. Consider a time-invariant system governed by the state equation,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{1.14}$$

where $\mathbf{u}(t)$ is the input vector. If it is possible to change the system's state from *any* arbitrary initial value, $\mathbf{x}(0)$, to *any* final value, $\mathbf{x}(t_f)$, in a *finite* time, t_f, solely

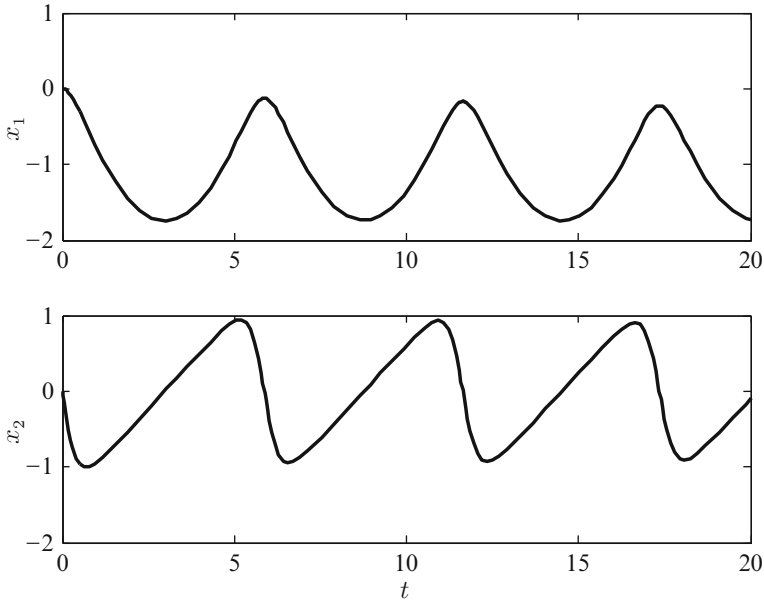


Fig. 1.6 Closed-loop initial response of a subsystem of Example 1.2 with $k_1 = -2, k_2 = -3$ and $x_1(0) = 0.001, x_2(0) = -0.001, x_3(0) = 0.005$

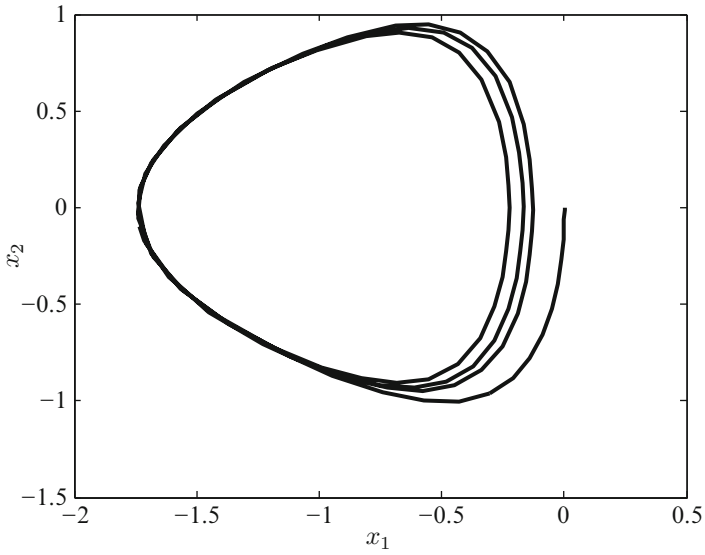


Fig. 1.7 Phase-plane plot, $x_2(= \dot{x}_1)$ vs. x_1 (Example 1.4) with $k_1 = -2, k_2 = -3$ and $x_1(0) = 0.001, x_2(0) = -0.001, x_3(0) = 0.005$

by the application of the control inputs then the system is said to be *controllable*. It is important to emphasize the words *any* and *finite*, because the ability to move the system among some specific states in a finite time by applying control inputs does not imply controllability. Similarly, requiring an infinite time for the change among arbitrary states implies an uncontrollable system.

For a system to be controllable, all of its state variables must be affected – either directly or indirectly – by the input variables. If one can identify an uncontrollable subsystem then the entire system is uncontrollable. In order to have controllability of a system, all the subsystems must be controllable.

Example 1.5. Let us reconsider the open-loop system of Example 1.1. We see that while both $x_1(t)$ and $x_2(t)$ are changed by applying the input $u(t)$, it is impossible to change the state $x_3(t)$ by any application of the input, however large. Thus we say that the subsystem represented by the third state equation,

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = -4x_3^2,$$

is uncontrollable, while the other subsystem described by $x_1(t)$ and $x_2(t)$ is controllable. We further note that as the uncontrollable subsystem is inherently stable about the origin, we need not worry about controlling it, if our only objective is to stabilize the complete system.

Example 1.5 illustrates the fact that uncontrollability would not pose a problem in stabilizing an unstable system through closed-loop control, as long as the uncontrollable subsystems are stable about the given equilibrium. This brings us to a feature called *stabilizability*, defined as the property by which an uncontrollable and unstable system can be stabilized solely by the application of control inputs. Clearly, stabilizability requires that all the uncontrollable subsystems must be stable.

Identification of a system's behavior through the measurement of output variables is enabled through another important property called *observability*. Observability is the property of an unforced system, (1.6), with the output equation,

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \tag{1.15}$$

by which it is possible to estimate *any* initial state, $\mathbf{x}(0)$, solely from a *finite* record, $t \geq 0$, of the output vector, $\mathbf{y}(t)$. For a system to be observable, all of its state variables must contribute, either directly or indirectly, to the output vector. If there is a subsystem that leaves the output vector unaffected then the entire system is unobservable. If it is possible to decompose an unobservable system into observable and unobservable subsystems then a system whose unobservability is caused by a stable subsystem at a given equilibrium is said to be *detectable*. Clearly, an undetectable system is the one which has at least one subsystem that is both unobservable and unstable at a given equilibrium.

Example 1.6. The system of Example 1.1 has the following output equations:

$$y_1 = x_1; \qquad y_2 = x_3.$$

The measurement of the two outputs provides us direct information about the state variables, $x_1(t)$ and $x_3(t)$. However, the remaining state variable, $x_2(t)$, contributes indirectly to the output vector through the state equations. Therefore, we have an observable system.

Controllability and observability are easier to analyze for linear systems, as we shall see later.

1.5 Automatic Controllers

A controller that performs its duty without human intervention is termed *automatic controller*, and obeys well defined mathematical relationships between plant state variables and control inputs, called *control laws*. An automatic controller is generally designed to fulfill a given objective based upon changing the dynamic characteristics of the plant for a given application. For example, in certain cases the nonlinear state equations of a plant can be linearized by using a particular control law called *feedback linearization*. Another typical control law design aims at stabilizing an unstable plant (or for improving the stability characteristics of a marginally stable plant) by a method called *stability augmentation*. A common objective in most control law design methodologies is to achieve a given control task with the minimization of a specific function of control and state variables. Such a design approach is termed *optimal control*.

Example 1.7. Recall from Example 1.2 that the open-loop plant had an exponentially increasing output, $y_1(t)$, for a given initial condition, $x_1(0), x_2(0), x_3(0)$. It can be shown that for the given plant, such a response would arise for any nonzero initial condition, however small in magnitude. As we have seen, this behavior is called *instability*, because the slightest disturbance applied to the plant at its rest (or equilibrium) condition of $x_1(0) = x_2(0) = x_3(0) = 0$ tends to take it away from the equilibrium. While the feedback control law of Example 1.2 produced a bounded response, it failed to restore the system to its original equilibrium (Example 1.4). In order to make the plant ultimately reattain its equilibrium state, consider the following nonlinear feedback control law:

$$u = -k_1 x_1 - k_2 x_2 - 2x_2^3 - 3\sin(x_3),$$

where k_1, k_2 are constant parameters. The state equations of the overall closed-loop system become:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2$$

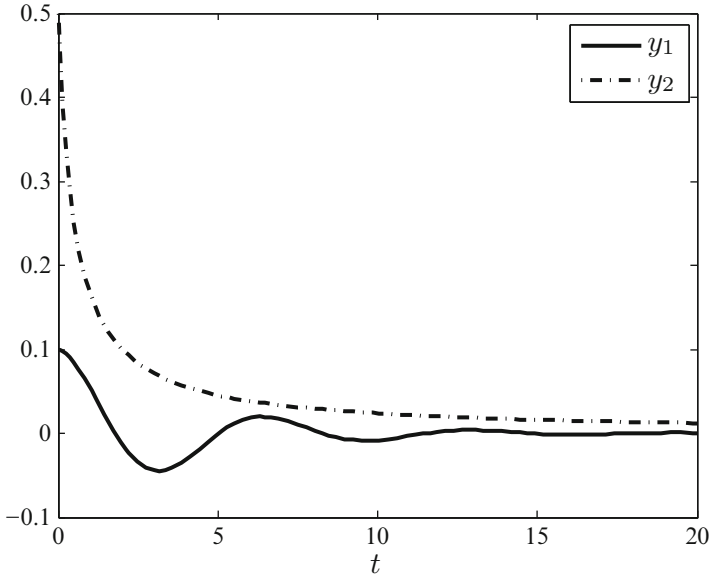


Fig. 1.8 Closed-loop initial response of the control system of Example 1.7

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = -k_1 x_1 - k_2 x_2$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = -4x_3^2.$$

By choosing such a control law, the nonlinear terms involving powers or transcendental functions of the state variables in the first two state equations have now vanished. This is termed *feedback linearization* of the concerned state equations. Furthermore, by a proper choice of the controller constants, k_1, k_2, one can make the closed-loop system's initial response, $y_1(t)$, tend to zero at large times. As the response of the second output, $y_2(t)$, while being unaffected by control, tends to zero on its own, we can see that the closed-loop system will now return to its original state of rest, whatever initial condition is applied at $t = 0$. However, the control law presented here is more sophisticated than that of Example 1.2 because it requires not only the measurement of the two outputs but also taking the time derivative of $y_1(t)$ in order to generate $x_2(t) = \mathrm{d}y_1/\mathrm{d}t$ for feedback. The closed-loop response for initial condition, $x_1(0) = 0.1, x_2(0) = -0.01, x_3(0) = 0.5$, with $k_1 = 1, k_2 = 0.5$ is plotted in Fig. 1.8.

Example 1.8. One of the simplest automatic control systems from implementation viewpoint is based on an *switching* (or *on–off*) controller that alternately applies either the maximum possible control input magnitude in the desired direction or a

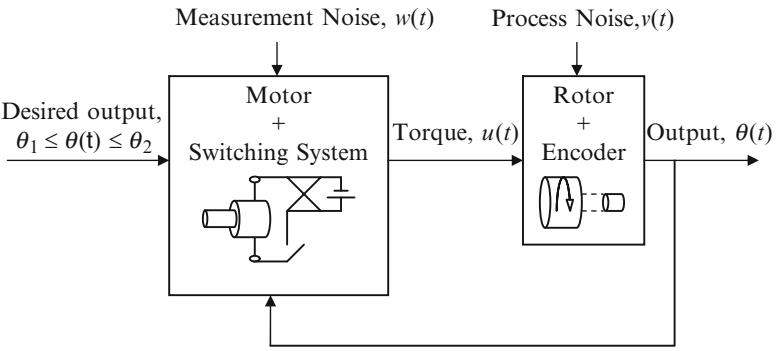


Fig. 1.9 Switching control system for a rotor

zero control input. Consider a rotor of moment of inertia, J, driven by a DC motor that can apply a maximum torque, u_{m}, in either direction. The equation of motion of the rotor is given by:

$$J\ddot{\theta} = u,$$

where $\theta(t)$ is the angular position of the rotor (measurable by a optical encoder) and $u(t)$, the torque input. A state-space representation of the plant (rotor plus encoder) is the following:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_1 &= \frac{1}{J}u\\ y_1 &= x_1 = \theta.\end{aligned}$$

It is desired to maintain the rotor between the limits $\theta_1 \leq \theta \leq \theta_2$ in the presence of disturbances. For this purpose, consider a switching controller that drives the motor according to the following control law:

$$u(t) = \begin{cases} u_{\mathrm{m}} & (\theta < \theta_1)\\ 0 & (\theta_1 \leq \theta \leq \theta_2)\\ -u_{\mathrm{m}} & (\theta > \theta_2)\end{cases}.$$

Thus the motor is a part of the controller and is driven by a switch that can either cut-off the constant input voltage or reverse its polarity according to the sensed position of the rotor, which must be maintaned between the limits, θ_1 and θ_2. A block-diagram of the control system is shown in Fig. 1.9. A numerical example for $J = 1{,}000$ kg m^2, $u_{\mathrm{m}} = 10$ N.m, $\theta_1 = 9.5°$, and $\theta_2 = 10.5°$ with initial condition, $\theta(0) = 9°$ and $\dot{\theta}(0) = 0$, is computed by the fourth-order MATLAB Runge-Kutta solver, *ode45.m*:

```
>>[t,x]=ode45('example1p8',[0 500],[9*pi/180 0]');
```

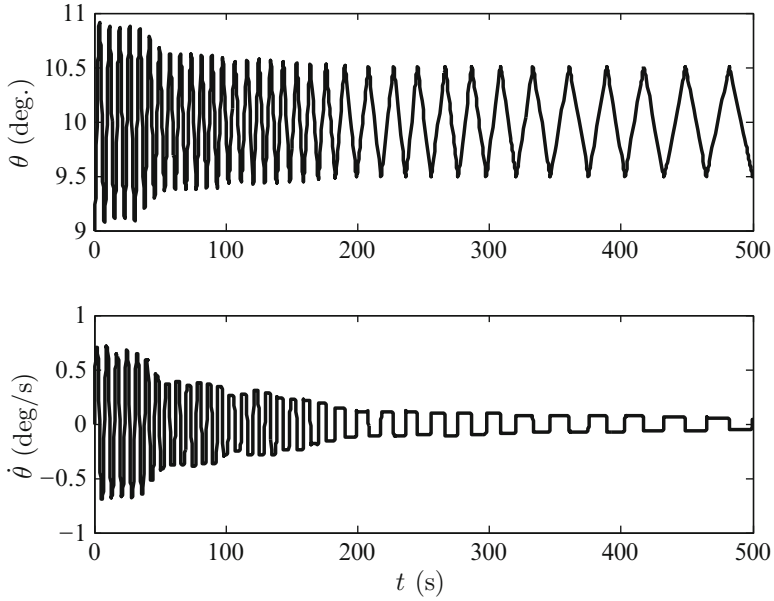


Fig. 1.10 Closed-loop initial response of the switching control system of Example 1.8

where *example1p8.m* is the following function M-file that specifies the state equations of the closed-loop system:

```
function xdot=example1p8(t,x)
J=1000;
um=10;
dtr=pi/180;
xdot(1,1)=x(2);
if x(1)<9.5*dtr
    u=um;
elseif x(1)>10.5*dtr
    u=-um;
else
    u=0;
end
xdot(2,1)=u/J;
```

The results of the closed-loop simulation are plotted in Figs. 1.10 and 1.11. Note that despite the initial error, the rotor is ultimately brought within the desired angular range $9.5 \leq \theta \leq 10.5°$ by a repeated switching action. The frequency of switching and the angular speed, $|\dot{\theta}|$, both decline as the system is brought closer to the desired angular band. As full drive is applied in case of even the slightest error, the control system has an excellent performance. A similar switching controller is commonly used for attitude control of spacecraft and rockets with on–off reaction jet thrusters. However, by virtue of switching, the control system is essentially a nonlinear one, although we have a linear plant dynamics.

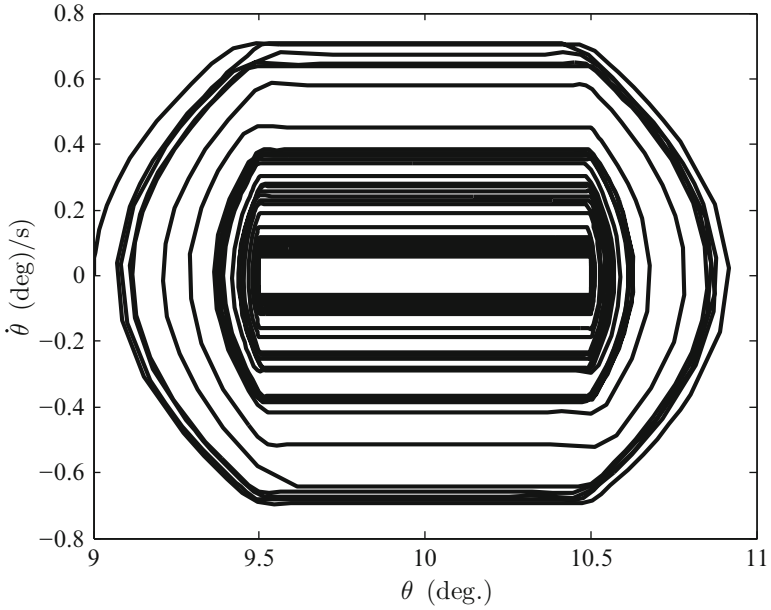


Fig. 1.11 Phase-plane plot, $\dot{\theta}$ vs. θ, of the switching control system of Example 1.8

Example 1.9. Consider an aerodynamic missile to be controlled in roll by moving a set of fins arrayed about the axis of symmetry, such that a deflection of all the fins by the same angle exerts a rolling moment (Chap. 4). The equation of roll dynamics is the following:

$$\ddot{\phi} + \dot{\phi} = 100\delta,$$

where $\phi(t)$ is the roll (or bank) angle (output) of the missile that can be measured by a rate-integrating gyro (Chap. 2) and $\delta(t)$ is the fin deflection angle (input). A simple feedback mechanism is devised such that a current proportional to the error between the desired and the actual bank angles, $e(t) = \phi_d - \phi(t)$, is used to drive a fin actuator (servo motor) with the following dynamics:

$$0.05\dot{\delta} + \delta = \delta_c,$$

where $\delta_c(t)$ is the fin deflection commanded by the feedback controller. The constant gain, feedback controller works according to the control law,

$$\delta_c = 0.1(\phi_d - \phi).$$

The fin deflection is physically limited by $|\delta(t)| \leq 5.7°$, called *saturation* limit. Thus, while the plant and the feedback controller are linear, the actuating mechanism is nonlinear. A Simulink block diagram of the overall control system for a step desired bank angle, $\phi_d = 57°$ is shown in Fig. 1.12, where the missile and fin actuator blocks

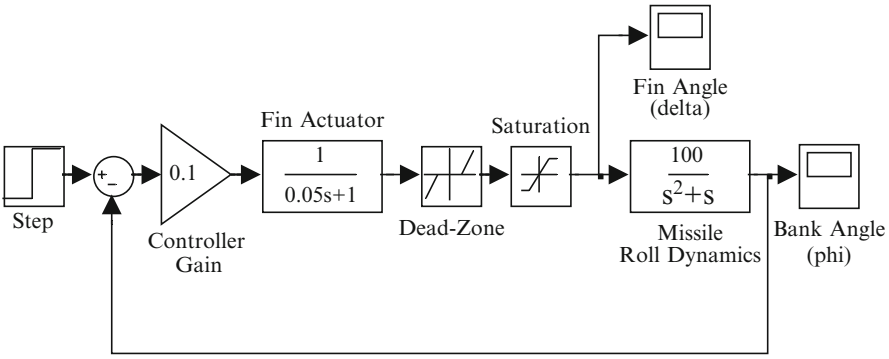


Fig. 1.12 Simulink block-diagram for the missile roll control system of Example 1.9

are represented by their respective *transfer functions*, i.e., ratios of the Laplace transforms of the respective output and input (Chap. 3). Note the saturation block at the input of missile dynamics. Furthermore, there is the option of adding a *dead-zone* between the actuator and the missile dynamics as shown in Fig. 1.12. The task of the dead-zone is to inhibit fin deflection unless the commanded fin deflection, δ_c, crosses a certain threshold magnitude, $\pm a$. The dead-zone is mathematically represented by:

$$\delta = \begin{cases} 0 & (|\,\delta_c\,| \leq a) \\ \delta_c & (|\,\delta_c\,| > a) \end{cases}$$

A small amount of dead-zone is inherent in all mechanical systems due to static friction (or stiction). However, in this case, dead-zone is deliberately added in order to improve closed-loop behavior. The presence of a dead-zone adds to the nonlinearity of the actuator.

Simulation is carried out for 10 s, both with and without a dead-zone of $a = 2.85°$, and the results are compared in Fig. 1.13. Note the oscillating response about the desired bank angle of 57° without the dead-zone, which persists for more than 10 s. Such an undesirable behavior of a closed-loop system is called *hunting*. On the other hand, the addition of the dead-zone results in a much smaller fin activity as well as a nonoscillatory response that quickly settles down to a steady state. This is due to a reduced sensitivity of the control input on the bank angle error, $e(t) = \phi_d - \phi(t)$. However, the steady state of the system with dead-zone is a little removed from the desired bank angle (an error of about 20%). By suitably adjusting the threshold of the dead-zone, a, a trade-off between the steady state error and the transient behavior can be achieved.

Examples 1.7–1.9 illustrate two different types of automatic control: a nonlinear plant controlled by a special control law that renders the closed-loop system linear, and a linear plant driven by either a feedback switching controller or a nonlinear

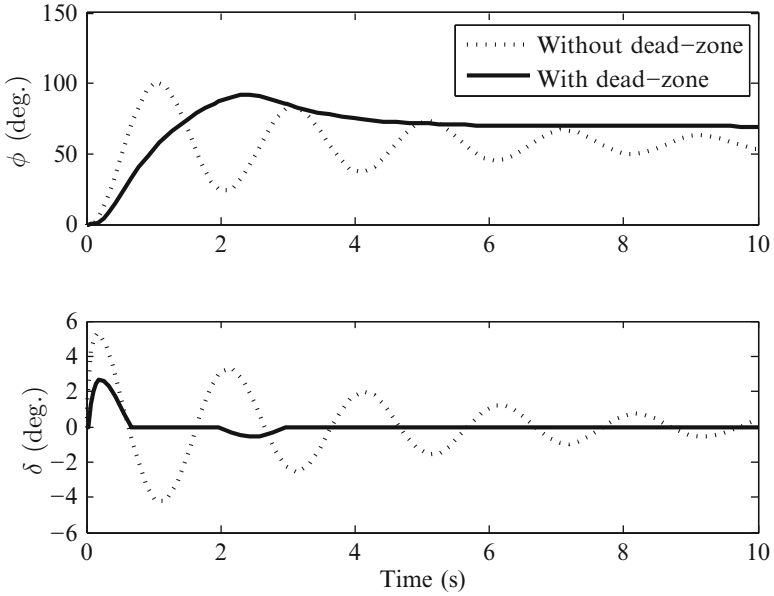


Fig. 1.13 Closed-loop step response of the missile roll control system of Example 1.9

actuator, resulting in a nonlinear control system. However, all the three examples deal with achieving a constant desired state at sufficiently large but unspecified time – an objective termed as *regulation* of the system. We shall now consider more general control applications, namely those that either involve tracking a changing reference state or achieving a given constant state in a specified time.

1.5.1 Terminal and Tracking Control

There are two broad categories of automatic control systems: (a) *Terminal Control*, and (b) *Tracking Control*. A terminal control system aims at changing the plant's state from an initial state, $\mathbf{x}(t_\mathrm{i})$, to a *terminal* (or final) state, $\mathbf{x}(t_\mathrm{f})$, in a specified time, t_f, by applying a control input, $\mathbf{u}(t)$, in the fixed control interval, $(t_\mathrm{i} \leq t \leq t_\mathrm{f})$. Examples of terminal control include guidance of spacecraft and rockets.

Example 1.10. Consider a simple example of rotating a rigid spacecraft by the application of an angular acceleration input, $u(t)$, in the desired direction. The equation of motion is similar to that of Example 1.8 and is expressed as follows:

$$\ddot{\theta} = u$$

The nature of the actuator is such that the angular acceleration increases linearly with time until achieving a maximum value, and then remains constant until the final time, t_f, when the control is turned off:

$$u(t) = \begin{cases} \dfrac{u_\mathrm{m}}{a} t & (0 \le t < a) \\ u_\mathrm{m} & (a \le t \le t_\mathrm{f}) \\ 0 & (t > t_\mathrm{f}) \end{cases}.$$

Here, the positive constant, a, depends upon the final time, $t_\mathrm{f} \ge a$, and is computed from the requirement of achieving either a given angle, θ_f, or a given angular rate, $\dot{\theta}_f$, at $t = t_\mathrm{f}$, beginning from a given initial condition, $\theta(0), \dot{\theta}(0)$.

Given the variation of the input acceleration, we can directly integrate the equation of motion for $t_\mathrm{f} \ge a$ to obtain the following solution:

$$\theta(t) = \begin{cases} \theta(0) + \dot{\theta}(0)t + \dfrac{u_\mathrm{m}}{6a} t^3 & (0 \le t < a) \\ \theta(0) + \dot{\theta}(0)t + u_\mathrm{m} \dfrac{a^2}{6} + \dfrac{u_\mathrm{m}}{2} t(t-a) & (a \le t \le t_\mathrm{f}) \\ \theta(0) + \dot{\theta}(0)t_\mathrm{f} + u_\mathrm{m} \dfrac{a^2}{6} + \dfrac{u_\mathrm{m}}{2} t_\mathrm{f}(t_\mathrm{f} - a) + u_\mathrm{m} \left(t_\mathrm{f} - \dfrac{a}{2} \right) (t - t_\mathrm{f}) & (t > t_\mathrm{f}) \end{cases}.$$

Let us take the case of $u_\mathrm{m} = 0.01$ rad/s^2, $\theta_f = 180°$, $t_\mathrm{f} = 30$ s, and $\theta(0) = 0, \dot{\theta}(0) = 0$. Enforcement of the terminal boundary condition, $\theta(t_\mathrm{f}) = \theta_f$, produces

$$a = \frac{3}{2} t_\mathrm{f} - \frac{1}{2} \sqrt{\frac{24\theta_f}{u_\mathrm{m}} - 3t_\mathrm{f}^2} = 10.2156 \text{ s}.$$

The control history and the nominal trajectory are plotted in Fig. 1.14. Note that a different value of a (and thus a different trajectory) would result from enforcing a final angular rate condition.

The objective of a tracking control system is to maintain the plant's state, $\mathbf{x}(t)$, close to a nominal, reference state, $\mathbf{x_d}(t)$, which is available as a solution to the unforced plant state equation, (1.2), by the application of the control input, $\mathbf{u}(t)$. Most of the flight control problems – such as aircraft guidance, orbital control of spacecraft, and attitude control of all aerospace vehicles – fall in this category.

While the terminal controller design is typically based on a nonlinear plant (1.1) and often involves iterative solution of a *two-point boundary value problem*, the design of a tracking controller can be carried out by linearizing the plant about the nominal trajectory, $\mathbf{x_d}(t)$. The tracking controller continually compares the plant's state, $\mathbf{x}(t)$, with the nominal (desired) state, $\mathbf{x_d}(t)$, and generates a control signal that depends upon the error vector,

$$\mathbf{e}(t) = \mathbf{x_d}(t) - \mathbf{x}(t). \tag{1.16}$$

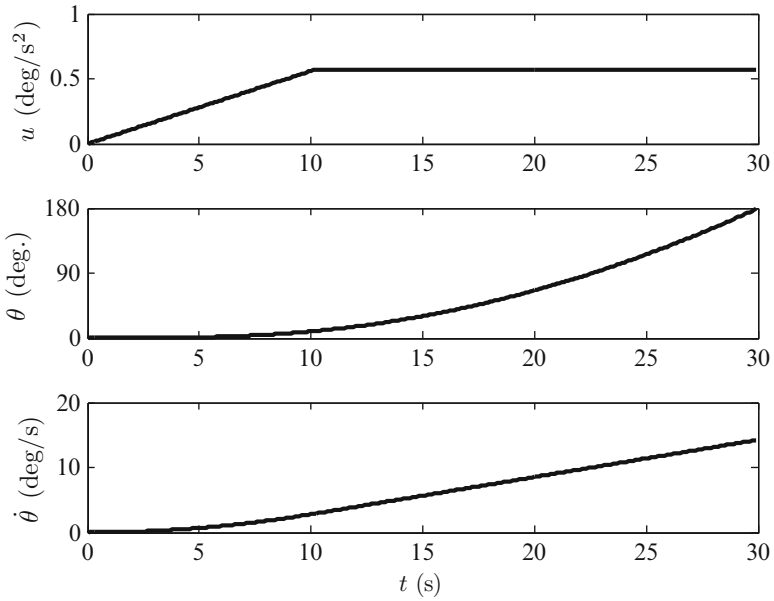


Fig. 1.14 Trajectory and control history for the terminal control problem of Example 1.10

A tracking control system can be further classified into *state feedback* and *output feedback* systems. While a state feedback system involves measurement and feedback of all the state variables of the plant (which is rarely practical), an output feedback system is based upon measurement and feedback of some output variables that form the plant's output vector, $\mathbf{y}(t)$. Clearly, the controller must be able to estimate the plant's state from the measured outputs and any control input vector, $\mathbf{u}(t)$, it applies to the plant. The estimation of plant's state vector from the available outputs and applied inputs is called *observation* (or *state estimation*), and the part of the controller that performs this essential task is called an *observer*. As the observation is seldom exact, one only has the state estimate, $\mathbf{x_o}(t)$, in lieu of the actual state, $\mathbf{x}(t)$. The state equation of the observer can be expressed as follows:

$$\dot{\mathbf{x}}_\mathbf{o} = \mathbf{f_o}(\mathbf{x_o}, \mathbf{y}, \mathbf{u}) \tag{1.17}$$

Apart from the observer, the controller has a separate subsystem called *regulator* for driving the tracking error vector, $\mathbf{e}(t)$, to zero over a reasonable time interval. The regulator is the heart of the tracking control system and generates a control input based upon the detected error,

$$\mathbf{e_o}(t) = \mathbf{x_d}(t) - \mathbf{x_o}(t). \tag{1.18}$$

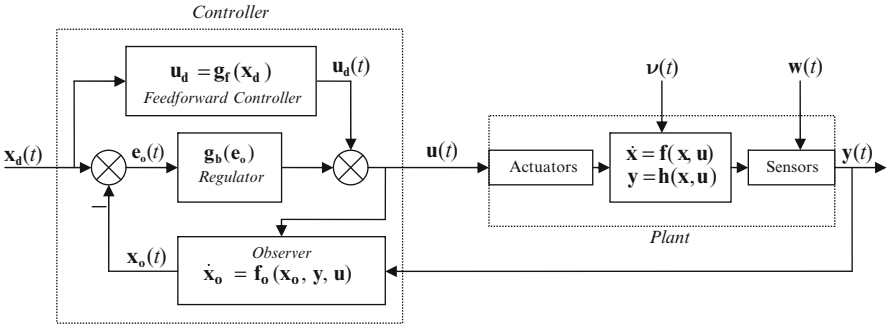


Fig. 1.15 Tracking control system with an observer

Hence, the control input, $\mathbf{u}(t)$, must depend upon the detected error vector, $\mathbf{e_o}(t)$. Moreover, $\mathbf{u}(t)$, may also depend explicitly upon the nominal, reference state, $\mathbf{x_d}(t)$, which must be fed forward in order to contribute to the total control input. Therefore, there must be a third subsystem of the tracking controller, called a *feedforward controller*, which generates part of the control input, $\mathbf{u_d}(t)$, depending upon the desired state.

A schematic block-diagram of the tracking control system with an observer is shown in Fig. 1.15. The round symbol with a cross in Fig. 1.15 is a *summing junction*, where the various incoming variables are added with appropriate signs (a negative sign indicated near the arrow of a variable implies that variable is to be subtracted from the other incoming variables), and the result of the arithmetic vector sum is the variable shown by the outgoing arrow. Evidently, the controller represents a mathematical relationship (control law) among the plant's observed (or estimated) state, $\mathbf{x_o}(t)$, the reference state, $\mathbf{x_d}(t)$, the control input, $\mathbf{u}(t)$, such as the following:

$$\mathbf{u} = \mathbf{g_f}(\mathbf{x_d}) + \mathbf{g_b}(\mathbf{x_d} - \mathbf{x_o}) = \mathbf{u_d} + \mathbf{g_b}(\mathbf{e_o}), \tag{1.19}$$

where $\mathbf{f_o}(.)$ is the vector functional describing the observer state equation, $\mathbf{g_f}(.)$ the *feedforward control law*, and $\mathbf{g_b}(.)$ is the *feedback control law*. A control system drives the actuators through electrical signals as control inputs, while the sensors produce electrical signals as plant outputs. A controller must manipulate these electrical signals through either an *analog* or a *digital* process, which basically regard the signals as either continuous or discrete variables of time [20], respectively. One has the option of clubbing the sensors and the actuators with either the controller or the plant. Often, it is more convenient to model actuators and sensors as separate subsystems. Note that in Fig. 1.15, we have adopted the convention of including sensors and actuators into the model for the plant. Also note the measurement noise, $\mathbf{w}(t)$, and the process noise, $\boldsymbol{\nu}(t)$, experienced by the plant as random inputs.

Example 1.11. Consider a tracking control system for the single-axis, rigid spacacraft rotation, whose terminal control was analyzed in Example 1.10. A simple,

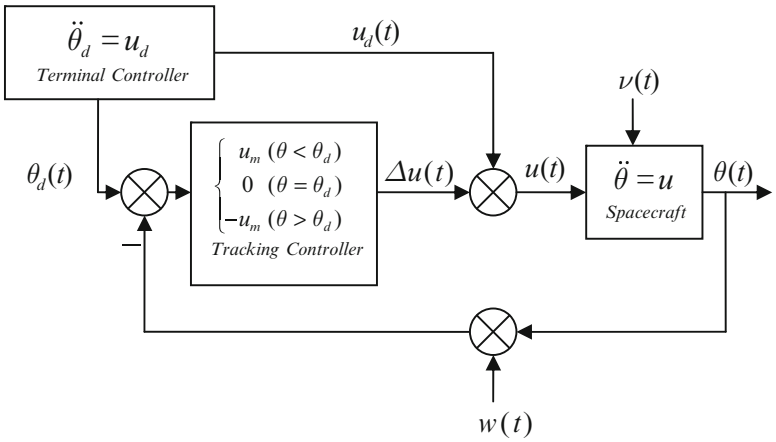


Fig. 1.16 Tracking system for spacecraft rotation with on–off reaction jets

on–off reaction jet actuator is available for correcting small deviations from the nominal trajectory, $\theta_d(t)$, derived in Example 1.10. For simplicity, we shall consider an output feedback law based upon angular deviation,

$$\Delta\theta = \theta - \theta_d,$$

measured by a rate-integrating gyro sensor (Chap. 2). The feedback control law consists of the following switching scheme for the reaction jet system:

$$\Delta u(t) = \begin{cases} u_{\mathrm{m}} & (\theta < \theta_d) \\ 0 & (\theta = \theta_d) \\ -u_{\mathrm{m}} & (\theta > \theta_d) \end{cases},$$

where u_{m} is the same angular acceleration input magnitude as for the nominal actuator. A schematic block-diagram of the tracking control system is shown in Fig. 1.16.

With the numerical data given in Example 1.10, consider an off-nominal initial condition of $\theta(0) = 0.1$ rad (5.7°), $\dot{\theta}(0) = -0.01$ rad/s, and tracking control magnitude, $c = 0.002$ rad/s^2. For simulating the closed-loop spacecraft rotation, we employ a fourth-order Runge-Kutta integration with the state equations coded in the file *example1p11.m* (Table 1.1), which is invoked by *ode45.m* as follows:

```
>>[t,x]=ode45('example1p11',[0 30],[0.1 -0.01]');
```

The required input profile is separately computed by post-processing of the closed-loop trajectory. The resulting control history and trajectory are plotted in Figs. 1.17 and 1.18. Note that the actuator switches four times in the entire control interval (Fig. 1.17), and achieves the final angle (180°) with an error of less than 0.5° (i.e., 0.3%). It may be easily verified by the reader that the terminal error increases

Table 1.1 Program *example1p11.m* for rigid spacecraft rotation controlled by on–off reaction jet actuator

```
function xdot=example1p11(t,x)
um=0.01; % Control magnitude (rad/s^2)
thf=pi; % Desired rotation angle (rad)
tf=30; % Final time (s)
th=x(1); % Actual rotation angle
% Nominal control and trajectory:
a=0.5*(3*tf-sqrt(24*thf/um-3*tf^2));
if t<a
    ud=um*t/a;
    thd=um*t^3/(6*a);
elseif t<tf
    ud=um;
    thd=um*a^2/6+0.5*um*t*(t-a);
else
    ud=0;
    thd=um*(t-tf)*(tf-a/2);
end
dth=th-thd; %Angular error
%Corrective control input:
if dth<0
    du=um;
elseif dth>0
    du=-um;
else
    du=0;
end
% Single-axis rotational dynamics:
xdot(1,1)=x(2);
xdot(2,1)=ud+du;
```

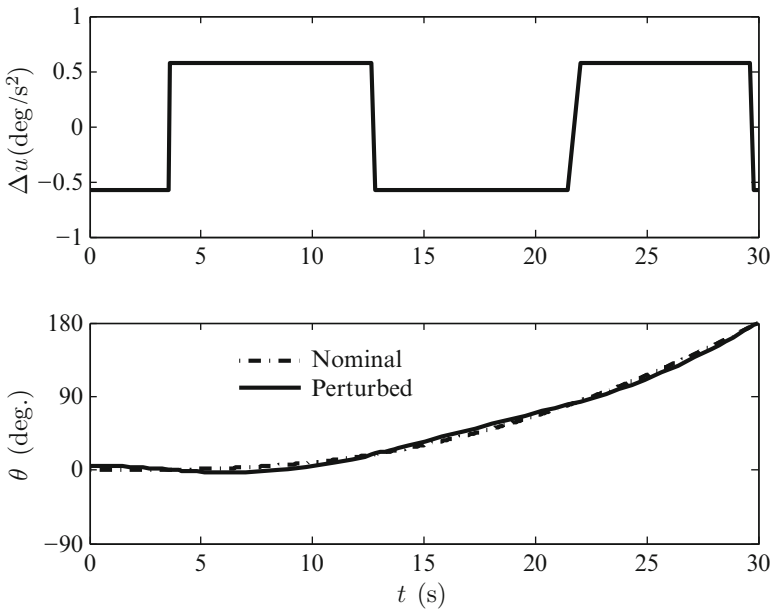


Fig. 1.17 Control history and angular trajectory for the tracking control problem of Example 1.11

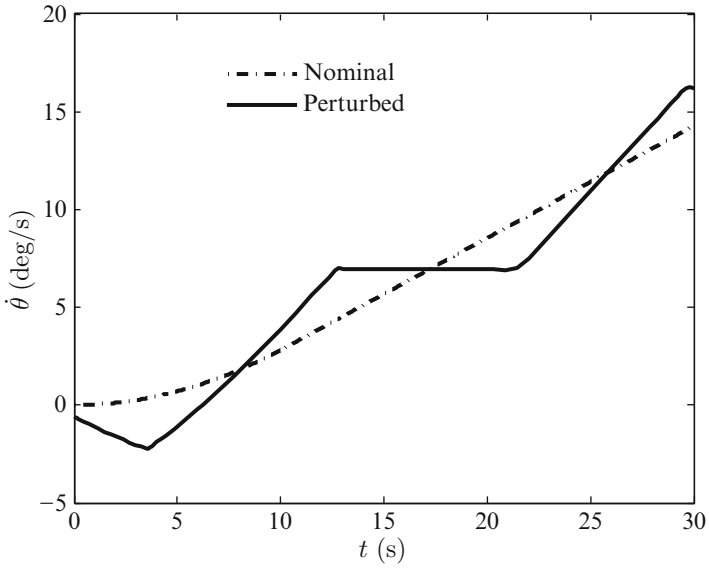


Fig. 1.18 Angular rate profile for the tracking control problem of Example 1.11

to about $-12°$ (7%) if the corrective control is not applied. The tracking action is evident in the plot of $\dot{\theta}$ (Fig. 1.18). It can be shown that the angular rate error would be constant at the initial perturbation, $\dot{\theta}(0) = -0.01$ rad/s ($-0.57°$/s) in the absence of a tracking system.

1.5.2 Control System Performance

The performance of a tracking control system is assessed primarily by the time taken by the plant's state to reach within a specific percentage of the desired reference trajectory, its deviation therefrom (called *overshoot*), and the magnitudes of the control inputs required in the process. Clearly, one can talk about performance of only stable control systems. The behavior of the control system is normally classified into: (a) response at large times, $t \to \infty$, and (b) that at small values of time when large overshoots from the reference trajectory could occur. If the response at large times is brought sufficiently close to the reference trajectory, while avoiding large overshoots in the interim, the control system is said to perform well. In Example 1.11, the maximum overshoot in $\dot{\theta}$ of the tracking system from the nominal trajectory is ± 2.25 deg/s (Fig. 1.18). The time required to come within a small percentage of the reference trajectory – called the *settling-time* – is a practical performance measure, and should be reasonably small. Again referring

to Example 1.11, the angle θ is brought within $\pm 5\%$ of the nominal trajectory in about 24 s. Another measure of performance is the largest control input magnitude required for tracking the reference trajectory.

In order to analyze control system behavior (stability, controllability, observability, performance, etc.), the assumption of small deviations from the reference state is generally employed, which results in a linearized system about the reference trajectory. For a time-invariant system, the reference state is regarded as the equilibrium of the linearized system. A great simplification occurs by making such an assumption, because we can apply the principle of *linear superposition* to a linearized control system, in order to yield the total output due to a linear combination of several input vectors. Linear superposition also enables us to utilize operational calculus and linear algebraic methods for design and analysis.

1.6 Linear Systems

Theory of linear systems is the starting point of the design and analysis of most control systems, and refers to the mathematical framework of a discretized (finite dimensional) system linearized about a particular solution.

A system is said to be linear if the input, $\mathbf{u}(t) = c_1\mathbf{u_1}(t) + c_2\mathbf{u_2}(t)$, produces the output given by $\mathbf{y}(t) = c_1\mathbf{y_1}(t) + c_2\mathbf{y_2}(t)$, where $\mathbf{y_1}(t)$ and $\mathbf{y_2}(t)$ are the respective outputs of the system to $\mathbf{u_1}(t)$ and $\mathbf{u_2}(t)$, and c_1, c_2 are arbitrary scalar constants. By inspecting the governing differential equations of a system, (1.1), it is possible to determine whether it is linear. If the governing differential equations do not contain products and transcendental functions of the state and input variables and their time derivatives, the system is linear.

Let the state equation of a system without disturbance inputs be the following:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U}, t), \tag{1.20}$$

where $\mathbf{X}(t)$ is the state vector and $\mathbf{U}(t)$ is the input vector. The nonlinear vector functional, $\mathbf{f}(.)$, is assumed to possess partial derivatives with respect to state and input variables in the neighborhood of the reference, nominal trajectory $\mathbf{X_d}(t)$, which is a solution to (1.20) and thus satisfies

$$\dot{\mathbf{X}}_\mathbf{d} = \mathbf{f}(\mathbf{X_d}, \mathbf{U_d}, t), \quad t_\mathrm{i} \leq t \leq t_\mathrm{f}, \tag{1.21}$$

where $\mathbf{U_d}(t)$ is the reference input (called the *nominal input*) applied to the system in the interval ($t_\mathrm{i} \leq t \leq t_\mathrm{f}$). In all control applications, the system is driven close to the nominal trajectory by the application of a *control input*, $\mathbf{u}(t)$, defined as the difference between the actual and the nominal input:

$$\mathbf{u} = \mathbf{U} - \mathbf{U_d}, \quad t_\mathrm{i} \leq t \leq t_\mathrm{f}, \tag{1.22}$$

such that the state deviation, $\mathbf{x}(t)$, from the nominal trajectory – given by –

$$\mathbf{x} = \mathbf{X} - \mathbf{X_d}, \quad t_i \leq t \leq t_f, \tag{1.23}$$

remains sufficiently small. The assumption of a small control input causing a small state deviation is inherent in a successful control system design, and leads to the following Taylor series expansion around the nominal trajectory:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(\mathbf{X_d}, \mathbf{U_d}, t)\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{U}}(\mathbf{X_d}, \mathbf{U_d}, t)\mathbf{u} + \mathcal{O}(2), \qquad t_i \leq t \leq t_f, \tag{1.24}$$

where $\mathcal{O}(2)$ denotes the second- and higher-order terms involving control and state deviations.

The Jacobian matrices of $\mathbf{f}$ with respect to $\mathbf{X}$ and $\mathbf{U}$ in the neighborhood of the nominal trajectory are denoted by:

$$\mathsf{A}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(\mathbf{X_d}, \mathbf{U_d}, t); \quad \mathsf{B}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{U}}(\mathbf{X_d}, \mathbf{U_d}, t). \tag{1.25}$$

Retaining only the linear terms in (1.24), we have the following state-space description of the system as a set of first-order, linear, ordinary differential equations:

$$\dot{\mathbf{x}}(t) = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)\mathbf{u}(t), \quad t_i \leq t \leq t_f, \tag{1.26}$$

with the initial condition $\mathbf{X} = \mathbf{X}(t_i)$. Often, the system's governing equations are linearized before expressing them as a set of first-order, nonlinear, state equations (1.20) leading to the same result as (1.26). For the time being, we are ignoring the disturbance inputs to the system, which can be easily included through an additional term on the right-hand side.

Example 1.12. In order to illustrate the method of linearization, consider the aircraft's level flight model of Example 1.3. It is required to navigate the aircraft (i.e., vary its latitude, δ, and longitude, λ) between two given points on planetary surface, by banking (or rolling) such that the altitude, h, and airspeed, v, are maintained constant, and the vehicle always points in the changing direction of flight, $\psi(t)$. The vehicle has a dedicated *autopilot* (Chap. 4), which keeps h, v constants and the aircraft aligned with the velocity vector, while generating the desired bank angle, $\sigma(t)$. The autopilot takes reference commands from a navigational computer that has nominal values of $\lambda(t), \delta(t), \psi(t)$, stored onboard. The equations of motion governing the navigational plant are the following:

$$\dot{\delta} = \frac{v_g \cos\psi}{r}, \tag{1.27}$$

$$\dot{\lambda} = \frac{v_g \sin\psi}{r\cos\delta}, \tag{1.28}$$

$$\dot{\psi} = \frac{\mathcal{L}\sin\sigma}{m v_{\mathrm{g}}} + \frac{v_{\mathrm{g}}}{r}\sin\psi\tan\delta + 2\Omega\sin\delta + \frac{1}{2v_{\mathrm{g}}}\Omega^2 r\sin\psi\sin 2\delta, \quad (1.29)$$

$$\dot{m} = -c, \quad (1.30)$$

where $r = R_0 + h$ is constant, R_0 the planet's radius, $L(t)$ the lift, $c > 0$ the constant fuel consumption, Ω the planetary rotational speed, and the ground speed, v_{g}, is related to the constant airspeed, v, the random wind velocity, $(v_{\mathrm{w}}, \psi_{\mathrm{w}})$, and flight direction, ψ, by the following equation [21]:

$$v_{\mathrm{g}} = \sqrt{v^2 - v_{\mathrm{w}}^2 \sin^2(\psi_{\mathrm{w}} - \psi)} - v_{\mathrm{w}}\cos(\psi_{\mathrm{w}} - \psi). \quad (1.31)$$

As the fuel consumption, c, is constant, we can eliminate the fourth state equation by writing instead,

$$m(t) = m_0 - ct. \quad (1.32)$$

Therefore, the system's order is reduced to 3, and we select the state vector of the nonlinear navigational plant as follows:

$$\mathbf{X}(t) = [\delta(t), \lambda(t), \psi(t)]^{\mathrm{T}}. \quad (1.33)$$

The normal acceleration,

$$U(t) = \frac{\mathcal{L}}{m}\sin\sigma, \quad (1.34)$$

is taken to be the sole control input. The autopilot modulates the lift, $\mathcal{L}$, the bank angle, σ, and the thrust, T, through actuators, such that a steady and level flight (h, v constants) is maintained at all times, irrespective of a changing mass, $m(t)$, and random wind velocity, $(v_{\mathrm{w}}, \psi_{\mathrm{w}})$. This requires satisfying the following additional equations:

$$\mathcal{L}\cos\sigma = mg; \qquad T = \mathcal{D}, \quad (1.35)$$

where the drag, $\mathcal{D}$, is a function of the lift, $\mathcal{L}$. A block-diagram of the navigational system is shown in Fig. 1.19. Usually, because the wind speed and direction are unknown random disturbances, the autopilot works by assuming $v_{\mathrm{g}} \simeq v$. Therefore, for maintaining steady and level flight as well as successful tracking of a nominal trajectory, the measured (or estimated) state vector is supplied to the autopilot by a feedback loop (as shown in Fig. 1.19).

We can express the state equations in the following vector form:

$$\dot{\mathbf{X}} = \mathbf{f}[\mathbf{X}(t), U(t), t], \quad (1.36)$$

$$\mathbf{f} = \begin{Bmatrix} \dfrac{v_{\mathrm{g}}}{r}\cos\psi \\ \dfrac{v_{\mathrm{g}}\sin\psi}{r\cos\delta} \\ \dfrac{U}{v_{\mathrm{g}}} + \dfrac{v_{\mathrm{g}}}{r}\sin\psi\tan\delta + 2\Omega\sin\delta + \dfrac{\Omega^2 r}{2v_{\mathrm{g}}}\sin\psi\sin 2\delta \end{Bmatrix}. \quad (1.37)$$

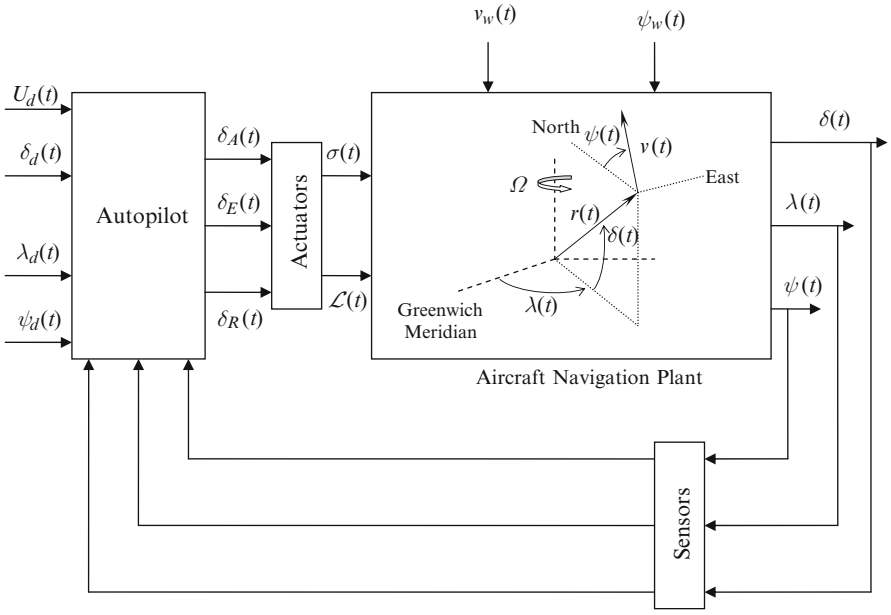


Fig. 1.19 Navigational system for an aircraft

Linearization of the state equations about a nominal (reference) trajectory, $[\delta_d(t), \lambda_d(t), \psi_d(t)]$, passing through the given waypoints is carried out as follows:

$$\dot{\mathbf{x}} = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)u(t), \tag{1.38}$$

where $\mathbf{x} = \mathbf{X} - \mathbf{X_d}$, $u = U - U_d$,

$$\mathbf{A}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\right|_d = \begin{pmatrix} 0 & 0 & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & 0 & a_{33} \end{pmatrix}; \qquad \mathbf{B}(t) = \left.\frac{\partial \mathbf{f}}{\partial U}\right|_d = \begin{pmatrix} 0 \\ 0 \\ 1/v_{\mathrm{g}} \end{pmatrix}, \tag{1.39}$$

with

$$a_{13} = -\frac{v_{\mathrm{g}}}{r}\sin\psi_d + \frac{\cos\psi_d}{r}\frac{\partial v_{\mathrm{g}}}{\partial \psi_d}, \tag{1.40}$$

$$a_{21} = \frac{v_{\mathrm{g}}}{r}\sin\psi_d \tan\delta_d \sec\delta_d, \tag{1.41}$$

$$a_{23} = \frac{\sec\delta_d}{r}\left(v_{\mathrm{g}}\cos\psi_d + \frac{\partial v_{\mathrm{g}}}{\partial \psi_d}\sin\psi_d\right), \tag{1.42}$$

$$a_{31} = \frac{v_g}{r} \sin\psi_d \sec^2\delta_d + \Omega^2 \frac{r}{v_g} \sin\psi_d \cos 2\delta_d + 2\Omega \cos\delta_d, \tag{1.43}$$

$$a_{33} = \frac{\tan\delta_d}{r}\left(v_g \cos\psi_d + \frac{\partial v_g}{\partial\psi_d}\sin\psi_d\right) - \frac{u_d}{v_g^2}\frac{\partial v_g}{\partial\psi_d} + \frac{\Omega^2 r}{2}\sin 2\delta_d \left(\frac{\cos\psi_d}{v_g} - \frac{\sin\psi_d}{v_g^2}\frac{\partial v}{\partial\psi_d}\right), \tag{1.44}$$

$$v_g = \sqrt{v^2 - v_w^2 \sin^2(\psi_w - \psi_d)} - v_w \cos(\psi_w - \psi_d), \tag{1.45}$$

and

$$\frac{\partial v_g}{\partial\psi_d} = \left[1 - \frac{v_w \cos(\psi_w - \psi_d)}{\sqrt{v^2 - v_w^2 \sin^2(\psi_w - \psi_d)}}\right] v_w \sin(\psi_w - \psi_d). \tag{1.46}$$

Here, the subscript d refers to the nominal values. A likely nominal trajectory is the *great circle route* [21], which is the shortest distance between any two points on a sphere.

1.6.1 Solution of Linear State Equations

When the applied control input is zero ($\mathbf{u}(t) = \mathbf{0}$), (1.26) becomes the following *homogeneous* equation:

$$\dot{\mathbf{x}}(t) = \mathsf{A}(t)\mathbf{x}(t). \tag{1.47}$$

The solution to the homogeneous state equation with the initial condition $\mathbf{x_i} = \mathbf{x}(t_i)$ is expressed as follows:

$$\mathbf{x}(t) = \Phi(t, t_i)\mathbf{x}(t_i), \quad \text{for all } t, \tag{1.48}$$

where $\Phi(t, t_i)$ is called the *state transition matrix*. The state-transition matrix thus has the important property of transforming the state at time t_i to another time t. Other important properties of the state-transition matrix are the following:

$$\Phi(t, t_i) = -\Phi(t_i, t). \tag{1.49}$$

$$\Phi(t, t_i) = \Phi(t, t_0)\Phi(t_0, t_i). \tag{1.50}$$

$$\frac{\mathrm{d}\Phi(t,t_i)}{\mathrm{d}t} = \mathsf{A}(t)\Phi(t, t_i). \tag{1.51}$$

Equations (1.49)–(1.51) are said to represent the inverse, associative, and derivative properties of the state-transition matrix.

The general solution of nonhomogeneous state equation, (1.26), subject to initial condition, $\mathbf{x_i} = \mathbf{x}(t_i)$, can be written in terms of $\Phi(t, t_i)$ as follows:

$$\mathbf{x}(t) = \Phi(t, t_i)\mathbf{x_i} + \int_{t_i}^{t} \Phi(t, \tau)\mathsf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau, \qquad \text{for all } t, \tag{1.52}$$

which can be verified by substituting into (1.26), along with the properties of $\Phi(t, t_i)$.

The output (or *response*) of a linear system can be expressed as follows:

$$\mathbf{y}(t) = \mathsf{C}(t)\mathbf{x}(t) + \mathsf{D}(t)\mathbf{u}(t), \tag{1.53}$$

where $\mathsf{C}(t)$ is called the *output coefficient matrix* and $\mathsf{D}(t)$ is called the *direct transmission matrix*. When a stochastic measurement noise is present in the system, an additional term appears on the right-hand side of (1.53).

Substituting (1.47) into (1.53), we have the following expression for the system's response:

$$\mathbf{y}(t) = \mathsf{C}(t)\Phi(t, t_i)\mathbf{x_i} + \int_{t_i}^{t} [\mathsf{C}(t)\Phi(t, \tau)\mathsf{B}(\tau) + \mathsf{D}(t)\delta(t - \tau)]\mathbf{u}(\tau)\mathrm{d}\tau, \quad t \geq t_i, \tag{1.54}$$

where $\delta(t - \tau)$ is *Dirac delta* function representing a unit impulse applied at $t = \tau$. The first term on the right-hand side of (1.54) is called the *initial response*, while the integral term is called the *convolution integral* of the linear system. The convolution integral gives the system's response when the initial condition is zero ($\mathbf{x_i} = \mathbf{0}$), and is denoted by:

$$\mathbf{y}(t) = \int_{t_i}^{t} \mathsf{G}(t, \tau)\mathbf{u}(\tau)\mathrm{d}\tau, \quad t \geq t_i, \tag{1.55}$$

where $\mathsf{G}(t, \tau)$ is called the *impulse response* matrix, given by:

$$\mathsf{G}(t, \tau) = \mathsf{C}(t)\Phi(t, \tau)\mathsf{B}(\tau) + \mathsf{D}(t)\delta(t - \tau), \quad t \geq \tau. \tag{1.56}$$

The element (i, j) of the impulse response matrix, $\mathsf{G}(t, \tau)$, is the value of the ith output variable at time t when the jth input variable is a unit impulse function applied at time τ, subject to zero initial condition, $\mathbf{x_i} = \mathbf{0}$. Similarly, we define the *step response* matrix as the integral of $\mathsf{G}(t, \tau)$:

$$\mathsf{S}(t, t_i) = \int_{t_i}^{t} \mathsf{G}(t, \tau)\mathrm{d}\tau, \quad t \geq t_i. \tag{1.57}$$

The element (i, j) of the step response matrix, $\mathsf{S}(t, t_i)$, is the value of the ith output variable at time t when the jth input variable is a unit step applied at time t_i, subject to zero initial condition, $\mathbf{x_i} = \mathbf{0}$. Both impulse and step response matrices hold a special significance in linear system analysis.

The derivation of the state-transition matrix (thus initial, impulse, and step responses) is a formidable task for a linear system with time-varying coefficient matrices (called a *linear time-varying system*). Only in some special cases, a closed-form expression for $\Phi(t, t_i)$ can be derived. Whenever $\Phi(t, t_i)$ cannot be obtained in a closed-form, it is necessary to apply numerical techniques for the solution of the state equation.

Example 1.13. Consider the following linear time-varying system:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{pmatrix} -\cos t & 0 & \sin 2t \\ -\sin 2t & t\cos t & 0 \\ 0 & -1 & e^{-t} \end{pmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{pmatrix} 0 & 1 \\ -1 & 0 \\ 0.2 & -0.1 \end{pmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

The response of the system to zero initial state, $\mathbf{x}(0) = \mathbf{0}$, $u_1(t) = 0.1 \sin 5t$, and $u_2(t) = 0.2 \cos 5t$, is computed for the first 2 s using *ode45.m* as follows:

```
[t,x]=ode45('example1p13',[0 2],[0 0 0]');
```

where *example1p13.m* is the following function M-file that specifies the state equations of the closed-loop system:

```
function xdot=example1p13(t,x)
u1=0.1*sin(5*t);
u2=0.2*cos(5*t);
xdot(1,1)=-x(1)*cos(t)+x(3)*sin(2*t)+u2;
xdot(2,1)=-x(1)*sin(2*t)+x(2)*t*cos(t)-u1;
xdot(3,1)=-x(2)+x(3)*exp(-t)+0.2*u1-0.1*u2;
```

The resulting plot of the state variables is shown in Fig. 1.20. The growing amplitude of response to bounded inputs indicates an unstable system.

1.6.2 Linear Time-Invariant Systems

Generally, the time scale of rotational dynamics is so small that most flight vehicles can be assumed to have nearly constant coefficient matrices of attitude dynamics at a given equilibrium point. In such a case, the system is approximated to be a *linear time-invariant* (LTI) system, with A, B treated as constant matrices. The linear time-invariant case is easily handled by writing

$$\Phi(t, t_i) = e^{\mathsf{A}(t-t_i)}, \tag{1.58}$$

where e^{M} is called the *matrix exponential* of a square matrix, M, and is defined by the following infinite series (similar to a scalar exponential):

$$e^{\mathsf{M}} \doteq \mathsf{I} + \mathsf{M} + \frac{1}{2}\mathsf{M}^2 + \cdots + \frac{1}{n!}\mathsf{M}^n + \cdots. \tag{1.59}$$

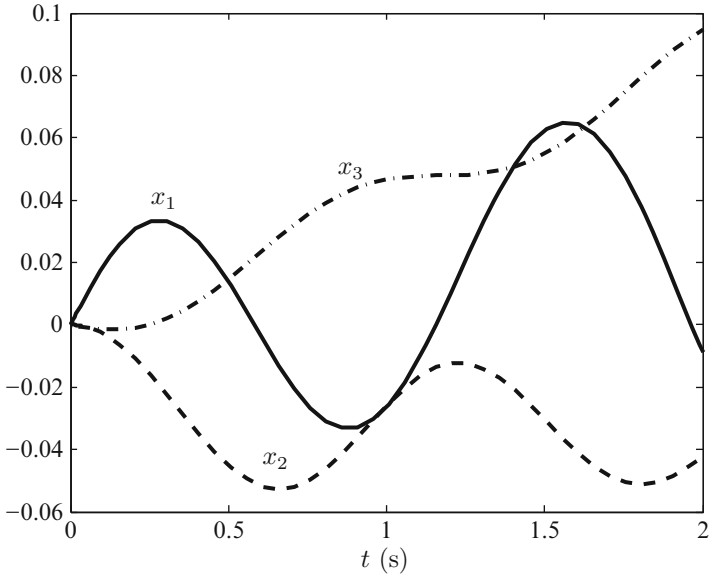


Fig. 1.20 Response of the linear time-varying system of Example 1.13

While the evaluation of $e^{\mathsf{A}(t-t_i)}$ by this infinite series would be impossible in practice, a suitable numerical approximation of (1.59) with only a finite number of terms can be utilized by breaking the time interval, $t - t_\mathrm{i}$, into a number of small intervals. This forms the basis of numerical evaluation of the state-transition matrix for time-invariant, linear systems. However, for a small order system, an exact, closed-form expression for $e^{\mathsf{A}(t-t_i)}$ can be obtained with the use of Laplace transform.

Consider the Laplace transform (Chap. 3) of the state equation, (1.26), for an LTI system subject to the initial condition, $\mathbf{x_0} = \mathbf{x}(0)$:

$$s\boldsymbol{\mathcal{X}}(s) - \mathbf{x_0} = \mathsf{A}\boldsymbol{\mathcal{X}}(s) + \mathsf{B}\boldsymbol{\mathcal{U}}(s), \tag{1.60}$$

where $\boldsymbol{\mathcal{X}}(s)$ and $\boldsymbol{\mathcal{U}}(s)$ are the Laplace transforms of $\mathbf{x}(t)$ and $\mathbf{u}(t)$, respectively. We have implicitly assumed that both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ satisfy the conditions for the existence of Laplace transform [20], i.e., they are piecewise continuous functions bounded by an exponential. In order to derive the state-transition matrix, we make the input vanish, which leads to

$$(s\mathsf{I} - \mathsf{A})\boldsymbol{\mathcal{X}}(s) = \mathbf{x_0}, \tag{1.61}$$

or,

$$\boldsymbol{\mathcal{X}}(s) = (s\mathsf{I} - \mathsf{A})^{-1}\mathbf{x_0}. \tag{1.62}$$

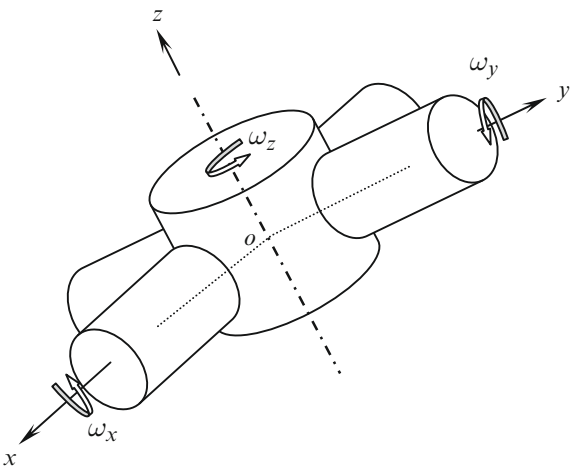


Fig. 1.21 Axisymmetric spacecraft rotation (Example 1.14)

By taking the inverse Laplace transform of (1.62), we have

$$\mathbf{x}(t) = \mathcal{L}^{-1}(s\mathsf{I} - \mathsf{A})^{-1}\mathbf{x_0}, \tag{1.63}$$

which upon comparison with (1.48) with $t_\mathrm{i} = 0$ yields

$$e^{\mathsf{A}t} = \mathcal{L}^{-1}(s\mathsf{I} - \mathsf{A})^{-1}. \tag{1.64}$$

By substituting (1.58) into (1.52) with $t_\mathrm{i} = 0$, we can write the following general expression for the state of a linear, time-invariant system in the presence of an arbitrary, Laplace transformable input, which begins to act at time $t = 0$ when the system's state is $\mathbf{x}(0) = \mathbf{x_0}$:

$$\mathbf{x}(t) = e^{\mathsf{A}t}\mathbf{x_0} + \int_0^t e^{\mathsf{A}(t-\tau)}\mathsf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau. \tag{1.65}$$

The first term on the right-hand side (initial response) decays to zero for an asymptotically stable system in the limit $t \to \infty$. However, in the same limit, the integral term may either converge to a finite value (called *steady state*) or assume the same functional form as that of the input (called *forced response*).

The solution given by (1.65) can be computed by *discretization* of time into small intervals, and evaluating the initial response as well as the integral term by appropriate approximations, as discussed in [20]. Such an approach is programmed in the intrinsic MATLAB function *lsim*, which can also be invoked by a *Simulink* block-diagram [20].

Example 1.14. Consider a torque-free spacecraft (Fig. 1.21) rotating about its axis of symmetry, oz, which yields $J_{xx} = J_{yy}$. The angular velocity components,

$\omega_x, \omega_y, \omega_z$), in the principal body-fixed frame, ($oxyz$), obey the following differential equations (Chap. 2):

$$\begin{aligned} J_{xx}\dot{\omega}_x + \omega_y\omega_z(J_{zz} - J_{xx}) &= 0 \\ J_{xx}\dot{\omega}_y + \omega_x\omega_z(J_{xx} - J_{zz}) &= 0 \\ J_{zz}\dot{\omega}_z &= 0. \end{aligned} \tag{1.66}$$

It is clear from (1.66) that the spacecraft is in a state of equilibrium whenever $\omega_x = \omega_y = 0$, called *pure spin* about the axis of symmetry. It is also evident from the last equation that $\dot{\omega}_z = 0$, or $\omega_z = n =$ constant, irrespective of the magnitudes of ω_x, ω_y. Let us assume that the spacecraft is initially in a state of pure spin, when a disturbance, $\omega_x(0), \omega_y(0)$, is applied at the time $t = 0$. The resulting motion of the spacecraft is given by the following state equation:

$$\begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \end{Bmatrix} = \begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \end{Bmatrix}, \tag{1.67}$$

where $k = n\frac{(J_{zz}-J_{xx})}{J_{xx}}$ is a real constant. The solution to (1.67) with the initial condition, $\omega_x(0), \omega_y(0)$, is given by:

$$\begin{Bmatrix} \omega_x(t) \\ \omega_y(t) \end{Bmatrix} = e^{\mathsf{A}t} \begin{Bmatrix} \omega_x(0) \\ \omega_y(0) \end{Bmatrix}, \tag{1.68}$$

where $e^{\mathsf{A}t}$ is the matrix exponential with

$$\mathsf{A} = \begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix}. \tag{1.69}$$

Expressing the matrix exponential as the inverse Laplace transform of the resolvent, we have

$$e^{\mathsf{A}t} = \mathcal{L}^{-1}(s\mathsf{I} - \mathsf{K})^{-1} = \begin{pmatrix} \cos(kt) & -\sin(kt) \\ \sin(kt) & \cos(kt) \end{pmatrix}. \tag{1.70}$$

Therefore, the solution to the state equation is given by:

$$\begin{aligned} \omega_x(t) &= \omega_x(0)\cos(kt) - \omega_y(0)\sin(kt) \\ \omega_y(t) &= \omega_x(0)\sin(kt) + \omega_y(0)\cos(kt), \end{aligned} \tag{1.71}$$

which implies that the rotation of an axisymmetric, rigid spacecraft, disturbed from the equilibrium state of pure spin about the axis of symmetry by a disturbance, $\omega_x(0), \omega_y(0)$, is purely oscillatory in the oxy plane, while the spin rate, $\omega_z = n$,

remains constant. This causes a coning motion (called *precession*) about the axis of symmetry. An important characteristic of the solution given by (1.71) is the following:

$$\omega_{xy}^2 \doteq \omega_x^2 + \omega_y^2 = \omega_x^2(0) + \omega_y^2(0) = \text{ constant}, \tag{1.72}$$

which implies that the magnitude of the angular velocity component in a cross-sectional plane is constant at ω_{xy}. Since the precessional rate is bounded by that of the applied disturbance, we say that the rotation of a rigid spacecraft about its axis of symmetry is unconditionally stable. This inherent stability forms the principle behind *spin stabilization* of other axisymmetric objects, such as rifle bullets, tank shells, and missiles, where aerodynamic effects cause a small disturbance normal to the axis of symmetry.

1.6.3 Linear Stability Criteria

Apart from determining the state-transition matrix, the Laplace transform approach is also useful in a linear stability analysis. Equation (1.61) represents an *eigenvalue problem*, whose solution yields the *eigenvalues*, s, and *eigenvectors*, $\boldsymbol{\mathcal{X}}(s)$. The eigenvalues of the linear system are obtained by the following *characteristic equation*, which results from a nontrivial solution of (1.61):

$$|s\mathsf{I} - \mathsf{A}| = 0. \tag{1.73}$$

Hence, the eigenvalues of the constant coefficient matrix, A, are the roots of the characteristic equation, (1.73). Generally, the characteristic equation of a system of order n is expressed as a polynomial equation as follows:

$$|s\mathsf{I} - \mathsf{A}| = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0 = 0, \tag{1.74}$$

where the *characteristic coefficients*, a_i, $i = 0 \ldots (n-1)$, are invariant with the choice of the state variables (i.e., they are unique parameters of the system). The n complex roots of the characteristic equation (eigenvalues of A) signify *stability*.

From the expression of the system's state, (1.65), it can be deduced that variation of the unforced ($\mathbf{u} = \mathbf{0}$) system's state with time is given by terms containing e^{st}.[1] When an eigenvalue, s_k, is repeated p times, its contribution to the system's state involves terms of the form $t^i e^{s_k t}$, $i = 0 \ldots (p-1)$. Considering that an eigenvalue

[1] It is easily seen that a *characteristic vector*, $\mathbf{x}(t) = \mathbf{X_k} e^{s_k t}$ satisfies the homogeneous state equation, where $\mathbf{X_k}$ is the eigenvector corresponding to the eigenvalue, s_k. The general solution, which must also satisfy the initial condition, $\mathbf{x}(0) = \mathbf{x_0}$, is a combination of all n characteristic vectors. The decomposition of a system's state into the characteristic vectors (or *modes*) is an alternative way of computing the state-transition matrix [20].

is generally complex, its imaginary part denotes the frequency of oscillation of the characteristic vector about the equilibrium point, and the real part signifies the growth (or decay) of its amplitude with time. We can now state the following criteria for the stability of a linear, time-invariant system:

(a) If all eigenvalues have negative real parts, the linear system is stable about its equilibrium point. Such a system is called an *asymptotically stable* system. A system having an eigenvalue with zero real part (and the remaining eigenvalues with negative real parts) displays oscillatory behavior of a constant amplitude, and is said to be stable (but not asymptotically stable). If at least one eigenvalue has a positive real part, its contribution to the system's state is an exponentially growing amplitude, and the system is said to be *unstable*.
(b) If a multiple eignevalue of multiplicity p has a zero real part, its contribution to the system's state has terms containing the factors t^i, $i = 0 \ldots (p-1)$, which signify an unbounded behavior with time. In such a case, the linear system would be unstable about its equilibrium. As complex eigenvalues occur in conjugate pairs, a repeated eigenvalue with zero real part must also have a zero imaginary part, i.e., $s = 0$.

The linear stability criteria can be applied to either numerically determined eigenvalues from the coefficient matrix, A, or indirectly by inspection of the characteristic polynomial. As we would normally have a numerical procedure available to us via MATLAB, we can adopt the direct evaluation of the eigenvalues. Much of linear systems theory is traditionally devoted to stability analysis by analytical and graphical methods such as *Root-Locus* and *Nyquist* diagrams [20]. However, these classical methods, while imparting significant insight into the system's behavior, are largely limited to *single-input, single-output* (SISO) systems.

Example 1.15. Consider an LTI system with the following state dynamics matrix:

$$\mathsf{A} = \begin{pmatrix} -1 & 2 & 3 & 0 \\ -2 & 0.5 & 0 & -1 \\ -3 & 1 & -2 & 0 \\ 0 & 1 & 2 & -3 \end{pmatrix}.$$

The eigenvalues of the matrix A can be determined by the MATLAB-CST command *damp* as follows (an alternative is the basic MATLAB command *eig*):

```
>> A=[-1 2 3 0; -2 0.5 0 -1; -3 1 -2 0; 0 1 2 -3]

A =
   -1.0000    2.0000    3.0000         0
   -2.0000    0.5000         0   -1.0000
   -3.0000    1.0000   -2.0000         0
         0    1.0000    2.0000   -3.0000

>> damp(A)
```

```
        Eigenvalue            Damping      Freq. (rad/s)

-8.74e-001 + 3.57e+000i     2.38e-001      3.67e+000
-8.74e-001 - 3.57e+000i     2.38e-001      3.67e+000
-3.12e+000                  1.00e+000      3.12e+000
-6.30e-001                  1.00e+000      6.30e-001
```

Thus the system – having all eigenvalues with negative real parts – is asymptotically stable. There is a complex conjugate pair and two real eigenvalues.

1.6.4 Controllability of Linear Systems

Linear, time-invariant (LTI) systems must satisfy a necessary and sufficient condition for controllability [8] wherein for the pair (A, B) to be controllable, the following test matrix must have rank n, the order of the system:

$$\mathsf{P} = \left(\mathsf{B},\ \mathsf{AB},\ \mathsf{A}^2\mathsf{B}\,, \ldots,\ \mathsf{A}^{n-1}\mathsf{B}\right). \tag{1.75}$$

Example 1.16. Analyze the stability and controllability of the following system:

$$\mathsf{A} = \begin{pmatrix} -1 & 0 & 2 \\ 0 & 0 & -2 \\ -3 & 0 & 0 \end{pmatrix}; \qquad \mathsf{B} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

The eigenvalues of A are obtained from the characteristic equation,

$$|s\mathsf{I} - \mathsf{A}| = s(s^2 + s + 6),$$

whose roots are $s_1 = 0$ and $s_{2,3} = -0.5(1 \pm i\sqrt{23})$. Hence, the system is stable, but not asymptotically stable.

We next form the controllability test matrix as follows:

$$\mathsf{P} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

which is clearly singular, and has rank only 1. Therefore, the system is uncontrollable.

MATLAB's Control Systems Toolbox (CST) provides the command *ctrb* for quickly forming and carrying out the controllability test for LTI systems [20].

Example 1.17. A spacecraft equipped with a pitch momentum wheel and roll-yaw reaction wheels (Chap. 7) has the following state-space representation:

$$\mathsf{A} = \begin{pmatrix} 0 & 0 & 0.0012 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ -0.0012 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 & 0.0659 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0994 & 0 & 0 \end{pmatrix}$$

$$\mathsf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.0067 & 0 & 0 \\ 0 & -0.0050 & 0 \\ 0 & 0 & -0.0100 \end{pmatrix}.$$

Examine the controllability of the system when:

(a) Only the first input is applied.
(b) Only the second input is applied.
(c) Only the third input is applied.
(d) The first and the second inputs are applied together.
(e) The first and the third inputs are applied together.

We use the MATLAB-CST command *ctrb* to conduct controllability tests as follows:

```
>> A=[   0         0    0.0012    1.0000         0         0
         0         0         0         0    1.0000         0
   -0.0012         0         0         0         0    1.0000
         0         0         0         0         0    0.0659
         0         0         0         0         0         0
         0         0         0   -0.0994         0         0 ];

>> B=[   0         0         0
         0         0         0
         0         0         0
   -0.0067         0         0
         0   -0.0050         0
         0         0   -0.0100  ];

>> rank(ctrb(A,B(:,1))) %case (a)

ans =     4

>> rank(ctrb(A,B(:,2))) %case (b)

ans =     2

>> rank(ctrb(A,B(:,3))) %case (c)

ans =     4
```

```
>> rank(ctrb(A,B(:,1:2))) %case (d)

ans =     6

>> rank(ctrb(A,[B(:,1) B(:,3)])) %case (e)

ans =     4
```

Thus, of all the cases given earlier, only Case (d) is controllable.

As discussed earlier, if a system is unstable but controllable, it can be stabilized by a feedback control system. It is often possible to decompose an uncontrollable LTI system into controllable and uncontrollable subsystems. A system that is both unstable and uncontrollable could also be stabilized, provided its uncontrollable subsystem is stable. In such a case, the system is said to be stabilizable.

Controllability (or at least stabilizability) of a plant is necessary before a regulator can be designed for it, as we shall see in Chap. 3.

1.6.5 Observability of Linear Systems

An unforced LTI system, $\dot{\mathbf{x}} = \mathsf{A}\mathbf{x}$, whose output is related to the state vector by

$$\mathbf{y} = \mathsf{C}\mathbf{x}, \tag{1.76}$$

must satisfy the necessary and sufficient condition for observability in a manner similar to the controllability test, wherein for the pair (A, C) to be observable, the following test matrix must have rank n, the order of the system:

$$\mathsf{N} = \left(\mathsf{C}^{\mathrm{T}},\ \mathsf{A}^{\mathrm{T}}\mathsf{C}^{\mathrm{T}},\ (\mathsf{A}^{\mathrm{T}})^2\mathsf{C}^{\mathrm{T}},\ \ldots,\ (\mathsf{A}^{\mathrm{T}})^{n-1}\mathsf{C}^{\mathrm{T}}\right). \tag{1.77}$$

Example 1.18. Analyze the observability of the following system:

$$\mathsf{A} = \begin{pmatrix} 0 & -1 & 2 \\ 0 & -2 & 1 \\ 0 & 1 & -3 \end{pmatrix}; \qquad \mathsf{C} = (1,\ 0,\ -1)\,.$$

We form the observability test matrix as follows:

$$\mathsf{N} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -2 & 9 \\ -1 & 5 & -17 \end{pmatrix},$$

which is clearly nonsingular, i.e., has full rank of 3. Therefore, the system is observable.

Observability (or at least detectability) of a plant is necessary before an observer can be designed for it.

1.7 Aerospace Vehicle Guidance and Control

All aerospace vehicles require a precise control of position, velocity, and attitude for a successful mission. The flight control tasks can be broadly classified into two distinct categories: (a) control of position and linear velocity – called *guidance* (or *navigation*) – of the vehicle's center of mass relative to a given reference frame and (b) control of the vehicle's angular orientation (or *attitude*) and angular velocity about the center of mass relative to another frame of reference. Typically, the reference frame for guidance is fixed to a planet with origin either at its center (planet centered frame) or on its surface (ground frame) with the latter being employed for aircraft and short-range missiles, and the former for spacecraft, long-range rockets, and entry vehicles. In contrast, the reference frame for attitude control has its origin at the vehicle's center of mass and thus translates with the vehicle, and may either be fixed with respect to distant stars (celestial or inertial frame) or may be rotating at a constant rate (e.g., local-horizon frame and wind-axes discussed in Chap. 2).

The characteristic time-scale of translational dynamics is typically an order of magnitude larger than that of attitude dynamics. In high-speed aircraft, rockets, and spacecraft, the attitude control loop has too small a characteristic time-scale to be managed manually, thus necessitating automatic control. In even slower vehicles, the task of accurately controlling the attitude for special guidance tasks (such as automated landing) can become very demanding for the pilot. Thus, it is natural to have the attitude control system act as a slave (or actuator) to the guidance control system, as shown in Fig. 1.22. The outer guidance loop takes reference inputs from either the pilot or an automated flight management system that has the desired trajectory stored on-board. The automatic attitude control system, in contrast, is not directly manipulated by the pilot, but rotates the vehicle as desired by the guidance system. A fully automated (autonomous) vehicle is one in which both guidance and control systems function without manual intervention.

Example 1.19. Consider the flight of a rocket in the vertical plane (Fig. 1.23), with airspeed, $v(t)$, radial location from Earth's center, $r(t)$, flight path angle, $\phi(t)$, mass $m(t)$, atmospheric density, $\rho(r)$, acceleration due to gravity, $g(r)$, of a spherical planet, parasite drag coefficient, C_{D0}, reference area, A, propellant exhaust speed, v_e, and propellant mass flow rate input, $\beta(t)$, that can be changed by throttling in the limits, $0 \leq \beta \leq \beta_m$. The important equations of translational motion (Chap. 2) are the following:

$$\begin{aligned}
\dot{r} &= v \sin\phi \\
\dot{v} &= \beta \frac{v_e}{m} - \frac{\rho C_{D0} A v^2}{2m} - g \sin\phi \\
\dot{\phi} &= \left(\frac{v}{r} - \frac{g}{v}\right) \cos\phi \\
\dot{m} &= -\beta,
\end{aligned} \tag{1.78}$$

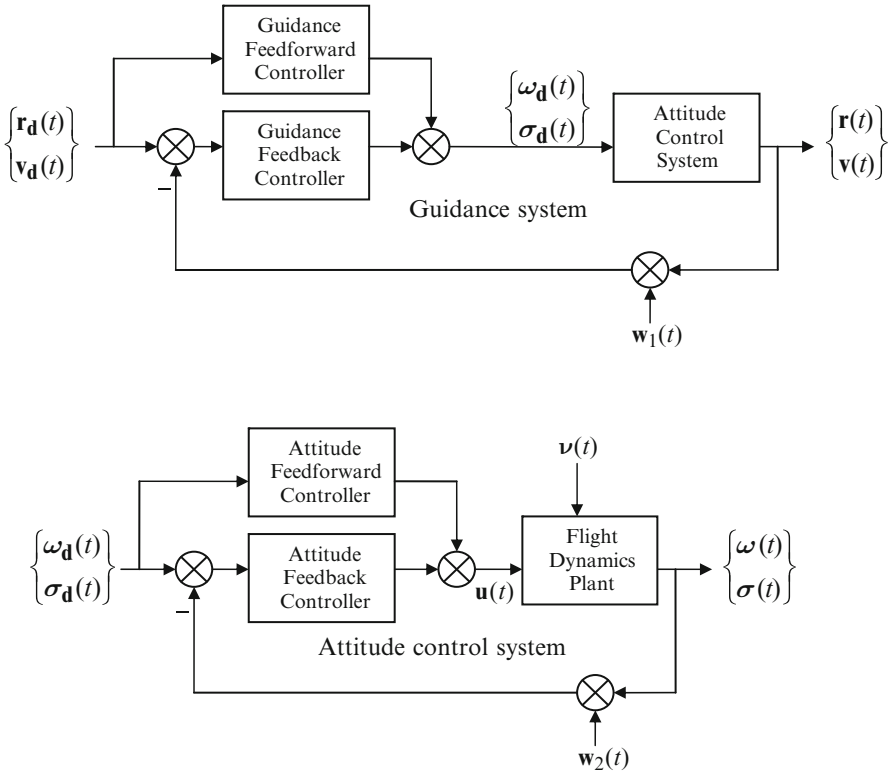


Fig. 1.22 Guidance and attitude control systems

where it is assumed that the normal acceleration due to lift, sideforce, and normal thrust component is maintained zero by an attitude control system. Such a trajectory is commonly flown by spacecraft launch vehicles and long-range ballistic missiles for structural reasons, and involves a naturally curving flight path under the influence of gravity called *gravity-turn*. The throttle input is varied by a terminal controller, whose objective is to reach a final speed, $v(t_f)$, and radius, $r(t_f)$, at the point of propellant exhaustion (or burn-out), $t = t_f$. The terminal controller consists of a feedforward path that provides predetermined nominal profiles, $r_d(t)$, $v_d(t)$, $\phi_d(t)$, and $\beta_d(t)$ that are stored on-board, and specified as reference trajectory to be tracked by a feedback guidance loop (Fig. 1.24).

The attitude dynamics of the rocket is represented by the following linearized state equations in terms of the pitch angle, $\theta(t)$, pitch rate, $q(t)$, and the (small) thrust deflection angle, μ:

$$\begin{aligned} \dot{\theta} &= q \\ \dot{q} &= \frac{\beta v_e}{J}\ell\mu + \frac{M_q}{J}q, \end{aligned} \tag{1.79}$$

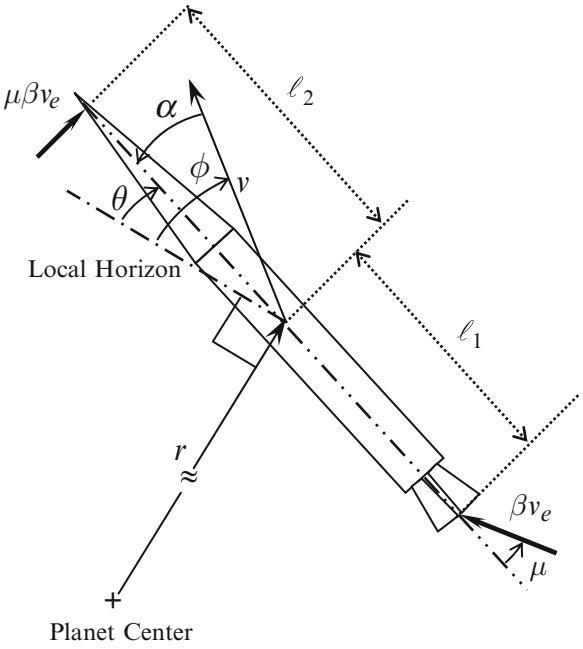


Fig. 1.23 Flight of a rocket in a vertical plane

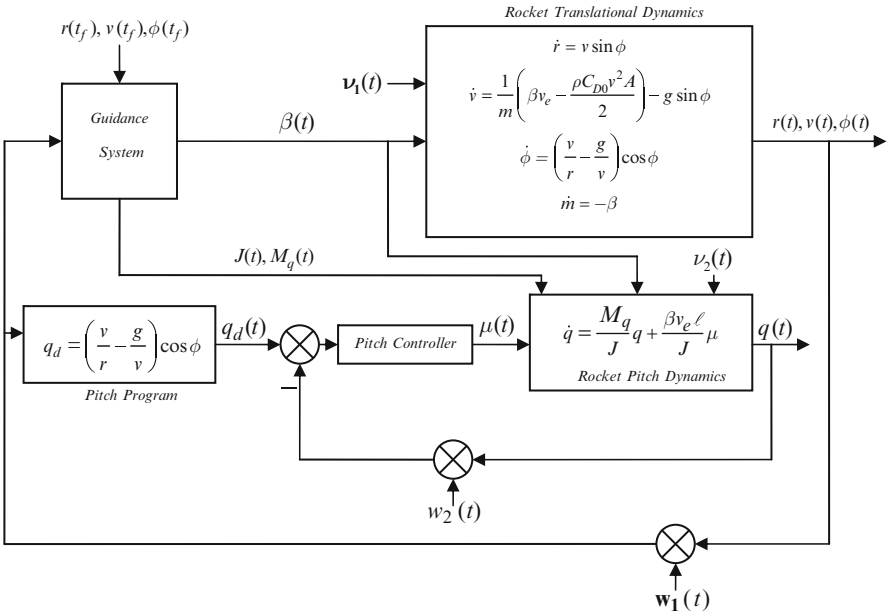


Fig. 1.24 Pitch guidance and control system of a rocket

where $J(t)$ is the pitch moment of inertia, $\ell = \ell_1 + \ell_2$, the constant distance of the rocket nozzle from the balancing reaction jet at the nose (Fig. 1.23), and $M_q(t)$ the aerodynamic parameter denoting damping in pitch that varies with both airspeed and altitude (radius). In order to fly a gravity-turn tajectory, the attitude controller must maintain a zero *angle-of-attack*, $\alpha = \phi - \theta$, at all times, which is practically achieved by tracking the following reference pitch rate generated by the guidance system:

$$q_d = \left(\frac{v}{r} - \frac{g}{v}\right)\cos\phi. \tag{1.80}$$

Clearly, the attitude control system must obey the reference pitch command, $q_d(t)$, based upon the nominal values, $\beta_d(t)$, $J(t)$, and $M_q(t)$, specified by the guidance system, as shown in Fig. 1.24. The automatic pitch rotation for tracking the guidance system command is called the *pitch program*. We shall return to the rocket guidance and control problem in Chap. 5.

1.8 Summary

A control system consists of a plant and a controller, along with actuators and sensors that are usually clubbed with the plant. Stability is the primary requirement of a control system. Controllability and observability are important properties of the plant that permit one to design a regulator and an observer, respectively. Closed-loop (automatic) control is the only practical alternative for reaching and maintaining a desired trajectory in the presence of disturbances. There are two distinct types of automatic controllers: (a) terminal controllers that take a nonlinear system to a final state in a given time and (b) tracking controllers which maintain a system close to a nominal, reference trajectory. A tracking controller generally consists of an observer, a regulator, and a feedforward controller. While most plants are governed by nonlinear state equations, they can be linearized about a nominal, reference trajectory, which is a particular solution of the plant's state equations. Linearized state equations are invaluable in designing practical tracking systems as they allow us to make use of linear systems theory, which easily lends itself to solution by transition matrix as well as provides systematic stability criteria. Automatic flight control generally involves a guidance of the vehicle's center of mass position, as well as control of vehicle's attitude by rotating it about the center of mass. Because of its shorter time-scale, the attitude control system is typically subservient to the guidance system in most aerospace vehicles.

Exercises

1.1. Consider a forced *van der Pol oscillator* with the following governing differential equation:

$$\ddot{y} + y + \left(y^2 - c\right)\dot{y} = u,$$

where c is a positive, real constant, and overdot represents derivative with respect to time.

(a) Express the system with output $y(t)$ and input $u(t)$ in a state-space form. What is the order of the system?
(b) What are the equilibrium points of the system?
(c) Write a computer program for integrating the unforced state equations [$u(t) = 0$] by a Runge-Kutta method [19], given c and the initial condition, $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$. (Hint: Try MATLAB's solver *ode45.m*.)
(d) Compute and plot the initial response, $y(t)$, with $c = 1$, $y_0 = 0$, $\dot{y}_0 = -0.01$. Also, plot the phase-plane portrait, $\dot{y}(t)$ vs. $y(t)$.
(e) Examine the stability of the unforced system at the equilibrium point, $y = 0, \dot{y} = 0$.
(f) Devise a feedback control law for linearizing the system and making it asymptotically stable at the equilibrium point, $y = 0, \dot{y} = 0$.

1.2. Consider the planar flight of a spacecraft with a low-thrust, ion propulsion engine described by the following differential equations in polar coordinates, (r, θ):

$$\begin{aligned} \ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} &= \frac{T}{m}\sin\alpha \\ r\ddot{\theta} + 2\dot{r}\dot{\theta} &= \frac{T}{m}\cos\alpha \\ \dot{m} &= -\frac{T}{v_e}, \end{aligned}$$

where T (thrust), μ (gravitational constant), and v_e (ion exhaust speed) are positive, real constants.

(a) Express the system in a state-space form with the outputs $y_1 = \dot{r}$, $y_2 = r\dot{\theta}$, and the applied inputs, $u_1 = \alpha$ (planar thrust direction), and $u_2 = T$. What is the order of the system?
(b) For a constant thrust, we have

$$m = m_0 - \frac{T}{v_e}t.$$

Beginning from an initial circular orbit of radius, r_0, we obtain the following approximate trajectory for a small thrust magnitude, $T \ll m\mu/r^2$, and a tangential thrust, $\alpha = 0$:

$$\frac{1}{\sqrt{r}} = \frac{1}{\sqrt{r_0}} - \frac{v_e}{\sqrt{\mu}}\log\left(\frac{m_0}{m_0 - \frac{T}{v_e}t}\right)$$

Determine the shape of the approximate trajectory, $r(\theta)$.

(c) For the NASA spacecraft *Deep Space-I* with $m_0 = 480$ kg, $T = 0.080$ N, $v_e = 31{,}400$ m/s, plot the trajectory $r(t)$ vs. $\theta(t)$ for 15,000 h around Sun ($\mu = 1.3271244 \times 10^{20}$ m^3/s^2) with an initial radius, $r_0 = 1.0321 A.U. = 1.544 \times 10^{11}$ m. (Hint: Plot the radius in A.U. (1.49598×10^{11} m), using the time unit of a mean solar day (24 h)).
(d) Linearize the system about the nominal trajectory derived in Part(b) and examine the stability of the unforced system.

1.3. For the spacecraft of Exercise 1.2, the following nonlinear feedback control law is to be employed:

$$\frac{T}{m}\sin\alpha = \frac{q\sqrt{\mu p}}{2r^2}\dot{r}\ ; \qquad \frac{T}{m}\cos\alpha = \frac{q\sqrt{\mu p}}{2r}\dot{\theta},$$

where $p = r_0 e^{q\theta}$ and $q \ll 1$ is a positive constant for an orbit beginning from initial condition, $r(0) = r_0$, $\theta(0) = 0$, which results in the following nominal trajectory [14]:,

$$r = \frac{r_0(1+q^2)e^{q\theta}}{1+q^2 e^{q\theta}\cos\theta}.$$

(a) Plot the radius, r, and the two control inputs for $q = 0.01$ in the range $0 \le \theta \le 7\pi$.
(b) Linearize the closed-loop system about the nominal trajectory.
(c) Is the closed-loop system stable?

1.4. Consider the flight of a lifting re-entry vehicle in the vertical plane with the following governing differential equations:

$$\begin{aligned}
\dot{h} &= v\sin\phi \\
\dot{v} &= -\frac{S}{2m}\rho_0 e^{-h/H}(C_{D0} + KC_L^2)v^2 - g_0\frac{R_0^2}{(R_0+h)^2}\sin\phi \\
\dot{\phi} &= \frac{S}{2m}\rho_0 e^{-h/H}C_L v + \left[\frac{v}{r} - g_0\frac{R_0^2}{v(R_0+h)^2}\right]\cos\phi,
\end{aligned}$$

where $m, S, g_0, R_0, \rho_0, H, C_{D0}$ are positive, real constants, and overdot represents derivative with respect to time.

(a) Express the system in a state-space form with the outputs $y_1(t) = h(t)$, $y_2(t) = v(t)$, and the applied input, $u(t) = C_L(t)$ (lift-coefficient). What is the order of the system?
(b) Given a nominal trajectory, $h_d(t), v_d(t), \phi_d(t)$ and the corresponding control history, $C_{Ld}(t)$, linearize the system assuming small perturbations from the

nominal. Show that the state-space coefficient matrices of the linearized system are the following:

$$\mathsf{A} = \begin{pmatrix} 0 & \sin\phi_d & v_d\cos\phi_d \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$\mathsf{B} = \frac{S}{m}\rho_0 e^{-h_d/H} v_d \begin{pmatrix} 0 \\ -Kv_d \\ \frac{1}{2} \end{pmatrix}$$

$$\mathsf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}; \qquad \mathsf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

where

$$a_{21} = \frac{\rho_0 S}{2mH} e^{-h_d/H}(C_{\mathrm{D}0} + KC_{Ld}^2)v_d^2 + 2g_0\frac{R_0^2}{(R_0+h_d)^3}\sin\phi_d$$

$$a_{22} = \frac{\rho_0 S}{m} e^{-h_d/H}(C_{\mathrm{D}0} + KC_{Ld}^2)v_d$$

$$a_{23} = -g_0\frac{R_0^2}{(R_0+h_d)^2}\cos\phi_d$$

$$a_{31} = -\frac{\rho_0 S}{2mH} e^{-h_d/H} C_{Ld} v_d + \left[-\frac{v_d}{(R_0+h_d)^2} + 2g_0\frac{R_0^2}{v_d(R_0+h_d)^3}\right]\cos\phi_d$$

$$a_{32} = \frac{\rho_0 S}{2m} e^{-h_d/H} C_{Ld} + \left[\frac{1}{(R_0+h_d)} + g_0\frac{R_0^2}{v_d^2(R_0+h_d)^2}\right]\cos\phi_d$$

$$a_{33} = -\left[\frac{v_d}{(R_0+h_d)} - g_0\frac{R_0^2}{v_d(R_0+h_d)^2}\right]\sin\phi_d .$$

(c) Under what conditions is the unforced system stable about the nominal trajectory.
(d) Is the linearized system controllable?
(e) Is the linearized system observable?

1.5. Consider the flight of a rocket in a vertical plane represented by the two subsystem blocks in Fig. 1.25. The thrust, T, is a constant, and the thrust deflection angles, μ, ϵ, as well as the fin deflection angle, δ_{f}, are the control inputs. The angular velocity vector of the rocket, $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^{\mathrm{T}}$, arises because of the rotational motion about its own center of mass, o. The other flight variables are the same as those discussed in Example 1.3.

(a) What is the order of the system, and what could be considered its state variables?

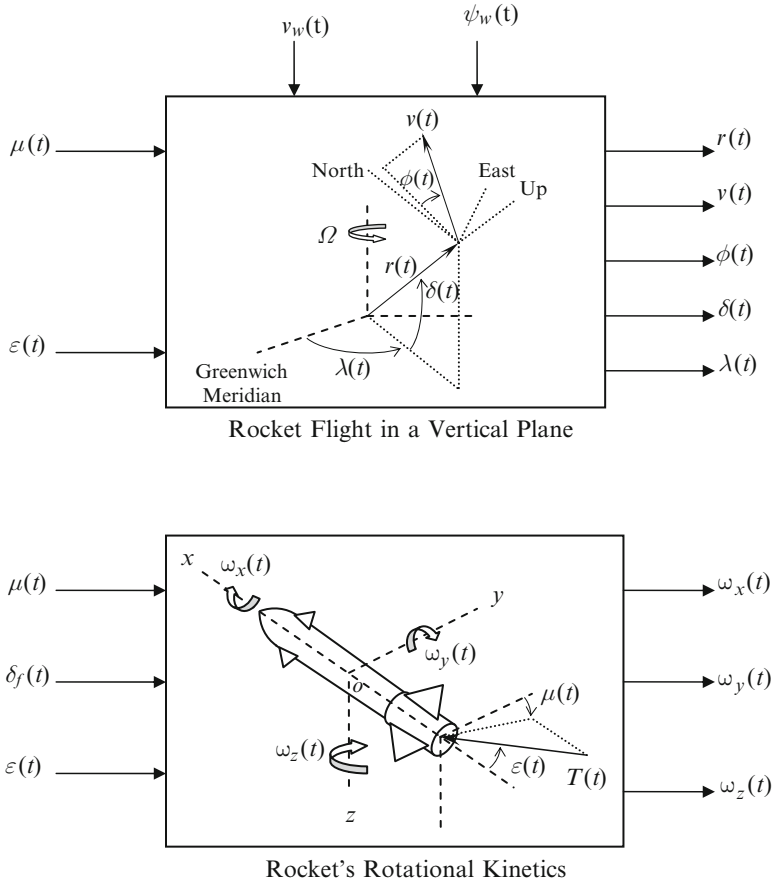


Fig. 1.25 Flight of a rocket in the vertical plane

(b) Suppose the applied control inputs are all zeros. In that case, is there any coupling between the two subsystems?
(c) If the rocket is flying exactly vertically ($\phi = \pi/2, \omega_y = \omega_z = 0$), what are the new state variables and the order of the system?

1.6. If in addition to the rocket's flight in a vertical plane and its rotational kinetics discussed in Exercise 1.5, it is also necessary to describe its orientation relative to a fixed frame, *SXYZ*, as shown in Fig. 1.26:

(a) What is the new order of the system?
(b) What are the state variables and control inputs of the rotational kinematics subsystem?
(c) Is the rocket's orientation important for determining its position in space?
(d) Does the rocket's orientation influence its angular velocity?

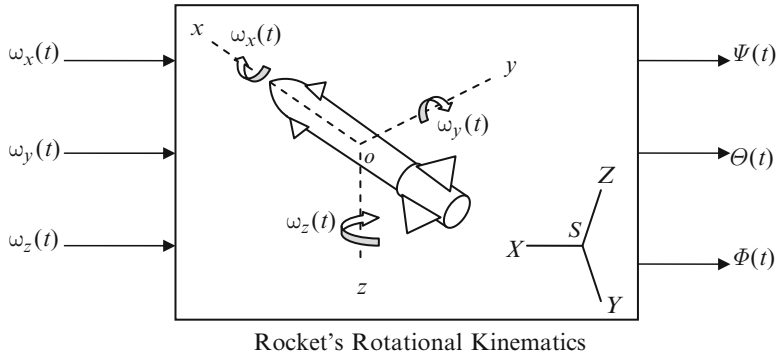


Fig. 1.26 Rotational kinematics of a rocket

1.7. Derive the trajectory solution corresponding to $t_f < a$ for the terminal control problem of Example 1.10.

1.8. A system has the following state equations:

$$\begin{aligned}\dot{x}_1 &= -2x_1 - 2x_2 + u_1\\ \dot{x}_2 &= x_3 + u_2\\ \dot{x}_3 &= -3x_2 - 4x_3 + u_1 + u_2.\end{aligned}$$

(a) If the initial condition at $t = 0$ is $x_1(0) = 10$, $x_2(0) = 5$, $x_3(0) = 2$ and $u(t) = (t, 1)^T$, solve the state equations.
(b) Is the system controllable?

1.9. For a system with the following state-space coefficient matrices:

$$\mathsf{A} = \begin{pmatrix} -2 & -3 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}; \qquad \mathsf{B} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

(a) Determine the response to zero initial condition and a unit impulse function applied at $t = 0$ as the input.
(b) Analyze the stability and controllability of the system.

1.10. An aircraft's longitudinal motion (Chap. 4) is governed by the following linear time-invariant, state-space coefficient matrices:

$$\mathsf{A} = \begin{pmatrix} 0.0007 & 0.0046 & 0 & -9.678 \\ -0.0687 & -0.2953 & 174.87 & 0 \\ 0.0017 & 0.0005 & -0.4462 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \qquad \mathsf{B} = \begin{pmatrix} 1.041 \\ -6.294 \\ -4.889 \\ 0 \end{pmatrix}.$$

(a) Analyze the stability of the system using the MATLAB command *eig* for eigenvalues of a matrix.
(b) Is the system controllable?
(c) If the sole output is the first state variable of the system, is the system observable?
(d) If the output vector consists of both the second and the third state variables, is the system observable?

You can use the MATLAB-CST command *ctrb* for controllability and observability tests.

Chapter 2
Flight Dynamic Models

2.1 Aims and Objectives

- To present the basic flight dynamics modeling approach applicable to aircraft, rockets, and spacecraft
- To derive the equations of motion for typical flight dynamic systems
- To describe the sensors used in flight control systems

2.2 Rigid Body Dynamics

The word "dynamics" refers to the motion of a body, which can be completely described by velocity (kinematics) and acceleration (kinetics). Kinematical equations of translation and rotation are derived using the geometry of motion. When Newton's laws are applied, we obtain kinetic equations for both translation and rotation. Considering the vehicle as a body (collection of a large number of particles) of mass m acted upon by external force $\mathbf{F}(t)$, we apply Newton's second law of motion relative to an *inertial frame* of reference $(o_G x_G y_G z_G)$ (Fig. 2.1) to individual particles of mass δm and take a sum over all particles:

$$\sum \delta \mathbf{F} = \frac{\mathrm{d}}{\mathrm{d}t} \sum \mathbf{v} \delta m. \tag{2.1}$$

Due to Newton's third law of motion, the internal forces acting between any two particles cancel out. Consequently, the net force on the body is the same as the external force $\mathbf{F}(t)$. Furthermore, the instantaneous velocity of a point o attached to the body is given as $\mathbf{v} = \frac{\mathrm{d}(\mathbf{r})}{\mathrm{d}t}$, where $\mathbf{r}(t)$ is the instantaneous position of o. There is a special point associated with every body, called the *center of mass*, such that the

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_2, © Springer Science+Business Media, LLC 2011

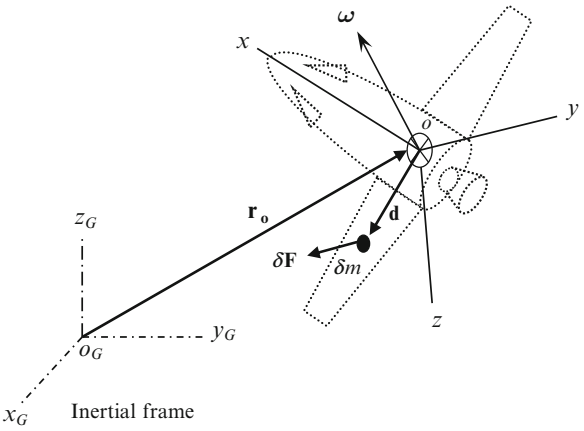


Fig. 2.1 Rotation and translation of a flight vehicle

mass moment, $\sum \mathbf{d}\delta m$ vanishes about that point (Fig. 2.1). If we choose the point o to be the center of mass, then the governing translational kinetics equation, (2.1), can be written as follows:

$$\mathbf{F} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}. \tag{2.2}$$

A generic flight vehicle's mass m changes with time, either due to burning and ejection of a propellant mass, or addition/subtraction of modules (e.g., expenditure of munitions, docking of two spacecraft, etc.). Aerospace vehicles powered by rocket engines and those involving addition/subtraction of modules have a large instantaneous rate of change of mass, thereby generating a reactive force. For all other flight vehicles, the instantaneous rate of change of mass is relatively small and the vehicle's mass can be assumed constant over short portions of flight. When applying (2.2) to a variable mass vehicle, it must be ensured that the external force $\mathbf{F}$ includes all reactive forces due to mass ejection/accretion. It is clear from (2.2) that for the limited purpose of studying the translational motion of a body, we can regard the body as if all its mass were concentrated at the center of mass, and the net external force were applied at the center of mass. This forms the basis for the idealization of the body as a particle.

In flight control applications, it is more useful to resolve the force $\mathbf{F}$ and the velocity $\mathbf{v}$ in a *body-fixed* frame ($oxyz$) – a coordinate frame rigidly fixed to the body (Fig. 2.1) – rather than the inertial frame ($o_G x_G y_G z_G$). In that case, (2.2) becomes

$$\mathbf{F} = m\frac{\partial \mathbf{v}}{\partial t} + m(\boldsymbol{\omega} \times \mathbf{v}), \tag{2.3}$$

where the partial derivative denotes time derivative relative to the body-fixed frame and $\boldsymbol{\omega}$ is the *angular velocity* of the body (same as the angular velocity of the frame

($oxyz$)). For convenience of notation, we shall replace the partial time derivative with an overdot and write (2.3) as follows:[1]

$$\mathbf{F} = m\dot{\mathbf{v}} + m(\boldsymbol{\omega} \times \mathbf{v}). \tag{2.4}$$

The angular velocity is not a free variable, but must satisfy the governing equation of *rotational kinetics* obtained by taking moments of (2.1) about the center of mass, o:

$$\boldsymbol{\tau} = \sum \left(\mathbf{d} \times \delta m \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} \right) = \int \left(\mathbf{d} \times \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} \right) \mathrm{d}m, \tag{2.5}$$

where $\mathbf{d}$ is the position of the elemental mass δm relative to o (Fig. 2.1), $\bar{\mathbf{v}} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{d}$ the velocity of the elemental mass, and $\boldsymbol{\tau} \doteq \sum(\mathbf{d} \times \delta\mathbf{F})$ is the net *external torque* about o. Note that (2.5) involves a replacement of the summation over particles by an integration over mass in the limit $\delta m \to 0$, and also implies mutual cancellation of the internal torques. As in the case of translational dynamics, we require that $\boldsymbol{\tau}$ includes all reactive torques arising because of mass ejection/accretion from the vehicle. Equation (2.5) arises due to the fact that all the external forces may not pass through the center of mass, o, and constitutes a departure from particle idealization. Equation (2.5) can be rewritten in terms of the *angular momentum*, which is defined for the elemental mass by $\delta\mathbf{H} \doteq \mathbf{d} \times \delta m \bar{\mathbf{v}}$, and for the body by the following equation:

$$\mathbf{H} = \int \mathbf{d} \times \bar{\mathbf{v}} \mathrm{d}m. \tag{2.6}$$

If we assume the flight vehicle to be a *rigid body*,[2] we have

$$\mathbf{H} = \int \mathbf{d} \times (\boldsymbol{\omega} \times \mathbf{d}) \mathrm{d}m. \tag{2.7}$$

Substitution of (2.7) into (2.5) yields the following equation governing the rotational kinetics:

$$\frac{\mathrm{d}\mathbf{H}}{\mathrm{d}t} = \boldsymbol{\tau}. \tag{2.8}$$

The inherent coupling of rotational and translational kinetics is evident in (2.2) and (2.8), where $\boldsymbol{\omega}$ is common to both, and is obtained from the solution to the rotational kinetics equation, (2.8). Furthermore – as in all atmospheric flight vehicles and some spacecraft – if the external force vector depends on the vehicle's attitude (rotational variables), and the external torque vector depends upon position and velocity (translational variables), then a further coupling takes place between the rotation and the translation. When the time scale of rotation is much smaller

[1] The time derivative relative to a rotating frame is given by the chain rule $\dot{\mathbf{A}} = \frac{\partial \mathbf{A}}{\partial t} = \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} - (\boldsymbol{\omega} \times \mathbf{A})$.

[2] The distance between any two points in a rigid body does not vary with time.

than that of translation (as in most spacecraft), then the two can be effectively decoupled. In such a case, the instantaneous rotational parameters are treated as inputs to the translatory motion, and the position and velocity are treated as almost constant parameters for the rotary motion. However, the decoupling of rotational and translational motions is not a good approximation either for a high performance aircraft or a missile, which would require a simultaneous solution of the two sets of equations ((2.2) and (2.8)), along with the kinematical equations discussed later.

In order to derive an expression for the angular momentum for use in the rotational kinetics equation (2.8), we resolve all the vectors in the body-fixed frame ($oxyz$) as follows:

$$\mathbf{d} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{2.9}$$

$$\boldsymbol{\omega} = \omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k} \tag{2.10}$$

$$\mathbf{H} = H_x\mathbf{i} + H_y\mathbf{j} + H_z\mathbf{k} \tag{2.11}$$

$$\boldsymbol{\tau} = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}. \tag{2.12}$$

On substituting these into (2.7) we obtain

$$\mathbf{H} = \mathsf{J}\boldsymbol{\omega}, \tag{2.13}$$

where J is a constant, symmetric matrix called *inertia tensor*, given by:

$$\mathsf{J} \doteq \begin{pmatrix} \int (y^2 + z^2)\mathrm{d}m & -\int xy\mathrm{d}m & -\int xz\mathrm{d}m \\ -\int xy\mathrm{d}m & \int (x^2 + z^2)\mathrm{d}m & -\int yz\mathrm{d}m \\ -\int xz\mathrm{d}m & -\int yz\mathrm{d}m & \int (x^2 + y^2)\mathrm{d}m \end{pmatrix}. \tag{2.14}$$

In terms of its components, J is written as follows:

$$\mathsf{J} \doteq \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{pmatrix}. \tag{2.15}$$

The components of the inertia tensor are divided into the *moments of inertia*, J_{xx}, J_{yy}, J_{zz}, and the *products of inertia*, J_{xy}, J_{yz}, J_{xz}. As the inertia tensor accounts for the mass distribution of the body, it is clear from (2.13) that the way mass is distributed in a vehicle is crucial to rotational flight dynamics.

The final step in our derivation of rotational kinetics equations is to express the inertial time derivative of (2.13) relative to the rotating body-fixed frame as follows:

$$\boldsymbol{\tau} = \mathsf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathsf{J}\boldsymbol{\omega}), \tag{2.16}$$

where, as previously, the overdot represents the partial time derivative taken with reference to the body-fixed frame,

$$\dot{\boldsymbol{\omega}} \doteq \begin{Bmatrix} \dfrac{\mathrm{d}\omega_x}{\mathrm{d}t} \\ \dfrac{\mathrm{d}\omega_y}{\mathrm{d}t} \\ \dfrac{\mathrm{d}\omega_z}{\mathrm{d}t} \end{Bmatrix} = \begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{Bmatrix}. \tag{2.17}$$

By replacing the vector product in (2.16) by a matrix product, we have

$$\boldsymbol{\tau} = \mathsf{J}\dot{\boldsymbol{\omega}} + \mathsf{S}(\boldsymbol{\omega})\mathsf{J}\boldsymbol{\omega}, \tag{2.18}$$

where

$$\mathsf{S}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \tag{2.19}$$

is a *skew-symmetric* matrix. Equation (2.18) represents three scalar, coupled, nonlinear, ordinary differential equations, and are called *Euler's equations* of rotational kinetics, whose solution gives the angular velocity, $\boldsymbol{\omega}$, at a given time.

2.3 Attitude Kinematics

In many cases, the rotation of a flight vehicle is important in itself for various reasons (aerodynamics, pointing of weapons, payload, or antennas, etc.), and governs the intantaneous *attitude* (orientation) of the vehicle. The vehicle's attitude is given by the orientation of the body-fixed frame (*oxyz*) relative to a reference frame (*oXYZ*). The time dependence of a frame's orientation relative to another frame is termed *rotational kinematics*, and can be described using the coordinate transformation between the two frames.

Let us represent the axes *ox, oy, oz* of the rotating frame, (*oxyz*), by the unit vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$, respectively. The orientation of (*oxyz*) relative to the reference frame (*oXYZ*) – whose axes *oX, oY, oZ* are represented by the unit triad $\mathbf{I}, \mathbf{J}, \mathbf{K}$ – is given by the following coordinate transformation:

$$\begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix} = \mathsf{C} \begin{Bmatrix} \mathbf{I} \\ \mathbf{J} \\ \mathbf{K} \end{Bmatrix}, \tag{2.20}$$

where C is called the *rotation matrix*. A rotation matrix is *orthogonal*, i.e., it satisfies $\mathsf{C}^{\mathrm{T}}\mathsf{C} = \mathsf{C}\mathsf{C}^{\mathrm{T}} = \mathsf{I}$. Furthermore, the determinant of a rotation matrix is unity. Associated with the rotation matrix is the concept of *Euler axis* and the *principal angle*. *Euler's theorem* states that the relative orientation of any pair of coordinate frames is uniquely determined by a rotation by angle, Φ, called the principal angle about a fixed axis through the common origin of the two frames, called Euler axis and denoted by the unit vector $\mathbf{e}$. As Euler axis is invariant under coordinate frame rotation, it can be shown that $\mathbf{e}$ is the eigenvector of C corresponding to a unity eigenvalue, and hence satisfies $\mathsf{C}\mathbf{e} = \mathbf{e}$. Similarly, it can be also shown that the principal angle satisfies the equation $\cos\Phi = \frac{1}{2}(trace\mathsf{C} - 1)$.

The rotational kinematics equation can be derived by differentiating with time both the sides of (2.20), thereby producing

$$\begin{Bmatrix} \boldsymbol{\omega} \times \mathbf{i} \\ \boldsymbol{\omega} \times \mathbf{j} \\ \boldsymbol{\omega} \times \mathbf{k} \end{Bmatrix} = \frac{\mathrm{d}\mathsf{C}}{\mathrm{d}t} \begin{Bmatrix} \mathbf{I} \\ \mathbf{J} \\ \mathbf{K} \end{Bmatrix}, \tag{2.21}$$

or, using the matrix product of (2.18), we have

$$-\mathsf{S}(\boldsymbol{\omega}) \begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix} = \frac{\mathrm{d}\mathsf{C}}{\mathrm{d}t}\mathsf{C}^{\mathrm{T}} \begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix}, \tag{2.22}$$

from which (along with the orthogonality of C) it follows that

$$\frac{\mathrm{d}\mathsf{C}}{\mathrm{d}t} = -\mathsf{S}(\boldsymbol{\omega})\mathsf{C}(t). \tag{2.23}$$

Equation (2.23) is the governing equation of attitude kinematics. Because the rotation matrix corresponding to a particular attitude can be represented by several alternative kinematic parameters (such as *Euler angles* and *quaternion* to be considered later), the attitude kinematics equation can be expressed in terms of various parameters. It is interesting to note the following definition of the angular velocity of a frame in terms of Euler axis and incremental principal angle, $\Delta\Phi$

$$\boldsymbol{\omega}(t) \doteq \lim_{\Delta t \to 0} \frac{\Delta\Phi\mathbf{e}}{\Delta t}. \tag{2.24}$$

As noted earlier, one can employ various alternative descriptions of the rotation matrix, called attitude representations. We will consider next the attitude representations commonly used in flight control applications – namely, Euler angles and quaternion. Each attitude representation has its advantages and disadvantages, and is suited to a particular flight situation. For example, Euler angles are best suited to small rotations of flight vehicles about their equilibrium orientation, while quaternion can handle large, multiaxis rotations that are often encountered in extreme flight maneuvers.

2.3.1 Euler Angles

Euler angles are based on the fact that a general orientation of a coordinate frame can be obtained by using successive rotations about the axes of the frame. Rotation of a frame about one of its axes is called an *elementary rotation*. The largest number of such rotations needed to uniquely specify a given orientation, called rotational *degrees of freedom*, is three. Hence, we can employ three angles, each about a particular coordinate axis, to describe a given orientation. Such a representation of the attitude by three elementary rotations is called an Euler angle representation, and the concerned angles are known as Euler angles.

An *anticlockwise* rotation is considered positive by the right-hand rule. A positive elementary rotation of the frame $(oXYZ)$ about oX by angle Φ is given by the rotation matrix

$$\mathsf{C}_1 \doteq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{pmatrix}, \tag{2.25}$$

whereas, the rotation matrix for a positive rotation about oY by the same angle is

$$\mathsf{C}_2 \doteq \begin{pmatrix} \cos\Phi & 0 & -\sin\Phi \\ 0 & 1 & 0 \\ \sin\Phi & 0 & \cos\Phi \end{pmatrix}. \tag{2.26}$$

Similarly, a positive rotation about oZ by Φ is given by:

$$\mathsf{C}_3 \doteq \begin{pmatrix} \cos\Phi & \sin\Phi & 0 \\ -\sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.27}$$

One can easily verify the orthogonality of the rotation matrix for each of the elementary rotations.

The sequence of elementary rotations is of utmost importance in the Euler angle representation. We can specify the Euler angles and the axes of elementary rotations using notation such as $(\Psi)_3, (\Theta)_2, (\sigma)_1$ shown in the right diagram of Fig. 2.2. This particular representation consists of a rotation of $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ by angle Ψ about $\mathbf{K}$, resulting in the intermediate orientation, $(\mathbf{I}', \mathbf{J}', \mathbf{K})$, followed by a rotation by angle Θ about $\mathbf{J}'$, thereby producing $(\mathbf{I}'', \mathbf{J}', \mathbf{K}'')$. Finally, a rotation by angle σ about $\mathbf{I}''$, yields the desired attitude, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. The $(\Psi)_3, (\Theta)_2, (\sigma)_1$ representation is commonly used for describing an aircraft's attitude relative to a local horizontal frame using the Euler angles, yaw (Ψ), pitch (Θ), and roll (σ). An alternative Euler representation employed in spacecraft orientation is the Euler sequence $(\Psi)_3, (\Theta)_1, (\sigma)_3$, shown on the left side of Fig. 2.2. It is clear from Fig. 2.2 that the Euler representations are not unique (i.e., different Euler angle sequences can be employed to represent the same attitude).

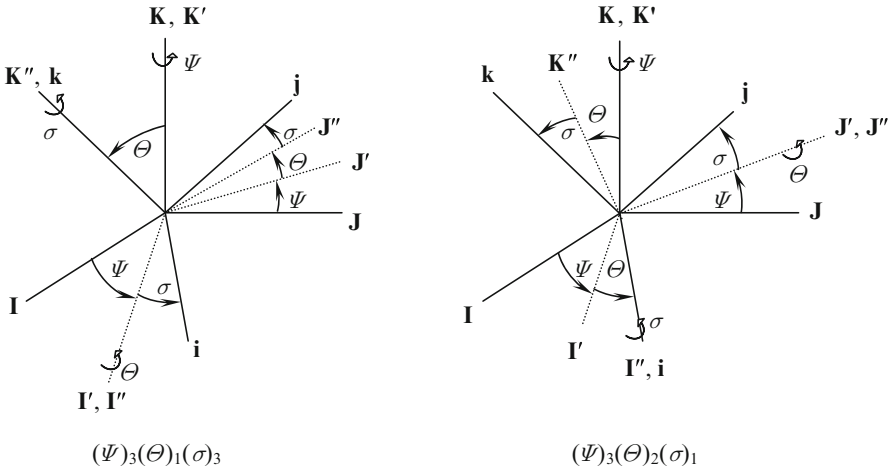


Fig. 2.2 The Euler angle representations, $(\Psi)_3, (\Theta)_1, (\sigma)_3$ and $(\Psi)_3, (\Theta)_2, (\sigma)_1$

The rotation matrix relating the final orientation of *(oxyz)* to *(oXYZ)* for the Euler angle representation $(\Psi)_3, (\Theta)_1, (\sigma)_3$ is given by:

$$\begin{aligned} \mathsf{C} &= \mathsf{C}_3(\sigma)\mathsf{C}_1(\Theta)\mathsf{C}_3(\Psi) \\ &= \begin{pmatrix} (\cos\Psi\cos\sigma - \sin\Psi\sin\sigma\cos\Theta) & (\sin\Psi\cos\sigma + \cos\Psi\sin\sigma\cos\Theta) & \sin\sigma\sin\Theta \\ -(\cos\Psi\sin\sigma + \sin\Psi\cos\sigma\cos\Theta) & (-\sin\Psi\sin\sigma + \cos\Psi\cos\sigma\cos\Theta) & \cos\sigma\sin\Theta \\ \sin\Psi\sin\Theta & -\cos\Psi\sin\Theta & \cos\Theta \end{pmatrix} \end{aligned} \tag{2.28}$$

Similarly, the rotation matrix for the orientation $(\Psi)_3, (\Theta)_2, (\sigma)_1$ is the following:

$$\begin{aligned} \mathsf{C} &= \mathsf{C}_1(\sigma)\mathsf{C}_2(\Theta)\mathsf{C}_3(\Psi) \\ &= \begin{pmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ (\sin\sigma\sin\Theta\cos\Psi - \cos\sigma\sin\Psi) & (\sin\sigma\sin\Theta\sin\Psi + \cos\sigma\cos\Psi) & \sin\sigma\cos\Theta \\ (\cos\sigma\sin\Theta\cos\Psi + \sin\sigma\sin\Psi) & (\cos\sigma\sin\Theta\sin\Psi - \sin\sigma\cos\Psi) & \cos\sigma\cos\Theta \end{pmatrix} \end{aligned} \tag{2.29}$$

Note that the rotation matrix given by (2.28) becomes the following when $\sin\Theta = 0$:

$$\mathsf{C} = \begin{pmatrix} \cos(\sigma \pm \Psi) & \sin(\sigma \pm \Psi) & 0 \\ \mp\sin(\sigma \pm \Psi) & \pm\cos(\sigma \pm \Psi) & 0 \\ 0 & 0 & \pm 1 \end{pmatrix}, \tag{2.30}$$

which shows that the angles σ and Ψ cannot be resolved from the rotation matrix, but only their sum (or difference) can be determined. Such special orientations for which Euler angles become ambiguous are called *singularities*. Thus, the representation $(\Psi)_3, (\Theta)_1, (\sigma)_3$ has singularities at $\Theta = n\pi$, $(n = 0, 1, 2, 3, \ldots)$, and the representation is thus limited to rotations of $0 \leq \Theta < \pi$ that generally apply to spacecraft attitudes. Similarly, it can be shown that the Euler angle representation $(\Psi)_3, (\Theta)_2, (\sigma)_1$ has singularities at $\Theta = n\frac{\pi}{2}$, $(n = 0, 1, 3, \ldots)$, which limits the utility of the representation to rotations involving $\pi/2 \leq \Theta < \pi/2$. From the singularities and the resulting domain of applicability of a particular Euler angle representation one can derive its application. In any case, Euler angles cannot be used for rotations of principal angle greater than 180°. For attitudes that do not lead to singularities in an Euler angle representation, the three angles can be unambiguously determined from the rotation matrix.

The differential equations governing attitude kinematics can be easily derived for Euler angles by employing the appropriate rotation matrix and expressing the angular velocity in terms of the angular rates. For example, using the $(\Psi)_3, (\Theta)_1, (\sigma)_3$ representation we can write the angular velocity, $\boldsymbol{\omega}(t)$, as follows:

$$\boldsymbol{\omega}(t) = \dot{\sigma}\mathbf{k} + \dot{\Theta}\mathbf{I}' + \dot{\Psi}\mathbf{K}, \tag{2.31}$$

which, considering the elementary rotations involved in (2.28), becomes

$$\boldsymbol{\omega}(t) = \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = \begin{Bmatrix} \dot{\Psi}\sin\sigma\sin\Theta + \dot{\Theta}\cos\sigma \\ \dot{\Psi}\cos\sigma\sin\Theta - \dot{\Theta}\sin\sigma \\ \dot{\Psi}\cos\Theta + \dot{\sigma} \end{Bmatrix}, \tag{2.32}$$

or,

$$\begin{Bmatrix} \dot{\Psi} \\ \dot{\Theta} \\ \dot{\sigma} \end{Bmatrix} = \frac{1}{\sin\Theta} \begin{pmatrix} \sin\sigma & \cos\sigma & 0 \\ \cos\sigma\sin\Theta & -\sin\sigma\sin\Theta & 0 \\ -\sin\sigma\cos\Theta & -\cos\sigma\cos\Theta & \sin\Theta \end{pmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}. \tag{2.33}$$

Equation (2.33) is the required set of kinematical equations, and clearly indicates the singularities of the particular representation for $\sin\Theta = 0$.

2.3.2 *Quaternion*

Any attitude representation based upon only three kinematic parameters (such as Euler angles) is bound to be singular. For a universal application, we desire a nonsingular representation. It can be easily shown that the four parameter system of Euler axis, $\mathbf{e}$, and principal angle, Φ, is one such nonsingular attitude representation. Another four parameter, nonsingular attitude representation that is closely related to Euler axis/principal angle combination is the *quaternion* representation. The attitude

quaternion is a special set composed of four mutually dependent scalar parameters, q_1, q_2, q_3, q_4, such that the first three form a vector, called the *vector part*,

$$\mathbf{q} \doteq \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix}, \tag{2.34}$$

and the fourth, q_4, represents the *scalar part*. The quaternion for attitude representation can be derived from the Euler axis, $\mathbf{e}$, and principal rotation angle, Φ, as follows:

$$\begin{aligned} q_i &\doteq e_i \sin\frac{\Phi}{2} \quad (i = 1, 2, 3) \\ q_4 &\doteq \cos\frac{\Phi}{2}. \end{aligned} \tag{2.35}$$

It is clear from (2.35) that q_1, q_2, q_3, q_4, must satisfy the constraint equation,

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{2.36}$$

This constraint implies that the quaternion yields only three independent, scalar parameters, as in the principal angle/Euler axis, or the Euler angle representations. As the four elements of the quaternion satisfy the constraint equation, (2.36), it can be said that attitude orientations vary along the surface of a *four-dimensional* unit sphere without any singularity. The chief advantage of the quaternion over the principal angle/Euler axis combination (also a nonsingular representation) lies in that the former does not require computationally intensive trigonometric function evaluations when derived from the rotation matrix.

The rotation matrix, C, can be written in terms of the quaternion as follows:

$$\mathsf{C} = (q_4^2 - \mathbf{q}^{\mathrm{T}}\mathbf{q})\mathsf{I} + 2\mathbf{q}\mathbf{q}^{\mathrm{T}} - 2q_4\mathsf{S}(\mathbf{q}), \tag{2.37}$$

where $\mathsf{S}(\mathbf{q})$ is the following skew-symmetric matrix function formed out of the elements of vector $\mathbf{q}$:

$$\mathsf{S}(\mathbf{q}) = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix}. \tag{2.38}$$

We can write (2.37) in terms of the individual quaternion elements as follows:

$$\mathsf{C} = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix}, \tag{2.39}$$

which yields the following expressions for calculating the quaternion elements from the elements of the rotation matrix, c_{ij}:

$$\begin{aligned} q_1 &= \frac{c_{23} - c_{32}}{4q_4} \\ q_2 &= \frac{c_{31} - c_{13}}{4q_4} \\ q_3 &= \frac{c_{12} - c_{21}}{4q_4}, \end{aligned} \tag{2.40}$$

where

$$q_4 = \pm\frac{1}{2}\sqrt{1 + c_{11} + c_{22} + c_{33}} = \pm\frac{1}{2}\sqrt{1 + trace\mathsf{C}}. \tag{2.41}$$

Equation (2.41) implies that the quaternion representation is nonunique (eventhough C is free from such an ambiguity). The sign ambiguity in determining the quaternion can be physically understood by the fact that a rotation by angle Φ about an axis $\mathbf{e}$ is the same as a rotation by $-\Phi$ about $-\mathbf{e}$. As there is no loss of generality in always taking a particular sign in (2.41), doing so removes the ambiguity in the quaternion representation. Of course, the derivation given earlier is valid only if $q_4 \neq 0$. If q_4 is close to zero, one can employ an alternative derivation, such as the following:

$$\begin{aligned} q_2 &= \frac{c_{12} + c_{21}}{4q_1} \\ q_3 &= \frac{c_{31} + c_{13}}{4q_1} \\ q_4 &= \frac{c_{23} - c_{32}}{4q_1}, \end{aligned} \tag{2.42}$$

where

$$q_1 = \pm\frac{1}{2}\sqrt{1 + c_{11} - c_{22} - c_{33}}. \tag{2.43}$$

Other alternative derivations of the quaternion from the rotation matrix involve a similar division by either q_2 or q_3.

The most useful feature of the quaternion representation for a frame's attitude is the rather simple *composition rule* by which successive rotations can be combined, given by:

$$\mathsf{C}(\mathbf{q}'', q_4'') = \mathsf{C}(\mathbf{q}', q_4')\mathsf{C}(\mathbf{q}, q_4), \tag{2.44}$$

or,

$$\begin{Bmatrix} q_1'' \\ q_2'' \\ q_3'' \\ q_4'' \end{Bmatrix} = \begin{pmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{pmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}. \tag{2.45}$$

The composition rule of (2.45) is the defining property of the quaternion. In fact, any four parameter set obeying (2.45) is termed the quaternion, of which the attitude quaternion of (2.35) is a special subset.

By substituting (2.35) for an infinitesimal rotation into (2.24), the attitude kinematics equation (2.23) can be expressed in terms of the quaternion as follows:

$$\frac{\mathrm{d}\{\mathbf{q}, q_4\}^{\mathrm{T}}}{\mathrm{d}t} = \frac{1}{2}\Omega\{\mathbf{q}(t), q_4(t)\}^{\mathrm{T}}, \tag{2.46}$$

where

$$\Omega = \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}. \tag{2.47}$$

The linear, algebraic form of the attitude kinematics equation, (2.47), is an obvious advantage of the quaternion representation over the nonlinear, transcendental kinematics equation (such as (2.33)) of an Euler angle representation.

The main disadvantage of the quaternion attitude representation is its redundancy, which causes an attitude to be represented by more than one quaternion and sometimes leads to practical problems when employed in a flight control system. In order to remove redundancy, the quaternion is often replaced by a closely related, three parameter set such as either the *Rodrigues* or the *modified Rodrigues* parameters. However, being three parameter representations like the Euler angles, such minimal representations suffer from inherent singularities, and thus can be applied in a limited range of principal angles.

2.4 Flight Dynamics

As noted earlier, flight dynamics consists of translation of the center of mass (2.2) and rotation of the vehicle about the center of mass (2.8). Therefore, flight dynamics can be separated into translational flight dynamics and rotational dynamics (also called *attitude dynamics*). Generally, translational flight dynamics deals with the motion of the center of mass in a reference frame at the center of a celestial body (e.g., Earth, Sun, Moon, other planets), and attitude dynamics refers to the rotation of a body-fixed frame about the center of mass of the vehicle. The attitude (or orientation) of the body-fixed frame is generally measured relative to a third coordinate frame.

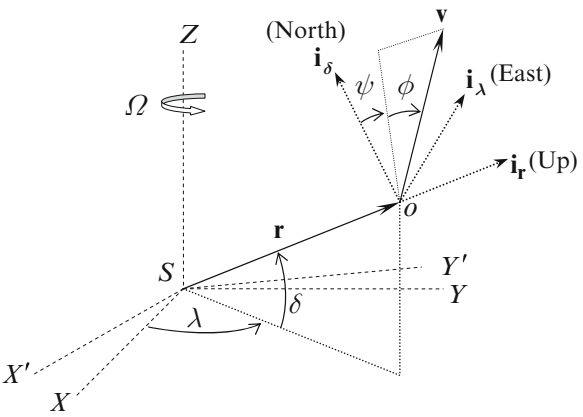


Fig. 2.3 Planet-fixed frame and the inertial reference frame

2.4.1 Translational Kinematics in Planet-Fixed Frame

For the purpose of long-range guidance and navigation, the equations of translational kinematics are resolved in a reference frame fixed to the planet, with origin at planetary center.

Consider a reference frame $(SXYZ)$ rigidly fixed to the center of a planet and rotating with it (Fig. 2.3), with the axes given by the unit vectors $\mathbf{I}, \mathbf{J}, \mathbf{K}$, respectively. The angular velocity of $(SXYZ)$ relative to a stationary frame, $(SX'Y'Z)$, at the center of the planet is given by $\boldsymbol{\Omega} = \Omega\mathbf{K}$. The instantaneous position, $\mathbf{r}(t)$, of the flight vehicle's center of mass, o, is resolved in $(SXYZ)$ using the spherical coordinates $(r(t), \lambda(t), \delta(t))$ denoting the radius, longitude, and latitude, respectively. The velocity vector is represented by another set of spherical coordinates, $(v(t), \phi(t), \psi(t))$, resolved in a moving frame called *local horizon frame* with axes $\mathbf{i}_\mathbf{r}, \mathbf{i}_\lambda, \mathbf{i}_\delta$ representing the local directions, vertical (*Up*), *East*, and *North*, respectively, as shown in Fig. 2.4. Here, $\phi(t)$ is the *flight path angle*, and $\psi(t)$ is the *heading angle* (or *velocity azimuth*). The *relative velocity*, $\mathbf{v}(t)$, is expressed as follows:

$$\mathbf{v} = v\left(\sin\phi\mathbf{i}_\mathbf{r} + \cos\phi\sin\psi\mathbf{i}_\lambda + \cos\phi\cos\psi\mathbf{i}_\delta\right), \tag{2.48}$$

and the coordinate transformation between the planet-fixed and the local horizon frames is given by Euler angle sequence $(\lambda)_3, (-\delta)_2$ as follows:

$$\begin{Bmatrix} \mathbf{i}_\mathbf{r} \\ \mathbf{i}_\lambda \\ \mathbf{i}_\delta \end{Bmatrix} = \mathsf{C}_{\mathsf{LH}} \begin{Bmatrix} \mathbf{I} \\ \mathbf{J} \\ \mathbf{K} \end{Bmatrix}, \tag{2.49}$$

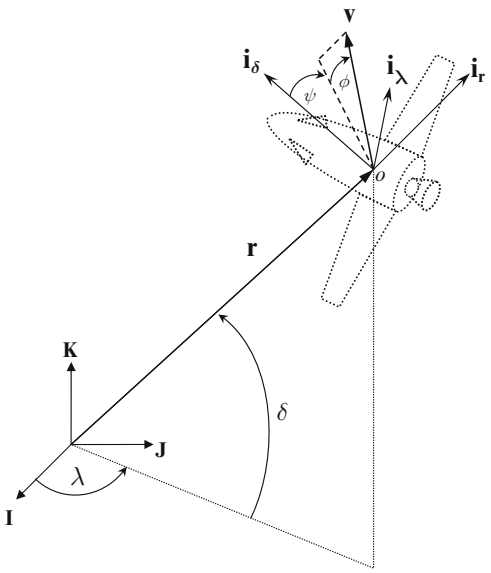


Fig. 2.4 The planet-fixed frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, and the local horizon frame, $(\mathbf{i}_\mathbf{r}, \mathbf{i}_{\boldsymbol{\lambda}}, \mathbf{i}_{\boldsymbol{\delta}})$

where

$$\mathsf{C}_{\mathsf{LH}} = \mathsf{C}_2(-\delta)\mathsf{C}_3(\lambda) = \begin{pmatrix} \cos\delta\cos\lambda & \cos\delta\sin\lambda & \sin\delta \\ -\sin\lambda & \cos\lambda & 0 \\ -\sin\delta\cos\lambda & -\sin\delta\sin\lambda & \cos\delta \end{pmatrix}. \tag{2.50}$$

With planetary angular velocity, $\boldsymbol{\Omega} = \Omega\mathbf{K}$, and the angular velocity of the local horizon frame relative to the planet-fixed frame given by:

$$\boldsymbol{\Omega}_{LH} = \dot{\lambda}\mathbf{K} - \dot{\delta}\mathbf{i}_{\boldsymbol{\lambda}} = \dot{\lambda}\sin\delta\mathbf{i}_\mathbf{r} - \dot{\delta}\mathbf{i}_{\boldsymbol{\lambda}} + \dot{\lambda}\cos\delta\mathbf{i}_{\boldsymbol{\delta}}, \tag{2.51}$$

we can express the relative and inertial velocities, $\mathbf{v}(t), \mathbf{v}_\mathbf{I}(t)$, respectively, as follows:

$$\mathbf{v} = \dot{h}\mathbf{i}_\mathbf{r} + (R_0 + h)(\dot{\lambda}\cos\delta\mathbf{i}_{\boldsymbol{\lambda}} + \dot{\delta}\mathbf{i}_{\boldsymbol{\delta}}), \tag{2.52}$$

$$\mathbf{v}_\mathbf{I} = \mathbf{v} + \Omega(R_0 + h)\cos\delta\mathbf{i}_{\boldsymbol{\lambda}}. \tag{2.53}$$

Here, $r(t) = R_0 + h(t)$ is the radial coordinate of the center of mass, with R_0 being the (constant) radius of the spherical planet and $h(t)$, the instantaneous altitude.

Comparing (2.52) with (2.48) we have the following translational kinematics equations:

$$\dot{h} = v\sin\phi, \tag{2.54}$$

$$\dot{\delta} = \frac{v \cos\phi \cos\psi}{R_0 + h}, \tag{2.55}$$

$$\dot{\lambda} = \frac{v \cos\phi \sin\psi}{(R_0 + h)\cos\delta}. \tag{2.56}$$

2.4.2 Attitude Flight Dynamics

The flight vehicle is often considered to be a collection of rigid bodies rotating with respect to one another. The outer frame of the vehicle thus contains a number of rotors spinning with respect to the vehicle about various axes. Using the body-fixed frame with origin at vehicle's center of mass, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, we can resolve the net angular momentum of the vehicle as follows:

$$\mathbf{H} = \mathsf{J}\boldsymbol{\omega} + \mathbf{H_R} = \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ & J_{yy} & J_{yz} \\ Symm. & & J_{zz} \end{pmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} + \begin{Bmatrix} H_{Rx} \\ H_{Ry} \\ H_{Rz} \end{Bmatrix}, \tag{2.57}$$

where $\mathbf{H_R} = H_{Rx}\mathbf{i} + H_{Ry}\mathbf{j} + H_{Rz}\mathbf{k}$ is the net (constant) angular momentum of all rotors relative to the vehicle frame. Writing the external torque vector in the body-fixed frame as:

$$\boldsymbol{\tau} = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}, \tag{2.58}$$

and substituting (2.57) and (2.58) into (2.16), we have the following rotational kinetics equation:

$$\begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = \mathsf{J} \begin{Bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{Bmatrix} + \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} \times \left(\mathsf{J} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} + \begin{Bmatrix} H_{Rx} \\ H_{Ry} \\ H_{Rz} \end{Bmatrix} \right). \tag{2.59}$$

Typically, the rotor angular momentum is negligible for aircraft and rockets, but can be significant for a spacecraft. Note that rotational kinetic equations have constant (equilibrium) solutions for the body rates P, Q, R when the external torque vanishes, $\mathbf{H_R}$ is constant, and any two body rates are zeros. Such equilibrium solutions are commonly employed in the design of flight control systems. Other equilibrium solutions are torque-free rotation of axisymmetric vehicles about their axis of symmetry, which form the basis of spin stabilization of rockets and spacecraft. For example, axisymmetric rockets can roll at a constant rate, P^{e}, but with zero pitch and yaw rates. On the other hand, airplanes and asymmetric satellites in horizontal flight can have a constant pitch rate, Q^{e}, but zero roll and yaw rates.

The rotational kinematics equation can be derived from (2.23) with the help of a suitable attitude representation, as discussed earlier. Usually, rotational stability

analysis requires a small angular displacement from a nominal attitude. In such a case, the attitude perturbation is represented by $(\Psi)_3, (\Theta)_2, (\sigma)_1$ Euler sequence (Fig. 2.2), where the angles Ψ, Θ, σ are small. When the small attitude perturbation is measured with reference to an axis normal to the nominal attitude, the $(\Psi)_3, (\Theta)_1, (\sigma)_3$ representation (Fig. 2.2) is used. In both cases, linearized models of attitude kinematics are possible. However, for an arbitrarily large perturbation, a nonsingular representation (such as quaternion) is necessary, resulting in an essentially nonlinear model.

2.5 Flight Dynamics System

We can now collect the equations of motion of a generic, rigid flight vehicle in a reference frame rotating with a constant angular velocity, $\mathbf{\Omega}$, in terms of position, $\mathbf{r}(t)$, and velocity, $\mathbf{v}(t)$, of the center of mass, o, as well as angular velocity, $\boldsymbol{\omega}(t)$, and rotation matrix defining angular orientation, $\mathsf{C}(t)$. The external force and moment vectors are expressed in terms of the contributing effects of control ($\mathbf{F_C}(t)$, $\mathbf{M_C}(t)$), environment ($\mathbf{F_E}(t)$, $\boldsymbol{\tau}_\mathbf{E}(t)$), and disturbance ($\mathbf{F_D}(t)$, $\boldsymbol{\tau}_\mathbf{D}(t)$) as follows:

$$\mathbf{v} = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \dot{\mathbf{r}}_\mathbf{o} + \mathbf{\Omega} \times \mathbf{r}. \tag{2.60}$$

$$\mathbf{F_C} + \mathbf{F_D} + \mathbf{F_E} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = m\left(\dot{\mathbf{v}}_\mathbf{o} + \mathbf{\Omega} \times \mathbf{v}\right). \tag{2.61}$$

$$\boldsymbol{\tau}_\mathbf{C} + \boldsymbol{\tau}_\mathbf{E} + \boldsymbol{\tau}_\mathbf{D} = \mathsf{J}\dot{\boldsymbol{\omega}} + \mathsf{S}(\boldsymbol{\omega})\mathsf{J}\boldsymbol{\omega}. \tag{2.62}$$

$$\frac{\mathrm{d}\mathsf{C}}{\mathrm{d}t} = -\mathsf{S}(\boldsymbol{\omega})\mathsf{C}(t). \tag{2.63}$$

Here, overdot represents time derivative. The mass, m, and the inertia tensor, J, of the vehicle could be changing with time due to propellant consumption, and the respective thrust and torque terms due to varying mass and mass distribution are assumed to be included in the environmental effects on the left-hand side of (2.61) and (2.62). Although the general attitude kinematics equation, (2.63), has been expressed in terms of the rotation matrix, it is more convenient in practice to use a generic parameter vector, $\boldsymbol{\xi}(t)$, that could be a set of Euler angles, quaternion, or any other attitude kinematical representation. Assuming that the environmental, control, and distrubance inputs are known, the equations of motion, (2.60)–(2.63), can be completely described by the variables of motion, $\mathbf{r}$, $\mathbf{v}$, $\boldsymbol{\omega}$, $\boldsymbol{\xi}$. Thus, we can represent the state of the flight dynamic system by the state vector, $\boldsymbol{\zeta}(t)$, given by:

$$\boldsymbol{\zeta}(t) = \begin{Bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \\ \boldsymbol{\omega}(t) \\ \boldsymbol{\xi}(t) \end{Bmatrix}. \tag{2.64}$$

The state vector is not unique, but depends upon the choice of frames of reference in the equations of motion. Any vector can be resolved in a variety of coordinate frames, resulting in different scalar differential equations of motion for the same flight vehicle.

The environmental inputs arise because of sustained flight in a given environment (either atmosphere or space). The environmental effects are described by additional equations – called environmental equations – involving functions of the state vector, $\boldsymbol{\zeta}(t)$, and time, written in a vector form as follows:

$$\begin{Bmatrix} \mathbf{F_E}(t) \\ \boldsymbol{\tau}_\mathbf{E}(t) \end{Bmatrix} = \mathbf{f_e}(\boldsymbol{\zeta}(t), t), \tag{2.65}$$

where $\mathbf{f_e}(.)$ is the environmental vector functional. In atmospheric flight, the environmental force, $\mathbf{F_E}$, is a sum of gravity, $\mathbf{F_g}$, thrust, $\mathbf{F_t}$, and aerodynamic force, $\mathbf{F_a}$, while the aerodynamic moment $\boldsymbol{\tau}_\mathbf{a}$ is the only environmental torque. In space flight, the typical environmental effects are gravity, solar-radiation pressure (force and torque, $\mathbf{F_s}$, $\boldsymbol{\tau}_\mathbf{s}$), third-body and nonspherical gravity force, $\mathbf{F_{\delta g}}$, gravity-gradient torque, $\boldsymbol{\tau}_\mathbf{g}$, and aerodynamic force and moment in rarefied flow ($\mathbf{F_a}$, $\boldsymbol{\tau}_\mathbf{a}$).

Often, it is not feasible to model all environmental effects as deterministic processes. Examples include random atmospheric gusts and sloshing of propellants. In such cases, some environmental effects are treated as disturbance forces and moments ($\mathbf{F_D}$, $\boldsymbol{\tau}_\mathbf{D}$), and are generally treated as stochastic (statistical) processes. The disturbance inputs include errors (or uncertainty) in modeling the flight dynamic system due to idealizing assumptions (such as rigid body approximation). Therefore, all disturbances in the flight dynamics model are clubbed into the *process noise* vector, $\boldsymbol{\nu}(t)$, that is modeled as a stochastic input:

$$\boldsymbol{\nu}(t) \doteq \begin{Bmatrix} \mathbf{F_D}(t) \\ \boldsymbol{\tau}_\mathbf{D}(t) \end{Bmatrix}. \tag{2.66}$$

Finally, control inputs are required either to maintain a nominal trajectory in presence of disturbances or to achieve a change of state in a given time. The control input vector,

$$\mathbf{u}(t) \doteq \begin{Bmatrix} \mathbf{F_C}(t) \\ \boldsymbol{\tau}_\mathbf{C}(t) \end{Bmatrix}, \tag{2.67}$$

obeys *control laws* involving functions of either the actual state vector, or its estimation from the measurements of an output vector. We will address the derivation of control laws in later chapters. Examples of control inputs are thrust-vectoring and aerodynamic control forces and moments for aircraft and rockets, and orbital and attitude thruster force and moment, gyroscopic moment, and magnetic torques for spacecraft.

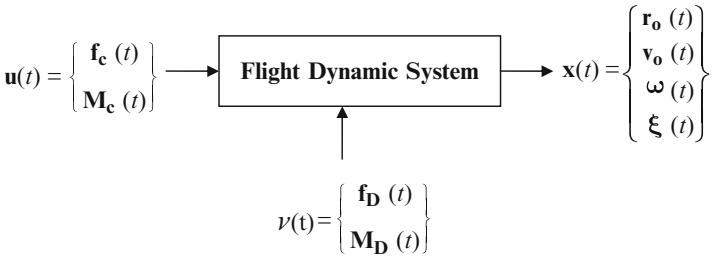


Fig. 2.5 A typical flight dynamic system

The generic flight dynamic system is depicted as a block-diagram in Fig. 2.5. Note that the environmental effects are inherent to the system, and hence are not shown as explicit inputs. Furthermore, the control laws, actuators and sensors, are excluded from Fig. 2.5. We can now express the equations governing flight dynamics as the following *state equation* to be satisfied by the state vector:

$$\frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t} = \mathbf{f}(\boldsymbol{\zeta}, \mathbf{u}, \boldsymbol{\nu}, t), \tag{2.68}$$

where $\mathbf{f}(.)$ is a vector functional possessing continuous derivatives of the state, control, and process noise vectors. The general flight dynamic system state equation is nonlinear in nature, requiring numerical methods for solution. Automatic control law derivation is generally based upon a deterministic plant, for which the noise vector vanishes, $\boldsymbol{\nu}(t) = \mathbf{0}$.

The output vector, $\mathbf{y}(t)$, of the flight dynamics system can be described by the following *output equation*:

$$\mathbf{y} = \mathbf{h}(\boldsymbol{\zeta}, \mathbf{u}, \mathbf{w}, t), \tag{2.69}$$

where $\mathbf{h}(.)$ is a vector functional and $\mathbf{w}(t)$ is a stochastic input called the *measurement noise* arising because of the imperfections in the sensors.

In advanced flight control applications called *terminal control*, (2.68) is solved iteratively in order to determine the control input, $\mathbf{u}(t)$, that drives the system's state from an initial state, $\boldsymbol{\zeta}(0)$, to a final state, $\boldsymbol{\zeta}(t_\mathrm{f})$, in a specified time, t_f. Such a mathematical problem is termed *two-point boundary value problem* (2PBVP) and requires a sophisticated solution procedure. Examples of nonlinear 2PBVP include long-range guidance of missiles and interplanetary spacecraft and are beyond our present scope. However, we shall consider some linear 2PBVP problems later.

As discussed in Chap. 1, another class of automatic flight control problems – called *tracking control* – involves tracking a known solution of the nonlinear state equation. In such cases, (2.68) is linearized about a *nominal trajectory*, $\boldsymbol{\zeta}_\mathbf{n}(t)$, that is the solution of the *unforced* state equation of the plant.

$$\frac{\mathrm{d}\boldsymbol{\zeta}_\mathbf{n}}{\mathrm{d}t} = \mathbf{f}(\boldsymbol{\zeta}_\mathbf{n}, \mathbf{0}, \mathbf{0}, t). \tag{2.70}$$

Clearly, the nominal trajectory does not require any control inputs as it is free of disturbances. When a small disturbance vector, $\boldsymbol{\nu}(t)$, is present the system requires a nonzero control input, $\mathbf{u}(t)$, to maintain the disturbed system close to the nominal trajectory. Therefore, for all intents and purposes of control law derivation, the deterministic flight dynamics plant is rendered linear by a Taylor's series expansion of (2.68) about the nominal trajectory, and the linearized state equation for deviation from the nominal trajectory, $\mathbf{x}(t) = \boldsymbol{\zeta}(t) - \boldsymbol{\zeta}_{\mathbf{n}}(t)$, is the following:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)\mathbf{u}(t), \tag{2.71}$$

where $\mathsf{A}(t), \mathsf{B}(t)$ are coefficient matrices. We will now consider some important flight dynamic systems.

2.6 Space Flight Dynamics

The generic flight dynamics system, (2.60)–(2.63), yields the following equations for space flight in the ideal vacuum environment, ($\mathbf{F_D} = \boldsymbol{\tau}_{\mathbf{E}} = \boldsymbol{\tau}_{\mathbf{D}} = \mathbf{0}$), with $\mathbf{F_E} = m\mathbf{g}$, an inertial frame of reference for translational flight and a body-fixed frame for attitude dynamics:

$$\mathbf{v} = \tfrac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \dot{\mathbf{r}}. \tag{2.72}$$

$$\mathbf{F_C} + m\mathbf{g} = m\tfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = m\ddot{\mathbf{r}}. \tag{2.73}$$

$$\boldsymbol{\tau}_{\mathbf{C}} = \mathsf{J}\dot{\boldsymbol{\omega}} + \mathsf{S}(\boldsymbol{\omega})\mathsf{J}\boldsymbol{\omega}. \tag{2.74}$$

$$\dot{\mathsf{C}} = -\mathsf{S}(\boldsymbol{\omega})\mathsf{C}. \tag{2.75}$$

The space flight translational dynamics equations have unforced ($\mathbf{F_C} = \mathbf{0}$) solutions called *two-body orbits*, and the corresponding unforced rotational states ($\boldsymbol{\tau}_{\mathbf{C}} = \mathbf{0}$) are referred to as the *torque-free rotation*. Here, we shall consider the unforced trajectories that can be regarded as nominal for designing a flight control system in the presence of disturbances.

2.6.1 Orbital Mechanics

Uncontrolled flight of a spacecraft is primarily influenced by the gravity of a massive celestial object such as the Earth or the Sun, approximated to be spherical in shape and mass distribution. The gravitational perturbation due to actual, nonspherical shapes, the gravity of other distant bodies, as well as atmospheric and solar radiation effects are generally quite small in comparison with the spherical gravity of the primary body, and thus are treated as small disturbances. These disturbances are

usually compensated for by a tracking system designed around unforced, nominal trajectories (*orbits*), which are determined by the solution of the problem of two spherical bodies in mutual gravitational attraction, called the *two-body problem*. The translational dynamics equation, (2.73), can be expressed as follows for a spacecraft of mass m under the gravitational attraction of a large spherical body of mass M (called *planet*) by Newton's law of gravitation:

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{0}, \tag{2.76}$$

where $\mathbf{r}(t)$ is the position of the spacecraft's center of mass relative to the planetary center and $\mu = GM$, the gravitational constant. As the planet is largely unaffected by the small spacecraft mass, the origin of the inertial frame can be taken to coincide with the planetary center that is approximated to be at rest. In order to determine the shape of an orbit, we take the vector product of both the sides of (2.76) with $\mathbf{r}$, resulting in

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{r} \times \dot{\mathbf{r}}) = \mathbf{0}, \tag{2.77}$$

which implies that the *specific orbital angular momentum*,

$$\mathbf{h} = \mathbf{r} \times \dot{\mathbf{r}} = \mathbf{r} \times \mathbf{v}, \tag{2.78}$$

is conserved. Because $\mathbf{h}$ is a constant vector, it follows that:

(a) The direction of $\mathbf{h}$ is a constant, which implies that the vectors $\mathbf{r}$ and $\mathbf{v}$ are always in the same plane – called *orbital plane* – and $\mathbf{h}$ is normal to that plane.
(b) The magnitude of $\mathbf{h}$ is constant. Writing $\mathbf{h}$ in polar coordinates, (r, θ),

$$h = |\,\mathbf{h}\,| = |\,\mathbf{r} \times \mathbf{v}\,| = r^2\dot{\theta}, \tag{2.79}$$

this implies that the radius vector, $\mathbf{r}$, sweeps out area at a constant rate, $\frac{1}{2}r^2\dot{\theta}$ (Kepler's second law of planetary motion).

The orbits are classified according to the magnitude and direction of the constant angular momentum, $\mathbf{h}$.

By taking the vector product of both sides of (2.76) with $\mathbf{h}$, it can be shown that the following *eccentricity vector*, $\mathbf{e}$, is a constant:

$$\mathbf{e} = \frac{1}{\mu}(\mathbf{v} \times \mathbf{h}) - \frac{\mathbf{r}}{r}. \tag{2.80}$$

As the eccentricity vector is normal to $\mathbf{h}$, it lies in the orbital plane and can be used as a reference for the direction of the position vector, $\mathbf{r}(t)$. The angle, $\theta(t)$, made by $\mathbf{r}(t)$ with $\mathbf{e}$ (measured along the flight direction) is called the *true anomaly*.

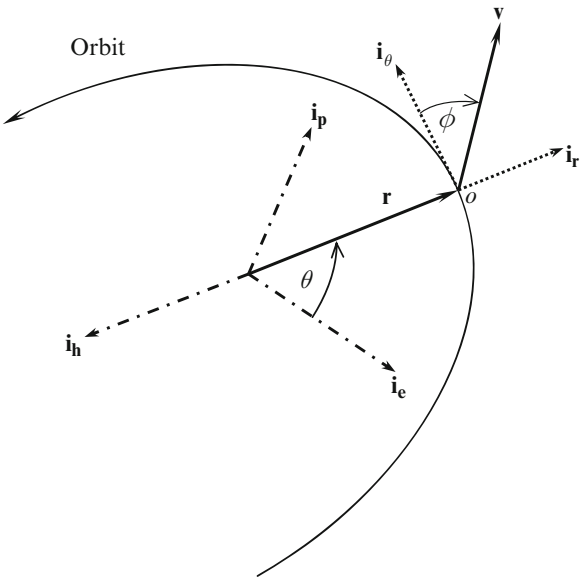


Fig. 2.6 Spacecraft's orbital position and velocity resolved in the perifocal frame, $\mathbf{i_e}, \mathbf{i_p}, \mathbf{i_h}$

Figure 2.6 depicts an orbital coordinate system consisting of unit vectors, $\mathbf{i_e}, \mathbf{i_h}$ along $\mathbf{e}, \mathbf{h}$, respectively, and a third unit vector, $\mathbf{i_p}$, normal to both $\mathbf{e}$ and $\mathbf{h}$ such that

$$\mathbf{i_h} = \mathbf{i_e} \times \mathbf{i_p}. \tag{2.81}$$

Such a coordinate system, called the *perifocal frame*, is useful in resolving the position vector in the orbital plane in terms of the true anomaly, $\theta(t)$:

$$\mathbf{r} = r\left(\mathbf{i_e}\cos\theta + \mathbf{i_p}\sin\theta\right). \tag{2.82}$$

As the velocity vector always lies in the orbital plane, $(\mathbf{i_e}, \mathbf{i_p})$, one can express the velocity in terms of the flight path angle, ϕ, as follows (Fig. 2.6):

$$\mathbf{v} = v\left[\sin(\phi-\theta)\mathbf{i_e} + \cos(\phi-\theta)\mathbf{i_p}\right]. \tag{2.83}$$

By substituting (2.82) and (2.83) into (2.78), we have

$$h = rv\cos\phi. \tag{2.84}$$

By taking the scalar product of $\mathbf{e}$ with $\mathbf{r}(t)$ we have

$$r + \mathbf{e}\cdot\mathbf{r} = \frac{1}{\mu}\mathbf{r}\cdot(\mathbf{v}\times\mathbf{h}) = \frac{h^2}{\mu}, \tag{2.85}$$

or,

$$r = \frac{h^2/\mu}{1 + e\cos\theta}. \tag{2.86}$$

Equation (2.86) – called the *orbit equation* – defines the shape of the orbit in polar coordinates, (r, θ), and indicates that the orbit is symmetrical about $\mathbf{i_e}$. Furthermore, the minimum separation of the two bodies (called *periapsis*) occurs for $\theta = 0$, implying that $\mathbf{i_e}$ points toward the periapsis. From the orbit equation it is clear that the general orbit is a *conic section*, i.e., the shape one gets by cutting a right-circular cone in a given way. For $e < 1$, the orbit is an *ellipse*. The circle is a special ellipse with $e = 0$ and $r = h^2/\mu$. For $e = 1$, the orbit is a *parabola*. The *rectilinear* trajectory is a special parabola with $h = 0$. For $e > 1$, the orbit is a *hyperbola*. In all cases, the *focus* of the orbit is at the center of the celestial body, and the *semimajor axis*, a, is given by:

$$a = \frac{h^2/\mu}{1 - e^2}. \tag{2.87}$$

Another useful orbital constant is the parameter, p, defined as the radius when $\theta = \pi/2$:

$$p = r|_{\theta=\pi/2} = h^2/\mu = a(1 - e^2), \tag{2.88}$$

which enables the orbit equation to be written as follows:

$$r = \frac{p}{1 + e\cos\theta}. \tag{2.89}$$

By substituting the orbit equation into (2.84), we have

$$\cos\phi = \frac{\mu(1 + e\cos\theta)}{hv}. \tag{2.90}$$

By taking the time derivative of the orbit equation and equating it to the radial velocity component, $\dot{r} = v\sin\phi$ (Fig. 2.6), we write

$$\dot{r} = v\sin\phi = \frac{ep\sin\theta}{1 + e\cos\theta}\dot{\theta}, \tag{2.91}$$

which by substitution of (2.79), becomes

$$\sin\phi = \frac{\mu e\sin\theta}{hv}. \tag{2.92}$$

Equations (2.90) and (2.92) allow us to determine the flight path angle, ϕ, uniquely from the true anomaly, θ. On eliminating the flight path angle from the velocity expression, (2.83), results in the following simplification:

$$\mathbf{v} = \frac{\mu}{h}\left[-\sin\theta\mathbf{i_e} + (e + \cos\theta)\mathbf{i_p}\right]. \tag{2.93}$$

2.6.1.1 Kepler's Equation

The vectors **h** and **e** completely determine the shape and orientation of a two-body trajectory, but do not provide any information about the location of the spacecraft at a given time. This missing data is usually expressed as an equation that relates the variation of true anomaly with time, $\theta(t)$, beginning with the *time of periapsis*, τ, for which $\theta = 0$. On substituting the orbit equation, (2.89), into (2.79), we have

$$\dot{\theta} = \sqrt{\frac{\mu}{p^3}}(1 + e\cos\theta)^2. \tag{2.94}$$

The time integral of (2.94) provides τ, thereby determining the function $\theta(t)$, and completing the solution to the two-body problem. However, because of the nonlinear nature of (2.94), such an integration is usually carried out by a numerical procedure. For illustration, we will consider only the elliptical orbit and refer the reader to Tewari [21] for other details.

For an elliptical orbit ($0 < e < 1$), (2.94) becomes

$$\frac{(1-e^2)^{\frac{3}{2}}\mathrm{d}\theta}{(1+e\cos\theta)^2} = n\mathrm{d}t, \tag{2.95}$$

where

$$n \doteq \sqrt{\frac{\mu}{a^3}} \tag{2.96}$$

is referred to as the orbital *mean motion*. In order to simplify (2.95), we introduce an *eccentric anomaly, E* defined by:

$$\cos E = \frac{e + \cos\theta}{1 + e\cos\theta} \tag{2.97}$$

which substituted into (2.95) yields the following *Kepler's equation*:

$$E - e\sin E = M, \tag{2.98}$$

where $M = n(t - \tau)$ is called the *mean anomaly*. The relationship between the true and the eccentric anomalies is given by the following:

$$\tan\frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}. \tag{2.99}$$

Note that $E/2$ and $\theta/2$ are always in the same quadrant. A numerical solution to Kepler's equation can be easily obtained by Newton–Raphson's method [21].

Given the position and velocity at some time, t_0, one would often like to determine the position and velocity at some other time, t, which translates into determining eccentric anomaly, E, from a given mean anomaly, M. After solving Kepler's equation for E, we can express the position and velocity vectors of an elliptical orbit directly as follows:

$$\begin{aligned}\mathbf{r} &= a(\cos E - e)\mathbf{i_e} + \sqrt{ap}\sin E\mathbf{i_p}\\ \mathbf{v} &= -\frac{\sqrt{\mu a}}{r}\sin E\mathbf{i_e} + \frac{\sqrt{\mu p}}{r}\cos E\mathbf{i_p}.\end{aligned} \tag{2.100}$$

2.6.1.2 Celestial Frame of Reference

The spacecraft's position and velocity are generally resolved in an inertial, *celestial frame*, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, which is fixed with respect to distant stars and has its origin at the center of the celestial body. An example of celestial frames is the *equatorial frame* of Earth, with the axis $\mathbf{I}$ pointing in the direction of *vernal equinox* – the unique point of the Sun's apparent annual crossing of the equatorial plane from the south to the north. The other two axes of the frame are chosen such that $\mathbf{K}$ is the polar axis, and $\mathbf{I}\times\mathbf{J} = \mathbf{K}$, as shown in Fig. 2.7. The orientation of the right-handed perifocal, orbital frame, $\mathbf{i_e}, \mathbf{i_p}, \mathbf{i_h}$, relative to the celestial frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, is classically represented by the Euler angles sequence, $(\Omega)_3, (i)_1, (\omega)_3$, as shown in Fig. 2.7. However, such a representation has singularities at $i = 0, \pi$, as we know from earlier discussion.

The intersection of the spacecraft's orbital plane with the equatorial plane yields the *line of nodes*. The *ascending node* is the name given to the point on the line of nodes where the orbit crosses the equatorial plane from the south to the north, as shown in Fig. 2.7. The unit nodal vector, $\mathbf{n}$, pointing toward the ascending node makes an angle Ω with the axis $\mathbf{I}$, measured in the equatorial plane in an anticlockwise direction (Fig. 2.7). The angle Ω is termed *right ascension of the ascending node*. The *orbital inclination*, i, is the angle made by the orbital plane with the equatorial plane, and is also the positive rotation about $\mathbf{n}$ required to produce $\mathbf{i_h}$ from the axis $\mathbf{K}$. The third Euler angle ω represents a positive rotation of $\mathbf{n}$ about $\mathbf{i_h}$ to produce $\mathbf{i_e}$ in the orbital plane, and is called *argument of periapsis*.

The coordinate transformation between the perifocal and the celestial frames is given by the following rotation matrix:

$$\mathsf{C} = \mathsf{C}_3(\omega)\mathsf{C}_1(i)\mathsf{C}_3(\Omega), \tag{2.101}$$

where

$$\begin{Bmatrix}\mathbf{I}\\ \mathbf{J}\\ \mathbf{K}\end{Bmatrix} = \mathsf{C}^{\mathrm{T}}\begin{Bmatrix}\mathbf{i_e}\\ \mathbf{i_p}\\ \mathbf{i_h}\end{Bmatrix}. \tag{2.102}$$

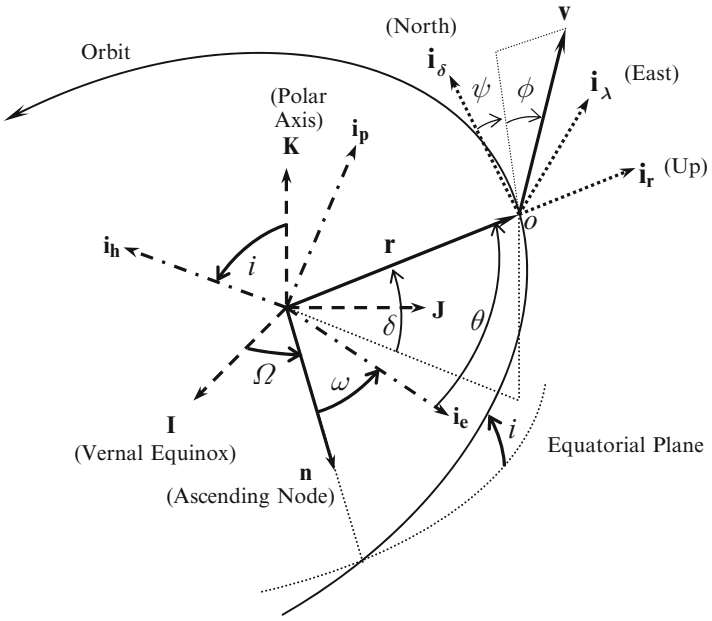


Fig. 2.7 Spacecraft's orbital geometry: (**a**) orientation of the perifocal frame, $\mathbf{i_e}, \mathbf{i_p}, \mathbf{i_h}$, relative to celestial plane, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, given by Euler angles, $(\Omega)_3, (i)_1, (\omega)_3$, (**b**) the orbital velocity vector resolved in the local horizon frame, $(\mathbf{i_r}, \mathbf{i_\lambda}, \mathbf{i_\delta})$

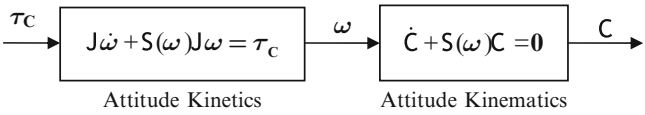


Fig. 2.8 Spacecraft's attitude plant subsystems in a cascade relationship

2.6.2 *Spacecraft Attitude Dynamics*

In contrast with atmospheric flight mechanics, there is no coupling between the attitudinal (rotational) and the translational motions of a spacecraft. This fact offers a great simplification, in which a cascade relationship is possible between rotational kinetics (2.74) and rotational kinematics (2.75). Such a relationship is depicted in Fig. 2.8, where we see that the control torque, $\boldsymbol{\tau}_\mathbf{C}(t)$, is the applied input to the rotational kinetics block whose output, $\boldsymbol{\omega}(t)$, serves as the input to the rotational kinematics subsystem. Hence, one can separately design control laws for the two subsystems. The control torque is applied either externally by rocket thrusters or internally by rotors (reaction wheels and gyros). We shall return to spacecraft attitude control in Chap. 6.

2.7 Atmospheric Flight Dynamics

Atmospheric flight is dominated by aerodynamic forces and moments. The linear velocity of the rigid flight vehicle's center of mass, $\mathbf{v}$, and the angular velocity about the center of mass, $\boldsymbol{\omega}$, both relative to a reference frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, are resolved in a body-fixed frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, (Fig. 2.9) with origin at the center of mass as follows:

$$\mathbf{v} = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}, \tag{2.103}$$

and

$$\boldsymbol{\omega} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}. \tag{2.104}$$

Borrowing from aeronautical terminology, U is called the *forward speed*, V the *sideslip speed*, and W the *plunge* (or *downward*) *speed*. Furthermore, P is called the *roll rate*, Q the *pitch rate*, and R the *yaw rate*. The aerodynamic moment, $\boldsymbol{\tau}_{\mathbf{a}}$, is resolved in the body-fixed frame, $(oxyz)$ as follows:

$$\boldsymbol{\tau}_{\mathbf{a}} = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}, \tag{2.105}$$

where L is called the *rolling moment*, M the *pitching moment*, and N is the *yawing moment*. The engine thrust, $\mathbf{T}(t)$, resolved in the body-fixed frame using *thrust angles*, $\mu_1(t)$, $\mu_2(t)$, (Fig. 2.10) is the following:

$$\mathbf{T} = T(\mathbf{i}\cos\mu_1\cos\mu_2 + \mathbf{j}\cos\mu_1\sin\mu_2 + \mathbf{k}\sin\mu_1). \tag{2.106}$$

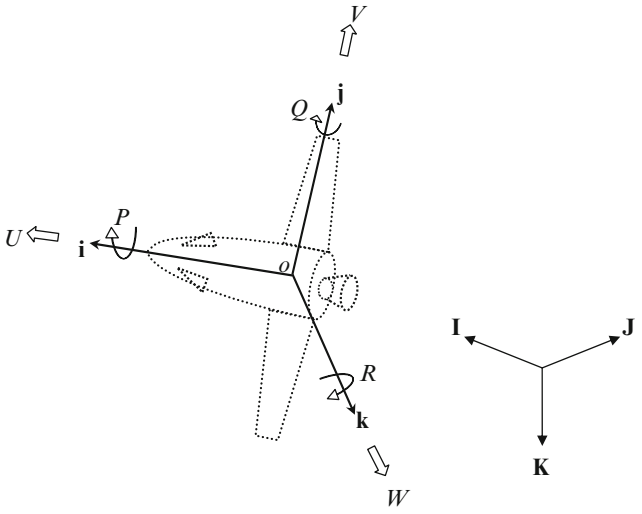


Fig. 2.9 A body-fixed frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, relative to a reference frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$

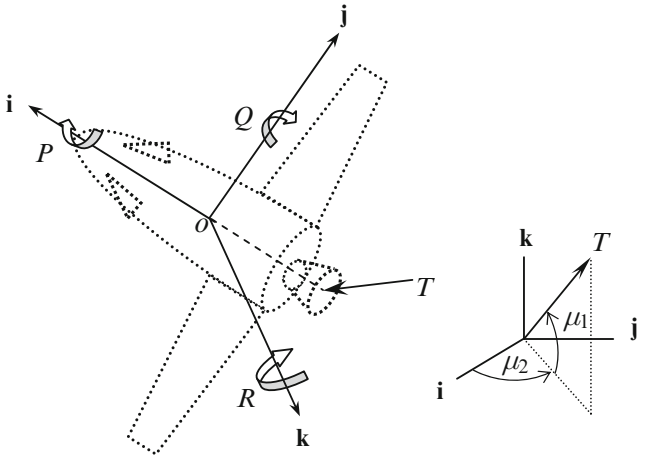


Fig. 2.10 The body-fixed frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, and thrust angles, ϵ and γ

Propulsive efficiency requires that the thrust angles should be small, which results in the approximation

$$\mathbf{T} \simeq T(\mathbf{i} + \mathbf{j}\mu_2 + \mathbf{k}\mu_1). \tag{2.107}$$

Furthermore, the control torque applied by thrust deflection is the following:

For aircraft stability and control applications (Chap. 4), the thrust is usually clubbed with the aerodynamic force vector, $\mathbf{F_a}$, as follows:

$$\mathbf{F_a} + \mathbf{T} = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}, \tag{2.108}$$

where X is called the *forward force*, Y the *sideforce*, and Z is the *downward force*.

The general attitude kinematics of an atmospheric flight vehicle can be represented by the quaternion, (q_1, q_2, q_3, q_4). Of the translational kinematic variables, only the altitude, h, is important as it governs the atmospheric properties and thus aerodynamic forces and moments. Therefore, we generally have a seven degree-of-freedom, eleventh-order, basic flight dynamics model shown in Fig. 2.11. However, when applied to the level flight dynamics of aircraft (Chap. 4), the variation in altitude for stability and control applications is negligible, and the quaternion is typically replaced by the Euler angles, resulting in a six degree-of-freedom, ninth-order system.

2.7.1 Wind Axes

It is often difficult to express the aerodynamic forces directly in the body-fixed frame, for which another moving reference frame with an axis, $\mathbf{i_v}$, along the

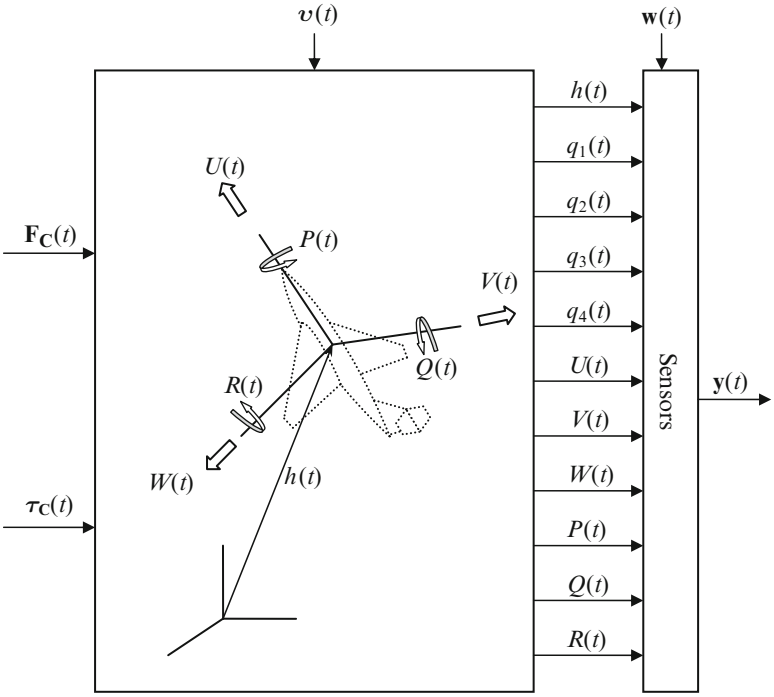


Fig. 2.11 Atmospheric flight dynamics system

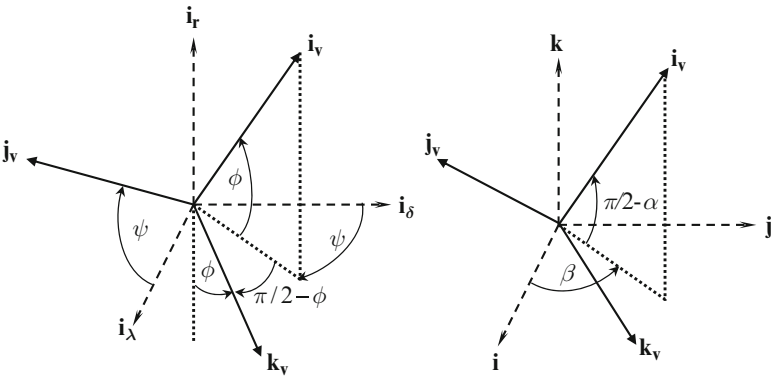


Fig. 2.12 The wind axes, $(\mathbf{i_v}, \mathbf{j_v}, \mathbf{k_v})$, relative to the body-fixed frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, and the local horizon frame, $(\mathbf{i_r}, \mathbf{i_\lambda}, \mathbf{i_\delta})$

instantaneous relative velocity vector, $\mathbf{v}(t)$, is chosen for convenience. Such a frame, $ox_v y_v z_v$, called *wind axes*, is shown in Fig. 2.12 and is very useful in expressing the aerodynamic force, $\mathbf{F_a}$, and aerodynamic moment, $\boldsymbol{\tau}_\mathbf{a}$, both of which, by definition,

depend upon the relative motion between the vehicle and the atmosphere.[3] The *lift*, $\mathcal{L}$, the *driftforce*, f_y, and the *drag*, $\mathcal{D}$ are the components of the aerodynamic force vector along the wind axes:

$$\mathbf{F_a} = -\mathcal{D}\mathbf{i_v} + f_y\mathbf{j_v} - \mathcal{L}\mathbf{k_v}. \tag{2.109}$$

Note that the lift and driftforce act normal to the flight direction, $\mathbf{i_v}$, thereby causing a change of flight path, while drag is always opposed to it and acts as an energy sink.

The coordinate transformation between the wind axes and the local horizon frame is given by the Euler angle sequence $(-\psi)_1, (\phi - \frac{\pi}{2})_2$:

$$\left\{\begin{array}{c}\mathbf{i_v}\\ \mathbf{j_v}\\ \mathbf{k_v}\end{array}\right\} = \mathsf{C}_W \left\{\begin{array}{c}\mathbf{i_r}\\ \mathbf{i_\lambda}\\ \mathbf{i_{ffi}}\end{array}\right\}, \tag{2.110}$$

where

$$\mathsf{C}_W = \mathsf{C}_2\left(\phi - \frac{\pi}{2}\right)\mathsf{C}_1(-\psi) = \begin{pmatrix}\sin\phi & \cos\phi\sin\psi & \cos\phi\cos\psi\\ 0 & \cos\psi & -\sin\psi\\ -\cos\phi & \sin\phi\sin\psi & \sin\phi\cos\psi\end{pmatrix} \tag{2.111}$$

The linear acceleration of the vehicle's center of mass can be expressed in the wind axes as follows:

$$\dot{\mathbf{v}} = a_{xv}\mathbf{i_v} + a_{yv}\mathbf{j_v} + a_{zv}\mathbf{k_v}, \tag{2.112}$$

where

$$\begin{aligned} a_{xv} &= \dot{v} + \Omega^2 r\cos\delta(\cos\phi\cos\psi\sin\delta - \sin\phi\cos\delta)\\ a_{yv} &= v\cos\phi\dot{\psi} - \frac{v^2}{r}\cos^2\phi\sin\psi\tan\delta - \Omega^2 r\sin\psi\sin\delta\cos\delta\\ &\quad +2\Omega v(\sin\phi\cos\psi\cos\delta - \cos\phi\sin\delta)\\ a_{zv} &= -v\dot{\phi} + \frac{v^2}{r}\cos\phi + 2\Omega v\sin\psi\cos\delta\\ &\quad +\Omega^2 r\cos\delta(\sin\phi\cos\psi\sin\delta + \cos\phi\cos\delta). \end{aligned} \tag{2.113}$$

In stability and control applications, the planetary rotation is typically neglected, leading to a simplification in the acceleration terms.

[3] Even when the aerodynamics is unimportant (as in space flight), the wind axes provide a convenient reference frame for the vehicle's attitude.

For aerodynamic purposes, the orientation of the body-fixed frame, $(oxyz)$, relative to the wind axes is given by the Euler angles, α (*angle-of-attack*) and β (*sideslip angle*) in the sequence $(\beta)_3, (-\alpha)_2$ (Fig. 2.12). The coordinate transformation between the wind axes and the body-fixed frame is thus given by:

$$\begin{Bmatrix} \mathbf{i_v} \\ \mathbf{j_v} \\ \mathbf{k_v} \end{Bmatrix} = \mathsf{C}_B \begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix}, \tag{2.114}$$

where

$$\mathsf{C}_B = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta & \sin\alpha \\ -\sin\beta & \cos\beta & 0 \\ -\sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{pmatrix} \tag{2.115}$$

Referring to Fig. 2.12 and the first row of (2.114) we write

$$\mathbf{v} = v(\cos\alpha\cos\beta\mathbf{i} + \cos\alpha\sin\beta\mathbf{j} + \sin\alpha\mathbf{k}), \tag{2.116}$$

where the angle-of-attack and the sideslip angle are related to the relative velocity components as follows:

$$\alpha = \sin^{-1}\frac{W}{\sqrt{U^2+V^2+W^2}}, \tag{2.117}$$

$$\beta = \tan^{-1}\frac{V}{U}. \tag{2.118}$$

The dependence of the aerodynamic forces on the angle-of-attack and the sideslip angle is crucial to stability and control.

2.7.2 *Aerodynamic Forces and Moments*

The fundamental sources of aerodynamic forces and moments are the surface distributions of static pressure, $p(x, y, z, t)$, acting normal to the surface, and shear stress, $\tau(x, y, z, t)$, along the relative flow direction. When integrated over the entire exposed surface of the vehicle, the pressure and shear stress distributions give rise to the aerodynamic force vector, $\mathbf{F_a}(t)$, and aerodynamic torque vector, $\boldsymbol{\tau}_a(t)$, as functions of the relative flow velocity far upstream, (v, α, β), called the *freestream velocity*, the linear flow acceleration, $(\dot{v}, \dot{\alpha}, \dot{\beta})$, the angular velocity of the vehicle, (P, Q, R), as well as thermodynamic properties of the freestream, namely density, ρ, static temperature, $\bar{T}$, dynamic viscosity, $\bar{\mu}$, and specific-heat ratio, γ. When the flow does not have any intermolecular voids, it is regarded to be a continuum governed by nondimensional parameters such as the *Mach number*, $\mathcal{M} = v/a$, where a is the speed of sound far upstream, and the *Reynolds number*, $Re = (\rho v\ell)/\bar{\mu}$, with ℓ being a characteristic length of the flow. When

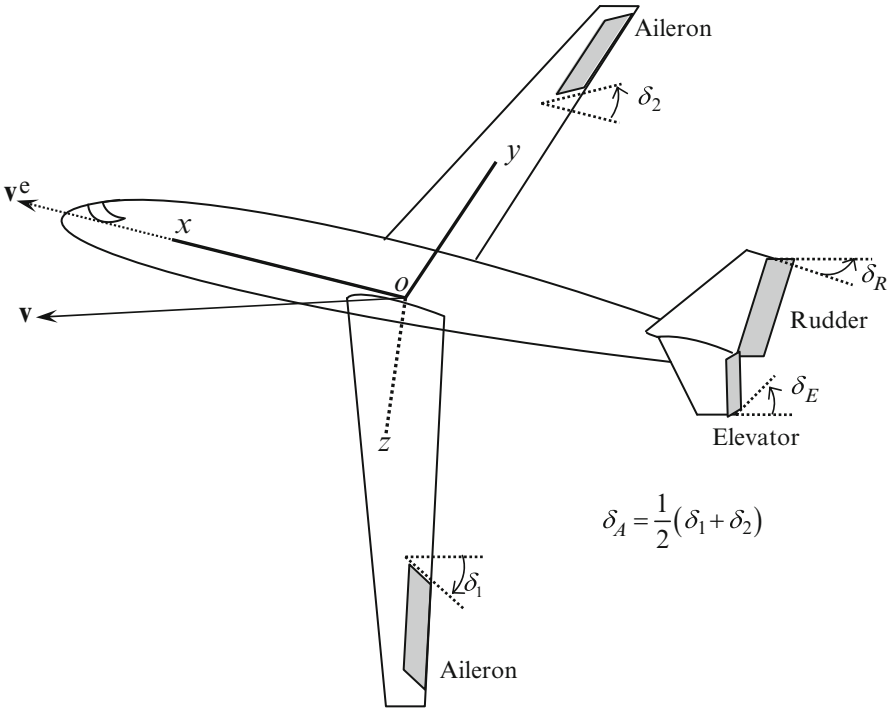


Fig. 2.13 The stability axes, $(oxyz)$, and the aerodynamic control surfaces

the characteristic length becomes either comparable to or larger than the *mean free-path*, $\bar{\lambda}$, of the gas molecules, we have a rarefied flow in which the governing flow parameter is *Knudsen number* given by $Kn = \bar{\lambda}/\ell$. Based upon Knudsen number, the flow regimes are commonly classified as (a) *Free-molecular* flow, $(Kn \geq 10)$, (b) *Transition flow* flow, $(0.01 \leq Kn \leq 10)$, (c) *Continuum* flow, $(Kn \leq 0.01)$.

2.7.2.1 Stability Axes, Small Perturbations, and Control Surfaces

When the nominal trajectory is a steady maneuver, the aerodynamic force and torque are balanced by other effects, and thus have constant equilibrium values, $\mathbf{F_a}^e, \boldsymbol{\tau}^e_{\mathbf{a}}$. In such a case, a special body-fixed frame – called *stability axes* – is generally employed having one axis initially aligned with the nominal flight direction (Fig. 2.13). Furthermore, as all atmospheric flight vehicles have at least one plane of symmetry, (oxz), the equilibrium velocity vector lies in the plane of symmetry and we have

$$\alpha^e = 0$$

$$\beta^e = 0,$$

or, $\mathbf{v}^e = U^e\mathbf{i}$.

A small perturbation, $\Delta\mathbf{v}(t), \Delta\boldsymbol{\omega}(t)$, that begins acting at $t = 0$ about the equilibrium point, $\mathbf{v}^{\mathrm{e}}, \boldsymbol{\omega}^{\mathrm{e}}$, causes small changes in the flow speed, $u(t)$, angle-of-attack, $\alpha(t)$, sideslip angle, $\beta(t)$, and body-rates, $p(t), q(t), r(t)$, measured with respect to the stability axes:

$$\begin{aligned} \mathbf{v} &= \mathbf{v}^{\mathrm{e}} + \Delta\mathbf{v} \\ \boldsymbol{\omega} &= \boldsymbol{\omega}^{\mathrm{e}} + \Delta\boldsymbol{\omega}, \end{aligned} \tag{2.119}$$

or,

$$\begin{aligned} U &= U^{\mathrm{e}} + u \\ V &= U^{\mathrm{e}}\beta \\ W &= U^{\mathrm{e}}\alpha \\ P &= P^{\mathrm{e}} + p \\ Q &= Q^{\mathrm{e}} + q \\ R &= R^{\mathrm{e}} + r. \end{aligned}$$

As mentioned earlier, the aerodynamic force and torque depend not only upon the linear and angular velocity perturbations, $u, \alpha, \beta, p, q, r$, but also on the linear acceleration perturbations, $\dot{u}, \dot{\alpha}, \dot{\beta}$, due to unsteady flow. These latter perturbations cause a delay (or lag) in the onset of aerodynamic changes and are thus termed *aerodynamic inertia* effects.

The aerodynamic force and torque are controlled by small deflections of control surfaces. Generally, the aerodynamic forces and moments obey linear relationships with control surface deflections. Furthermore, the control surfaces are driven by actuators with a dynamic response about ten times faster than that of the vehicle's rigid body dynamics. Therefore, flow unsteadiness due to control surface deflections can often be neglected when considering rigid body aerodynamics. Hence, a quasi-steady approximation of aerodynamic loads, linearly varying with control surface deflection, is usually employed.[4]

The aerodynamic control surface for longitudinal control is the *elevator* with rotation axis parallel to oy (Fig. 2.13). The elevator can be mounted either forward or behind oy, usually on the trailing edge of a larger stabilizing surface in the oxy plane, as shown in Fig. 2.13. Elevator deflection, δ_{E}, creates a forward force, $X_{\delta}\delta_{\mathrm{E}}$, downforce, $Z_{\delta}\delta_{\mathrm{E}}$, and a control pitching moment, $M_{\delta}\delta_{\mathrm{E}}$.

A pair of control surfaces in the oxy plane, located symmetrically about the axis ox and deflected in mutually opposite directions by angles δ_1 and δ_2 (Fig. 2.13) are

[4]When the vehicle's structural flexibility is taken into account, it is necessary to include the unsteady aerodynamic effects of control surface deflections, i.e., dependence of aerodynamic forces and moments on deflection rates, $\dot{\delta}_{\mathrm{E}}, \dot{\delta}_{\mathrm{A}}, \dot{\delta}_{\mathrm{R}}$.

called *ailerons* that are used as roll control devices. The *aileron deflection* is the average of the two separate deflections, $\delta_A = (\delta_1 + \delta_2)/2$, and is designed such that a control rolling moment, $L_A\delta_A$, is produced along with a much smaller, undesirable yawing moment, $N_A\delta_A$, called adverse aileron yaw. A control surface mounted on a fin in the oxz plane behind the oy axis is called the *rudder*, and is shown in Fig. 2.13. A rudder deflection, δ_R, creates a sideforce, $Y_R\delta_R$, control yawing moment, $N_R\delta_R$, and a much smaller rolling moment, $L_R\delta_R$. Rudder is primarily used to correct a lateral flight asymmetry. For a rocket's flight through the atmosphere, the control surfaces consist of movable fins that can exert rolling, pitching, and yawing moments by simultaneous deflection. In addition to the aerodynamic controls, some vehicles have a thrust vectoring capability that can be used for applying control forces and moments.

We note that due to inertia coupling, rotational kinetics produces equilibrium solutions whenever any two body rates vanish for a zero net external torque. Hence, we assume that for equilibrium, only one of the body rates is nonzero, and all the aerodynamic moments are zeros. Furthermore, the aerodynamic moments in atmospheric flight are much larger than the typical internal rotor torques, which are commonly neglected in rotational dynamics. All of the assumptions in this paragraph cause the aerodynamic perturbations in the plane of symmetry oxz – called the *longitudinal plane* (or *pitch plane*) – to be decoupled from those taking place out-of-plane. Thus, the pitching moment, M, forward force, X, and downforce, Z, depend only on the longitudinal perturbations, $u, \alpha, \dot{u}, \dot{\alpha}, q$. On the other hand, the rolling moment, L, sideforce, Y, and yawing moment, N, depend only on the lateral perturbations, $\beta, \sigma, \dot{\beta}, p, r$.[5]

On expanding the aerodynamic forces and moments in a Taylor series about the equilibrium point and assuming small distubances, we have the following linear approximations:

$$\begin{aligned} X &= X^e + X_u u + X_\alpha \alpha + X_{\dot{u}}\dot{u} + X_{\dot{\alpha}}\dot{\alpha} + X_q q + X_\delta \delta_E \\ Z &= Z^e + Z_u u + Z_\alpha \alpha + Z_{\dot{u}}\dot{u} + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_\delta \delta_E \\ M &= M^e + M_u u + M_\alpha \alpha + M_{\dot{u}}\dot{u} + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_\delta \delta_E, \end{aligned} \tag{2.120}$$

and

$$\begin{aligned} Y &= Y^e + Y_\beta \beta + Y_\sigma \sigma + Y_{\dot{\beta}}\dot{\beta} + Y_p p + Y_r r + Y_R \delta_R \\ L &= L^e + L_\beta \beta + L_\sigma \sigma + L_{\dot{\beta}}\dot{\beta} + L_p p + L_r r + L_A \delta_A + L_R \delta_R \\ N &= N^e + N_\beta \beta + N_\sigma \sigma + N_{\dot{\beta}}\dot{\beta} + N_p p + N_r r + N_A \delta_A + N_R \delta_R, \end{aligned} \tag{2.121}$$

[5] Missiles rolling at a high rate can have a significant aerodynamic coupling between longitudinal and lateral aerodynamics. However, as such motions are inherently nonlinear, they are beyond our present scope.

where the coefficients stand for partial derivatives with respect to the subscripted variable at equilibrium, e.g.,

$$X_u = \left.\frac{\partial X}{\partial u}\right|_{u=0}. \tag{2.122}$$

The coefficients related to the motion variables, $u, \alpha, \beta, \dot{u}, \dot{\alpha}, \dot{\beta}, \sigma, p, q, r$, are called *stability derivatives* as they influence the stability of the equilibrium point, while those related to the control inputs, $\delta_{\mathrm{E}}, \delta_{\mathrm{A}}, \delta_{\mathrm{R}}$, are termed *control derivatives*.

The aerodynamic force and torque are usually nondimensionalized as follows:

$$\begin{aligned}\mathbf{F_a} &= \frac{1}{2}\rho v^2 S \mathbf{C_F}\\ \boldsymbol{\tau}_\mathbf{a} &= \frac{1}{2}\rho v^2 S \ell \mathbf{C}_{\boldsymbol{\tau}},\end{aligned} \tag{2.123}$$

where $\mathbf{C_F}, \mathbf{C}_{\boldsymbol{\tau}}$ are nondimensional force and torque coefficient vectors, respectively, whose functional dependence on the nondimensional flow variables, $\alpha, \beta, Kn, \mathcal{M}, Re$, is obtained either experimentally in a wind-tunnel or by a computational fluid dynamics (CFD) model. The external shape of the vehicle is primarily important for such nondimensional relationships. The nondimensional lift, C_{L}, drag, C_{D}, and sideforce, C_y, coefficients are the components of $\mathbf{C_F}$ resolved along the wind axes, while the rolling moment, C_ℓ, pitching moment, C_m, and yawing moment, C_n coefficients are the components of $\mathbf{C}_{\boldsymbol{\tau}}$ resolved along the body axes. As the parameters governing the flow (i.e., $v, \rho, a, Kn, \mathcal{M}, Re$) vary little in the vicinity of a given equilibrium flight condition, they are assumed constants in a stability and control analysis, which only requires small disturbances from equilibrium, as discussed earlier. Thus, for the purpose of control system design, the aerodynamic system can be modeled as a linear system.

Example 2.1. Let us revisit the aircraft level flight model presented in Example 1.3. With airspeed, $v(t)$, azimuth, $\psi(t)$, latitude, $\delta(t)$, and longitude, $\lambda(t)$, the equations of translational motion are the following:

$$\begin{aligned}\dot{\delta} &= \frac{v\cos\psi + v_{\mathrm{w}}\cos\psi_{\mathrm{w}}}{R_0 + h}\\ \dot{\lambda} &= \frac{v\sin\psi + v_{\mathrm{w}}\sin\psi_{\mathrm{w}}}{(R_0 + h)\cos\delta}\\ v\dot{\psi} &= g\tan\sigma + \frac{1}{2}\Omega^2(R_0 + h)\sin\psi\sin 2\delta - 2\Omega v\cos\psi\cos\delta\\ m\dot{v} &= T(v) + \Delta T - \frac{1}{2}\rho v^2 S\left(C_{\mathrm{D}0} + \frac{4Km^2g^2\sec^2\sigma}{\rho^2 S^2 v^4}\right)\\ &\quad -\frac{1}{2}m\Omega^2(R_0 + h)\cos\psi\sin 2\delta,\end{aligned} \tag{2.124}$$

where R_0 (Earth's radius), Ω (Earth's rotational rate), ρ (atmospheric density), S (wing area), C_{D0} (parasite drag-coefficient), K (lift-dependent drag factor), and g (acceleration due to gravity) are constants. The mass, $m(t)$, is a specified function of time. The engine thrust, $T(v)$, is a known function of speed at a constant throttle setting, while $\Delta T(t)$ is a control input applied to the aircraft by changing the throttle setting. Another control input is the bank angle, $\sigma(t)$, measured from the local horizontal plane. Both the inputs are applied by either a pilot or an automatic control system. The wind speed, $v_w(t)$, and wind azimuth, $\psi_w(t)$, are stochastic (random) inputs, regarded as disturbances. It is assumed that the angle-of-attack (and thus lift) is adjusted automatically in order to maintain level flight, and all turns are automatically coordinated by rudder input.

The thrust actuating mechanism is modeled by the following second-order differential equation:

$$a_1 \frac{d^2 \Delta T}{dt^2} + a_2 \frac{d\Delta T}{dt} + a_3 \Delta T = u_1, \tag{2.125}$$

where $a_1, \ldots, a_3$ are constant coefficients representing engine characteristics. The banking actuator comprising aileron servomotor and aircraft rolling motion is described by the following differential equations:

$$\begin{aligned} a_4 \ddot{\sigma} + a_5 \dot{\sigma} &= \delta_A \\ a_6 \ddot{\delta}_A + a_7 \dot{\delta}_A + a_8 \delta_A &= u_2, \end{aligned} \tag{2.126}$$

where $a_4, \ldots, a_8$ are constant coefficients representing aileron servo and aircraft rolling characteristics. The electrical inputs, $u_1(t), u_2(t)$, are supplied by either the pilot or the autopilot according to the desired flight path.

Example 2.2. Automatic control of rotation about the longitudinal axis (roll) is an important example of flight control systems. Several flight applications result in a decoupling of roll from other degrees of freedom (pitch, plunge, forward motion, yaw, and sideslip). While there is a significant aerodynamic interaction among roll, yaw, and sideslip in aircraft, such a coupling does not exist in spacecraft and rockets. Instead, rolling missiles and spinning spacecraft have an inertia coupling between pitch and yaw, as well as between plunge and sideslip. Aerodynamically and inertially coupled dynamics require multivariable control design methods to be considered later. However, there are many situations in which roll can be controlled as a single-axis rotation. Examples include roll-rate regulation of spin stabilized missiles, axisymmetric spacecraft, fighter aircraft, and bank-to-turn missiles.

The simplest roll autopilot is for nonspinning rockets (such as launch vehicles) where an opposing rolling control moment is provided for bringing any roll-rate disturbance to zero. As roll-rate is the coupling between pitch and yaw motions, keeping roll-rate zero uncouples the two transverse rotations, thereby enabling a separate design of pitch and yaw autopilots. Hence, a classical attitude control system can be designed based upon separate roll, pitch, and yaw loops, provided that roll regulator works much faster than either the pitch or the yaw autopilots.

Roll control of rockets that are not required to bank for turning normally consists of a regulator that quickly drives an initial roll-rate to zero. In such cases, roll-rate, $p(t)$, is fed back through a rate gyro (see later) to generate a control rolling moment via control surfaces (fins or exhaust vanes) deflection, $\delta_{\mathrm{f}}(t)$. Consider the following first-order roll-rate dynamics:

$$J_{xx}\dot{p} = L_p p + L_f \delta_{\mathrm{f}}, \tag{2.127}$$

whose unforced response to initial roll-rate disturbance, $p(0)$, is given by:

$$p(t) = p(0)\mathrm{e}^{tL_p/J_{xx}}. \tag{2.128}$$

As the damping in roll ($L_p < 0$) is generally quite small in magnitude for rockets, the roll-rate proportional feedback increases the damping of the overall control system,

$$\delta_{\mathrm{f}} = -Kp, \tag{2.129}$$

where $K > 0$, resulting in the following closed-loop initial response:

$$p(t) = p(0)\mathrm{e}^{t(L_p - KL_f)/J_{xx}}. \tag{2.130}$$

The feedback gain, K, is suitably adjusted in order to achieve a desired settling time, t_{s}, for the roll-rate to settle to within $\pm 2\%$ of the initial disturbance:

$$t_{\mathrm{s}} = -\frac{J_{xx}}{L_p - KL_f}\ln(0.02). \tag{2.131}$$

Generally, the deflection of the control surfaces is limited by aerodynamic effectiveness, $|\delta_{\mathrm{f}}| \leq \delta_{\max}$. In order to ensure that the maximum deflection is not exceeded in normal operation, the regulator gain is selected such that

$$K \leq \frac{\delta_{\max}}{|p(0)|}. \tag{2.132}$$

Of course, such a restriction can be imposed only if an apriori knowledge of expected roll disturbance magnitude, $|p(0)|$, is available by a careful analysis.

2.8 Flight Sensors

Sensors are an integral part of every automatic control system because they supply information about the actual state of the system. In flight applications, sensors are necessary for measuring both translational and rotational motion of the flight vehicle relative to an reference frame approximated to be inertial, as well as that

relative to the atmosphere. Therefore, the flight can be broadly classified as either (a) motion sensors or (b) flow sensors. In motion sensors we have Doppler radars, accelerometers, gyroscopes, horizon, and sun/star scanners, while the flow sensors include airspeed, vertical speed, angle-of-attack, and angle of sideslip sensors.

Virtually all sensors work on the general principle of producing an electrical voltage proportional to the motion variable being measured, and are thus called *voltage transducers*. By calibrating the voltage output, e_y, against the motion variable, y, (i.e., displacement, speed, or acceleration) one can have a linear functional relationship of the following type:

$$e_y(t) = ay(t) + b, \tag{2.133}$$

where a, b are calibration constants. The range of validity of (2.133), $y_a \leq y \leq y_b$ is called the *linear range* of the sensor. Outside the linear range, sensors suffer from nonlinearities like hysteresis, deadzone, and saturation, and therefore cease to be useful. Clearly, a good sensor is the one that has the largest possible linear range.

Being based upon electromagnetism, most flight sensors have at their heart the Faraday's law of induction and are generally of the noncontact type for eliminating noise due to friction. For measuring displacement, *Hall effect* and *linear variable differential transformer* (LVDT) transducers can sense the varying direction of a magnetic field relative to a fixed plate or coil, while angular rate measurement is possible by *incremental encoders* and *back e.m.f* based tachometers. Variations of these basic sensing devices can be found in all flight sensors, such as accelerometers, flow sensors, and rate-integrating gyros that produce displacement of a mass or spring, and rate gyros that produce angular rate outputs. An alternative angular position and rate measurement are provided by the *optical encoders*, which have a rotating transparent disc with some coded opaque areas. A light source and a photo detector "read" the optical pattern resulting from the disc's position at a given instant, which is then translated by a microprocessor to determine the angle and speed of the shaft. As they are based upon digital signals, the optical encoders are less prone to high-frequency noise that plagues analog sensors.

The traditional attitudinal sensors in flight vehicles are gyroscopes that are based upon the torque produced by displacing the axis of a spinning rotor. The study of gyroscopic motion is termed *gyrodynamics* and is our next topic of discussion. Although state-of-the art "gyros" have moved away from rotors and are instead based upon tuning forks, oscillating crystals, and laser "rings" for better ruggedness and fidelity, it is instructive to study the physical characteristics of a classical gyroscope.

2.8.1 Gyrodynamics

Rotors are widely used as both actuators and sensors in flight dynamic plants. They are also present in the form of airbreathing engines in an aircraft, and can affect

its stability and control. Spin stability of an axisymmetric rotor (wheel) – termed gyroscopic stability – is the basic principle behind a gyroscope (in short, gyro), which is a device commonly employed in flight control applications.

A gyroscope consists of a wheel spinning at a constant rate[6] mounted on the flight vehicle in such a way that the wheel's spin axis can pivot in some special directions called *gimbals*. If there is only one gimbal, the wheel can tilt in only one direction and such a gyroscope is called a *single-axis gyro*. If there are two gimbals, the wheel can tilt with respect to the vehicle in two directions and the resulting gyro is termed a *two-degree-of-freedom gyro*. Lastly, there is the *fully gimbaled gyro* consisting of three gimbals, in which the wheel's axis always maintains a fixed direction irrespective of the vehicle's orientation.

When electric motors are used to change the wheel's spin axis relative to the vehicle, the gyroscope acts as an actuator. However, if the gimbals are restrained by springs and viscous dampers, the wheel experiences a torque resulting in gimbal displacements that can be measured by Hall effect/LVDT/optical encoder devices. Thus, a gyro becomes a sensing device for the inertial angular velocity of the flight vehicle and is at the heart of an *inertial measurement unit* (IMU). We will next consider the typical gyroscopic applications in automatic flight control.

2.8.1.1 Rate Gyro

Let us consider a gyroscope that is used to sense a single-axis rotation rate of a flight vehicle. As shown in Fig. 2.14, the single degree-of-freedom rate gyro consists of a rotor spinning relative to a gimbal at a constant angular momentum, H_r, driven by a *servomotor* about the axis *ox*. The gimbal axis, *oy*, can turn by a small angle, $\theta(t)$, relative to the flight vehicle through a restraining mechanism consisting of a linear spring of stiffness, k, and a linear, viscous damper of damping constant, c. The flight vehicle's rotation about the body axis, *oz*, by an inertial angle, ψ, and rate, $\dot{\psi}$, is to be sensed.

As the angular speed of the rotor relative to the gimbal is constant, and the direction of its angular momentum, $\mathbf{H_r}$, does not change relative to the gimbal (due to a rigid construction), there is no change in the rotor's angular momentum relative to the gimbal. Thus, the rate of change of $\mathbf{H_r}$ in the inertial space due to the combined rotation of the vehicle and gimbal is given by:

$$\begin{aligned}\dot{\mathbf{H}}_\mathbf{r} &= \boldsymbol{\omega}_\mathbf{r} \times \mathbf{H}_\mathbf{r} = (\dot{\theta}\mathbf{j} + \dot{\psi}\mathbf{k}) \times H_\mathrm{r}(\cos\theta\mathbf{i} - \sin\theta\mathbf{k}) \\ &= H_\mathrm{r}[-\dot{\theta}(\cos\theta\mathbf{k} + \sin\theta\mathbf{i}) + \dot{\psi}\cos\theta\mathbf{j}]. \end{aligned} \tag{2.134}$$

[6]The wheel's speed is maintained constant by a feedback control system called *servomotor*, which works by taking an angular speed feedback from a tachometer (or an incremental angle encoder) as a sensor, and applies a torque by a direct-current(DC) motor as the actuator.

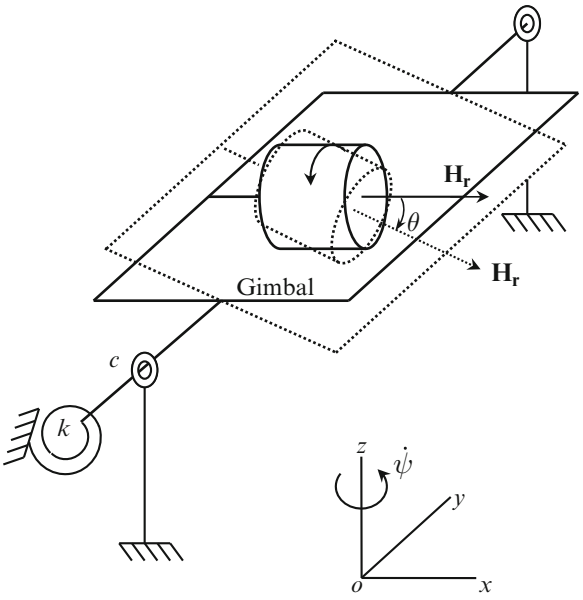


Fig. 2.14 A single degree-of-freedom rate gyroscope

This change of angular momentum results in a torque about the gimbal axis, oy, as well as small transverse torques about ox and oz. While the axial torque causes gimbal rotation, the transverse torques are absorbed by the rotor-gimbal bearing. As both gimbal angle, θ, and gimbal rate, $\dot{\theta}$, are kept small by the restraining spring and damper, we can approximate (2.134) by the linear relationship,

$$\dot{\mathbf{H}}_\mathbf{r} \simeq H_\mathrm{r}\dot{\psi}\mathbf{j}. \tag{2.135}$$

By Newton's second-law, the rate of change of the rotor's angular momentum is equal to the net torque experienced by the rotor, which by Newton's third-law, is equal and opposite to the torque applied by the rotor on the gimbal. Therefore, the gimbal's linearized dynamical equation can be written as follows:

$$J\ddot{\theta}(t) + c\dot{\theta}(t) + k\theta(t) = -H_\mathrm{r}\dot{\psi}(t), \tag{2.136}$$

where J is the moment of inertia of the gimbal and rotor assembly about the axis oy. Equation (2.136) has the following equilibrium solution, $\theta(t) = \theta_\mathrm{e}$, in the steady-state ($t \to \infty$), obtained by letting $\ddot{\theta} = \dot{\theta} = 0$:

$$\theta_\mathrm{e} = -\frac{H_\mathrm{r}\dot{\psi}}{k}, \tag{2.137}$$

which implies a gimbal angle proportional to the vehicle's rotation rate. For this reason, the gyroscope of Fig. 2.14 is called a *rate gyro*, as it can be calibrated to measure a vehicle's rotational rate about the input axis. Two (or more) rate gyros mounted on mutually perpendicular body axis can provide information about a vehicle rotating steadily about multiple body axes. A sudden change in the vehicle's rate, however, will take some time to be registered as the equilibrium gimbal angle output. The time taken to approximately reach the steady-state for a given change in the vehicle's rate depends upon the damping constant, c, as well as the moment of inertia, J, while the equilibrium value of the gimbal angle, (2.137), depends only upon the ratio of the rotor's angular momentum, H_r with the spring stiffness, k. By adjusting this latter ratio, the rate gyro can be made more (or less) sensitive to the vehicle's rate. On the other hand, by adjusting the damping constant, the gyro dynamics can be speeded up, or slowed down, making its respond quickly (or slowly) to change in vehicle's rate.

A voltage transducer (such as Hall effect sensor or LVDT) of constant gain, K_θ, is invariably employed to convert the output angle, $\theta(t)$, into a corresponding voltage, $e_\theta(t) = K_\theta\theta(t)$, which can be easily fed back, or otherwise manipulated by suitable electrical networks according to a specific control law. The transducer gain, K_θ, can be selected based on the desired sensitivity.

Taking the Laplace transform of (2.136) with zero initial conditions ($\theta(0) = \dot{\theta}(0) = 0$ and $\psi(0) = \dot{\psi} = 0$), we have the following *transfer function* (Chap. 3) for the rate gyro, relating the Laplace transforms of the gimbal angle (output), $\Theta(s)$, and the vehicle's inertial rotational rate (input), $s\Psi(s)$:

$$\frac{e_\theta(s)}{s\Psi(s)} = -\frac{K_\theta H_\mathrm{r}}{Js^2 + cs + k}. \tag{2.138}$$

As the rate gyro dynamics is usually much faster than the vehicle's rotation, the transfer function given by (2.138) is practically replaced by the following rate gyro gain constant:

$$K_{rg} = \frac{K_\theta H_\mathrm{r}}{k}. \tag{2.139}$$

2.8.1.2 Rate-Integrating (Displacement) Gyro

If the restraining spring is removed from the rate gyro, the transfer function of the resulting mechanism becomes

$$\frac{\Theta(s)}{s\Psi(s)} = -\frac{H_\mathrm{r}}{s(Js + c)}. \tag{2.140}$$

or,

$$\frac{\Theta(s)}{\Psi(s)} = -\frac{H_\mathrm{r}}{Js + c}. \tag{2.141}$$

It is clear that the resulting gyro is capable of measuring either the vehicle's angular rate, $\dot{\psi}$, through an integral action or its angular displacement, ψ, by a proportional, steady-state, equilibrium gimbal angle, θ_e,

$$\theta_e = -\frac{H_r \psi}{c}. \tag{2.142}$$

Therefore, such a gyro is called a *rate-integrating* gyro (or *displacement gyro*). The rate-integrating gyro can be made more (or less) sensitive to the vehicle's rotation by adjusting the viscous damping constant, c. Additionally, the linear transducer gain, K_θ, is used to adjust the sensitivity of the output voltage, $e_\theta(t)$.

For a gyro with a fast response, the transfer function is replaced by a scaled integral when the rate is input,

$$\frac{\Theta(s)}{s\Psi(s)} = \frac{K_{rig}}{s} = -\frac{K_\theta H_r}{cs}, \tag{2.143}$$

or by a constant gain when the angular displacement is input,

$$\frac{\Theta(s)}{\Psi(s)} = K_{rig} = -\frac{K_\theta H_r}{c}. \tag{2.144}$$

The first-order transfer function of the displacement gyro implies an exponential gimbal angle output for an indicial change in the vehicle's attitude,

$$\theta(t) = -\frac{H_r}{c}\left(1 - e^{-\frac{c}{J}t}\right). \tag{2.145}$$

The time-lag with which the gyro can track a changing vehicle attitude is thus given by the time-constant, $T = \frac{J}{c}$. By increasing c, the speed of response is increased, but the sensitivity is reduced. Hence, a balance must be struck between the speed and the sensitivity of a displacement gyro.

The integral action provided by the rate-integrating gyro is a valuable feature in reducing the steady-state error in a closed-loop system, and enables PID controller implementation (Chap. 3). A combination of rate and rate integrating gyros can handle most practical attitude control tasks.

Example 2.3. Consider a rate gyro with the following characteristics: $H_r = 10^4$ g cm^2/s, $J = 34$ g cm^2, $k = 3.03 \times 10^5$ g cm^2/s^2, and $c = 5{,}000$ g cm^2/s. The gimbal angle response is converted into electrical voltage, e_θ, by a linear voltage transducer with a constant gain 100 V/rad.

(a) Evaluate the gimbal angle response for the rate gyro to a step change (Chap. 3) in the vehicle's angular rate by 0.1 rad/s.
(b) By removing the spring, i.e., making $k = 0$ the rate gyro is converted into a rate-integrating gyro. The transducer gain for the rate-integrating gyro is reduced to 10 V/rad. Find the response of the rate-integrating gyro to a step change in the vehicle's angular orientation by 0.1 rad.

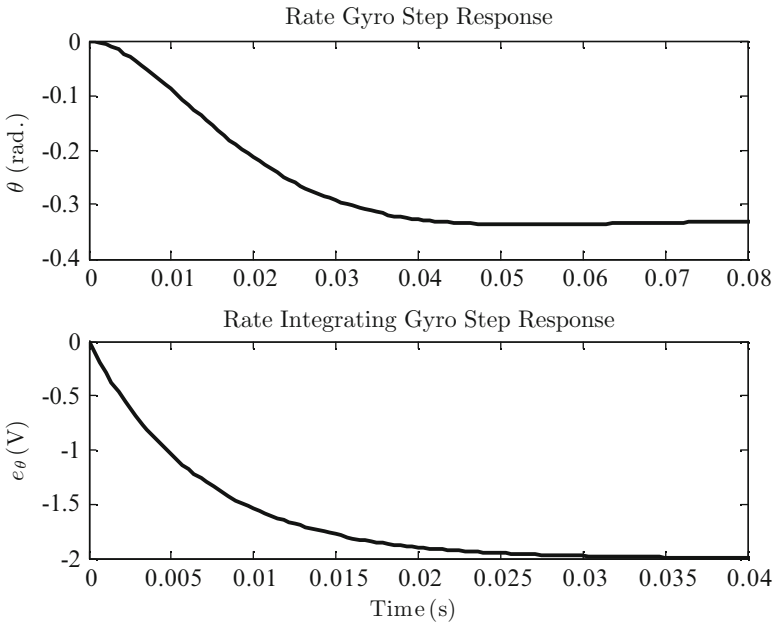


Fig. 2.15 Individual responses of rate and rate-integrating gyros to a step change in the vehicle's angular orientation by 0.1 rad

The necessary computations are carried out using the MATLAB-CST commands *tf* and *step* as follows:

```
>> num=[0 -10000];den=[34 5000 3.03e5];sys=tf(num,den);[y1,t1]=step(sys);
>> num=[0 -10000];den=[34 5000];sys=tf(num,den);[y2,t2]=step(sys);
```

The two step responses are compared in Fig. 2.15. Note that the rate-integrating gyro has a settling-time of 0.04 s that is about half that of the rate gyro. Application of a gyro is restricted to sensing vehicle dynamics about ten times slower than the settling-time of the gyro.

A common single-axis gyro is the *directional gyro* for measuring a vehicle's yaw angle (heading angle) and thus the direction of flight in the horizontal plane. Of course, the directional gyro can measure the heading accurately only if the angular displacements of the vehicle about the other two axes (roll and pitch) are small.

2.8.1.3 Two-Degree-of-Freedom Gyroscope

Consider a wheel mounted on a case through a bi-axial mechanism called *inner gimbal* and *outer gimbal* – as shown in Fig. 2.16 – that allows a bi-directional variation of the wheel's spin axis relative to the case. Such a gyro is termed a two-degree-of-freedom gyroscope. The case, in turn, is rigidly attached to a flight

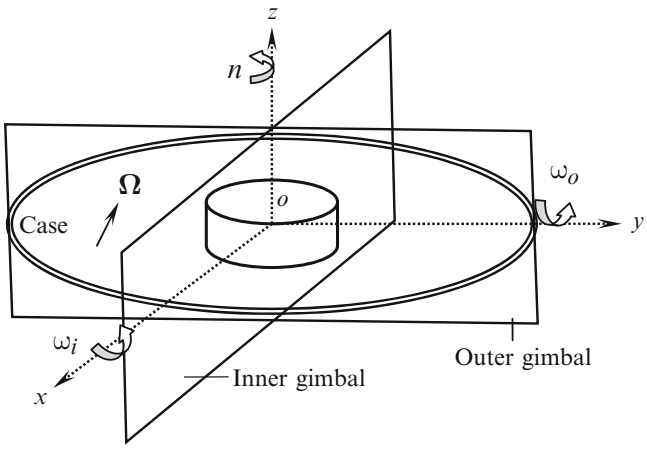


Fig. 2.16 The two-degree-of-freedom gyroscope

vehicle that is rotating with an angular velocity, $\boldsymbol{\Omega}$, with respect to an inertial frame of reference. The wheel is spinning at a rate, n, relative to the inner gimbal about the axis of symmetry, oz, as shown in Fig. 2.16. As the gimbals are always parallel to the wheel's transverse axes, ox, and oy, respectively, we call the wheel-fixed principal body frame $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ the *gimbal axes*. The angular velocity of the inner gimbal with respect to the outer gimbal is $\omega_{\mathrm{i}}\mathbf{i}$, and $\omega_{\mathrm{o}}\mathbf{j}$ is that of the outer gimbal relative to the case (vehicle).

The wheel's inertial angular velocity can be expressed as follows:

$$\boldsymbol{\omega} = n\mathbf{k} + \omega_{\mathrm{i}}\mathbf{i} + \omega_{\mathrm{o}}\mathbf{j} + \boldsymbol{\Omega}, \tag{2.146}$$

while the net angular momentum of the wheel is the following:

$$\mathbf{h} = nJ_{zz}\mathbf{k} + J_{xx}(\omega_{\mathrm{i}}\mathbf{i} + \omega_{\mathrm{o}}\mathbf{j}) + \mathsf{J}\boldsymbol{\Omega}, \tag{2.147}$$

where

$$\mathsf{J} = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{xx} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix}, \tag{2.148}$$

and

$$\boldsymbol{\Omega} = \Omega_x\mathbf{i} + \Omega_y\mathbf{j} + \Omega_z\mathbf{k} \tag{2.149}$$

is the flight vehicle's angular velocity resolved in the gimbal axes. The following torque is experienced by the wheel due to its changing angular momentum (2.8):

$$\begin{aligned} \mathbf{M}^{\mathrm{w}} &= M^{\mathrm{w}}{}_x\mathbf{i} + M^{\mathrm{w}}{}_y\mathbf{j} + M^{\mathrm{w}}{}_z\mathbf{k} \\ &= \frac{\mathrm{d}\mathbf{h}}{\mathrm{d}t} = \dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}, \end{aligned} \tag{2.150}$$

where

$$\dot{\mathbf{h}} = J_{xx}\left[\left(\dot{\Omega}_x + \dot{\omega}_i\right)\mathbf{i} + \left(\dot{\Omega}_y + \dot{\omega}_o\mathbf{j}\right)\right] + J_{zz}\left(\dot{\Omega}_z\right)\mathbf{k}. \tag{2.151}$$

On taking the cross product and collecting terms in (2.150), we have

$$M^{\mathrm{w}}{}_x = J_{xx}\left(\dot{\Omega}_x + \dot{\omega}_i\right) + \left(J_{zz} - J_{xx}\right)\left(\Omega_y\Omega_z + n\Omega_y + \omega_{\mathrm{o}}\Omega_z + n\omega_{\mathrm{o}}\right) \tag{2.152}$$

$$M^{\mathrm{w}}{}_y = J_{xx}\left(\dot{\Omega}_y + \dot{\omega}_o\right) + \left(J_{xx} - J_{zz}\right)\left(\Omega_x\Omega_z + n\Omega_x + \omega_{\mathrm{i}}\Omega_z + n\omega_{\mathrm{i}}\right) \tag{2.153}$$

$$M^{\mathrm{w}}{}_z = J_{zz}\dot{\Omega}_z. \tag{2.154}$$

If the vehicle is spinning at a constant angular velocity that is small in comparison with the wheel's, one can approximate the torque experienced by the wheel as follows:

$$M^{\mathrm{w}}{}_x \simeq \left(J_{zz} - J_{xx}\right)\left(n\Omega_y + n\omega_{\mathrm{o}}\right) \tag{2.155}$$

$$M^{\mathrm{w}}{}_y \simeq \left(J_{xx} - J_{zz}\right)\left(n\Omega_x + n\omega_{\mathrm{i}}\right) \tag{2.156}$$

$$M^{\mathrm{w}}{}_z \simeq 0. \tag{2.157}$$

By having springs and dampers at the two gimbal locations, the torque components $M^{\mathrm{w}}{}_x, M^{\mathrm{w}}{}_y$, and gimbal rates, $\omega_{\mathrm{i}}, \omega_{\mathrm{o}}$, can be made to produce respective angular displacements, that can be measured and calibrated in a manner similar to the single-axis rate gyro. Thus, the vehicle's spin rates, Ω_x, Ω_y, are obtained from (2.134) and (2.135). However, the third angular velocity component, Ω_z, along the wheel axis cannot be measured. By removing the gimbal restraining springs, the vehicle's angular displacements (rather than the rates) can be measured in a manner similar to the rate-integrating gyro.

A *vertical gyro* is a special two-degree of freedom gyro mounted in such a way that its spin axis is always vertical. Consequently, it can measure pitch and roll angular displacements (or rates) of the vehicle (but not yaw). Such a gyro is also called an *artificial horizon* in aircraft flight displays and provides a basic reference for maintaining a wings-level (unbanked), horizontal attitude. The *Sperry* autopilot of 1909 was based on a vertical gyro, and was tasked principally with maintaining a wings-level attitude during cruise of the early aircraft. Its utility was demonstrated in the long-distance flights by many aviation pioneers, often in bad weather, or at night (such as the solo, transatlantic flight undertaken by Lindbergh in 1927).

The vertical gyro output voltages can be amplified, and used to directly drive electric motors on mutually perpendicular axes, for either moving the control surfaces or tilting a rocket nozzle for thrust deflection. Therefore, a vertical gyro can perform the important role of a two-axis attitude controller. Examples of the first such gyroscopic flight control systems were the unmanned *V-1* "flying bomb" and the *V-2* rocket of the 1930s and 40s, and have continued to be useful in the present age in the form of *inertial navigation systems* for airliners, unmanned aircraft, long-range missiles, and spacecraft.

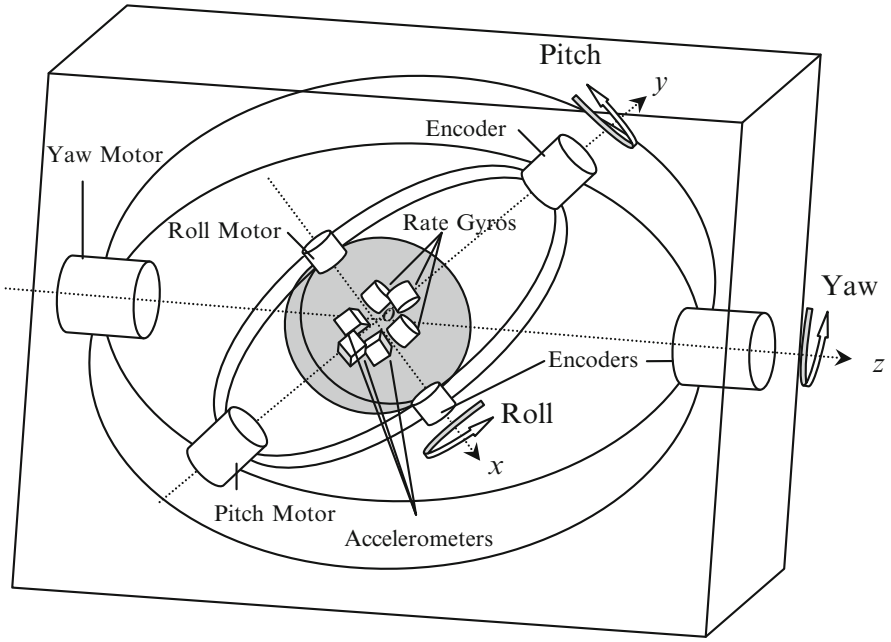


Fig. 2.17 Fully gimbaled, gyro stabilized platform of an inertial measurement unit (IMU)

2.8.2 Inertial Measurement Units

Inertial measurement units (IMU) are a set of gyros and accelerometers that produce motion variables as outputs. Traditionally, the IMU consists of a fully gimbaled gyro platform on which the motion sensors are mounted. It contains three gimbal axes, the inner gimbal, *ox*, the middle gimbal, *oy*, and the outer gimbal, *oz*, which are typically aligned with the vehicle's roll, pitch, and yaw axes, as shown in Fig. 2.17. Each gimbal is driven by a servomotor for keeping the gyro platform fixed in inertial space eventhough the vehicle may have simultaneous rotations about all principal axes. The angle encoders mounted on the gimbal axes provide a feedback to the servos. Such a platform is said to be three-axis stabilized, much like a spacecraft (Chap. 7). The accelerometers and rate gyros mounted on the platform generate the angular rates and displacements by integration [3]. The angular displacements can also be obtained from the gimbal axes encoders by measuring the roll, pitch, and yaw inclinations of the vehicle relative to the platform.

For initially erecting the platform to a desired orientation, distant star constellations are employed as an inertial reference. Conversely, by measuring the angular deviation of the vehicle from the fixed platform, the vehicle's spatial coordinates at any given time relative to distant stars can be computed. This forms the basis of the typical *inertial navigation system* (INS) that is found in virtually every long-range flight vehicle. In order to work in a curved flight path around the Earth, the INS

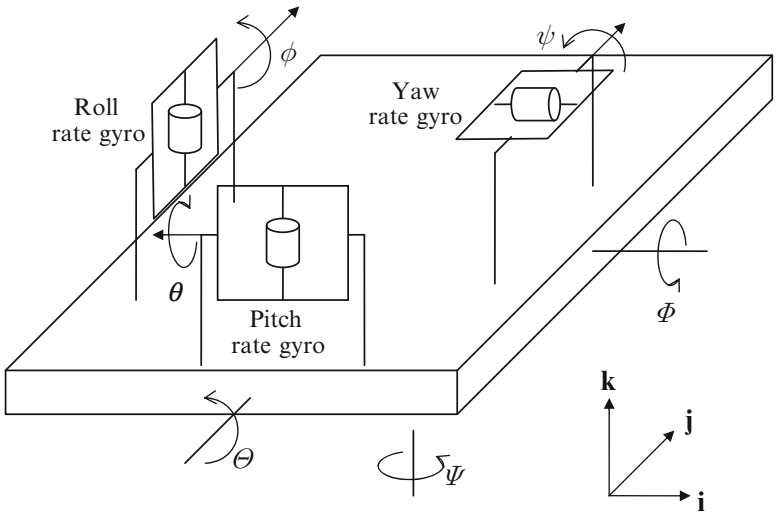


Fig. 2.18 A gyro stabilized platform with three rate gyros

must have a platform that is not exactly stationary, but continuously tilts itself as the vehicle moves, in order to maintain a nearly horizontal reference attitude. Thus, the gimbal servos driving the platform must track the rate, $\dot{\theta} = v/(R_0 + h)$, where R_0 is the planet's radius, v the speed, and h the constant altitude of the vehicle. Interestingly, the required angular acceleration of the platform can be obtained from the linear acceleration, a, sensed by an accelerometer according to $\ddot{\theta} = a/(R_0 + h)$, which is the equation of motion of a simple pendulum of length $R_0 + h$. Such a pendulum is called *Schuler's pendulum* and thus the required adjustment of the platform is known as Schuler's tuning.

Example 2.4. Consider a three-axis, gyro stabilized platform consisting of three identical rate gyros mounted about the roll, pitch, and yaw axes as shown in Fig. 2.18. The angular rate sensed by each gyro is fed to a servomotor, which applies a corrective, restoring torque to the platform. Let Φ, Θ, Ψ be the platform's roll, pitch, and yaw displacements, respectively, relative to an inertial frame. The roll rate gyro senses the relative angular rate, $\dot{\Phi}$, and undergoes an inertial gimbal displacement, ϕ, about the pitch axis according to the rate gyro transfer function,

$$\frac{\phi(s)}{s\Phi(s)} = -\frac{H}{Js^2 + cs + k} = G(s), \tag{2.158}$$

where H is the constant angular momentum of the rotor, and J, c, k the moment of inertia, viscous damping coefficient, and spring stiffness, respectively, about the gimbal axis. The net gimbal rotation relative to the platform is given by:

$$\alpha = \phi - \Theta, \tag{2.159}$$

which yields an electrical voltage output signal, $e_\alpha = K\alpha(s)$, via a transducer of gain K. Similarly, the gimbal rotations of the pitch and yaw rate gyros relative to the platform are

$$\begin{aligned} \beta &= \theta + \Phi \\ \sigma &= \psi - \Theta, \end{aligned} \tag{2.160}$$

respectively, with the identical transfer function given by:

$$\frac{\theta(s)}{s\Theta(s)} = \frac{\psi(s)}{s\Psi(s)} = G(s), \tag{2.161}$$

and the respective voltage pick-offs, $e_\beta = K\beta$ and $e_\sigma = K\sigma$. Since the restoring servomotor torques, τ_x, τ_y, τ_z, are directly proportional to the driving voltages, $e_\alpha, e_\beta, e_\sigma$, respectively, and are related by the same servomotor transfer function, $F(s)$, we write

$$\begin{aligned} \tau_x(s) &= F(s)e_\alpha(s) \\ \tau_y(s) &= F(s)e_\beta(s) \\ \tau_z(s) &= F(s)e_\sigma(s). \end{aligned} \tag{2.162}$$

On substituting (2.158)–(2.161), we have the following *transfer matrix* (Chap. 3) relating the control torques to the platform angular deviations:

$$\begin{Bmatrix} \tau_x(s) \\ \tau_y(s) \\ \tau_z(s) \end{Bmatrix} = \begin{bmatrix} sKF(s)G(s) & -KF(s) & 0 \\ KF(s) & sKF(s)G(s) & 0 \\ 0 & -KF(s) & sKF(s)G(s) \end{bmatrix} \begin{Bmatrix} \Phi(s) \\ \Theta(s) \\ \Psi(s) \end{Bmatrix}. \tag{2.163}$$

Such a coupled relationship between input and output variables is termed *multivariable control*. In Chap. 3, we shall consider design of such control systems.

While gyro stabilized platforms have been traditionally employed in most high-performance flight vehicles (airliners, satellite launch vehicles, cruise and ballistic missiles, and satellites), of late they have been replaced by IMU units that are directly strapped on to the vehicle's body frame. Such strapdown IMUs have the advantage of simplicity and ruggedness due to the presence of smaller moving parts compared to the gyro stabilized platform. They are also free from the errors inherent in gyros, such as drift, bias, misaligned rotors, and other nonlinearities. But the question naturally arises: how can a moving platform provide an inertial attitude reference? The simple answer is, it does not need to. The strapdown IMUs have gyros for measuring the angular orientation of the body-fixed axes on which the three accelerometers are mounted, thereby providing the attitude and acceleration of the vehicle relative to a reference state of the vehicle itself, rather than an inertial

reference frame. The attitude and acceleration data is then processed by a computer in order to generate the vehicle's orientation, position, and velocity in space. In other words, the mechanical INS reference provided by the gyro stabilized platform is replaced by a mathematical one. With the availability of the high-speed digital computers, it has become possible to compute the vehicle's attitude and navigational state in real time from the strapdown IMU data. Furthermore, the availability of the global positioning system (GPS) allows in-flight realignment of the strapdown accelerometers' axes, which was impossible with the gyro stabilized IMU.

The traditional gyros and mechanical accelerometers of the past have been replaced by modern solid-state electronic devices that have no moving parts for a better ruggedness and accuracy. An example is the *ring laser gyro* (RLG) that works on the principles of laser optics rather than Newton's laws. Similarly, the mechanical accelerometers have been replaced by piezoelectric sensors that can offer a better resolution and dynamic range of measurement.

2.9 Summary

Flight dynamics consists of motion of the center of mass (translational dynamics) and rotation of the vehicle about the center of mass (attitude dynamics). While translational flight dynamics is referred to an inertial frame, the attitude dynamics describes the rotation of a body-fixed frame relative to a nonrotating coordinate frame. Aircraft flight regards the local horizon frame to be inertial, whereas rocket and space flight is more commonly approximated by treating a planet-fixed frame as an inertial reference frame. The aerodynamic forces and moments depend upon the vehicle's attitude relative to the velocity vector, which is resolved in the wind axes system. The stability axes, having one axis along the equilibrium flight direction, are useful in carrying out the small perturbation approximation for aircraft dynamics plant. The dependence of aerodynamic forces and moments on the shape, size, speed, and orientation of the vehicle as well as the atmospheric properties is modeled by the angle-of-attack, the angle of sideslip, and nondimensional flow parameters – Mach number, Reynolds number, Knudsen number. Flight sensors are important as the eyes and ears of the automatic controller, and consist of motion sensors and flow sensors. A gyroscope is a typical motion sensor that can be employed in measuring either single- or multiaxis rotations. A single-axis rate gyro produces an electrical output proportional to the angular rate, while a rate-integrating gyro indicates the net angular motion from a reference point. A vertical gyro is a two degree-of-freedom sensor with roll and pitch capability. A fully gimbaled gyro platform is the classical heart of the inertial measurement unit (IMU) employed in inertial navigation systems (INS), while the strapdown IMU can be found in the modern INS and is normally coupled with the global positioning system (GPS) for a better accuracy and ruggedness. A generic flight dynamics plant is driven by both control and environmental inputs, and can be linearized about a nominal trajectory.

Exercises

2.1. Derive the kinematic equations of motion for the 3-2-1 Euler angle sequence, $(\psi)_3, (\theta)_2, (\phi)_1$, in terms of the body rates, (P, Q, R). Identify the singular points of this attitude representation.

2.2. A spacecraft is in an elliptical orbit around the Earth ($\mu = 398600.4$ km^3/s^2, mean radius, $R_0 = 6378.14$ km) with an eccentricity of 0.2 and a semimajor axis of 15,000 km. Determine the radius, speed, and flight path angle when the true anomaly is 60°.

2.3. A spacecraft has its perigee (minimum radius) point given by speed of 8 km/s and altitude 300 km above the Earth's surface. Calculate the position, velocity, and time since perigee when the true anomaly is 100°.

2.4. For the spacecraft of Exercise 2.3, what are the celestial position and velocity coordinates at the true anomaly of 100° if the orbital plane is given by $\Omega = 30°$, $i = 60°$, and $\omega = 200°$?

2.5. Using the coordinate transformation between the celestial and the local horizon frames, derive the relationship among orbital inclination, velocity azimuth, and latitude of a spacecraft.

2.6. An axisymmetric spacecraft with principal moments of inertia, (J_x, J_x, J_z) (products of inertia zeros), is initially in a nonrotating state. It has a small rotor with moment of inertia about the spin axis, J_r, spinning about the spacecraft's axis of symmetry at a rate, ω_r. Determine the final spacecraft angular velocity if: (a) the rotor axis is turned by a 45° angle about spacecraft's body axis, $\mathbf{k}$, and (b) the rotor is speeded-up to twice the original rate.

2.7. Derive the equations of motion relative to the body-fixed principal axes for the flight of a rocket in a vertical plane ($\psi = \text{const.}$). Assume a constant thrust, T, with mass variation given by

$$\dot{m} = -\frac{T}{v_\mathrm{e}},$$

where v_e is the constant exhaust speed of the propellant gases. The thrust vector's inclination in the principal body-fixed frame is given by the angles μ_1, μ_2, as shown in Fig. 2.10. The drag force is given by:

$$\mathcal{D} = \frac{1}{2}\rho v^2 S C_\mathrm{D},$$

where ρ is local atmospheric density, S a reference area, and C_D the drag-coefficient. The rocket must have a zero force normal to the flight path as well as a zero rolling moment. Use $(\psi)_3, (\theta)_1, (\phi)_3$ for the rocket's principal body-axis orientation relative to the local horizon frame. Qualitatively discuss how an automatic controller should maintain flight in the vertical plane despite lateral atmospheric disturbances due to planetary rotation and winds.

2.8. Derive the equations of motion relative to the body-fixed principal axes for the flight of an airplane in a horizontal plane ($\phi = 0$). Assume a constant engine thrust, T, the drag force is given by the same expression as in Exercise 2.7, and normal force and sideforce are given by:

$$Z = -\frac{1}{2}\rho v^2 S C_{\mathrm{L}}(\alpha),$$

$$Y = \frac{1}{2}\rho v^2 S C_Y(\beta),$$

where ρ is local atmospheric density, S a reference area, and C_{L}, C_Y are nondimensional coefficients. Use the Euler angles, $(\psi)_3, (\theta)_2, (\phi)_1$, for the airplane's principal body-axis orientation relative to the local horizon frame. Qualitatively discuss how an automatic controller should maintain flight in the horizontal plane despite lateral atmospheric disturbances due to planetary rotation and winds.

2.9. Can a rate gyro respond to either an impulsive or a step change in a vehicle's angular displacement? Answer this question using the final-value theorem of Laplace transform. Simulate the gimbal output response of the rate gyro of Example 2.3 to the following inputs:

(a) An impulsive angular displacement of the vehicle by 0.1 rad.
(b) A step change in the vehicle's angular orientation by 0.1 rad.

2.10. Consider attitude control of a thrust vectoring rocket (Exercise 2.7) in space, where the aerodynamic torque is negligible. The rocket is axisymmetric with $J_{yy} = J_{zz}$ and has an angular velocity perturbation, $\boldsymbol{\omega} = Q\mathbf{j} + R\mathbf{k}$, with respect to the equilibrium attitude. The control system is based upon two-degree-of-freedom gyro outputs, pitch rate, Q, and yaw rate, R, for driving the thrust gimbal angles, ϵ, μ, according to the following control laws:

$$\epsilon = -k_1 R; \qquad \mu = -k_2 Q,$$

where k_1, k_2 are positive constants called *feedback gains* and are related to the voltage amplifiers between the gyro outputs and the gimbal motor inputs.

(a) Derive the equations of rotational kinetics of the system.
(b) Analyze the stability of the control system.
(c) Derive an expression for the closed-loop response of the system when $k_1 = k_2 = k$, subject to the initial condition, $Q(0) = Q_0, R(0) = R_0$.

2.11. From the flight model of Example 2.1, derive the linearized plant for level flight of an aircraft relative to a spherical, rotating Earth with airspeed, $v(t)$, azimuth, $\psi(t)$, latitude, $\delta(t)$, and longitude, $\lambda(t)$, as the state variables. Control inputs are the electrical signals, $u_1(t)$ and $u_2(t)$ to the engine and aileron actuators, respectively. Wind speed, $v_{\mathrm{w}}(t)$, and wind azimuth, $\psi_{\mathrm{w}}(t)$, are stochastic (random) inputs, regarded as disturbances (process noise). Assume that a constant altitude is

maintained by a separate control system – not modeled here – called *altitude hold autopilot*, which ensures that the vertical component of the lift balances the weight at all times.

2.12. For the gyro stabilized platform of Example 2.4, derive the transfer matrix relating the servomotor torques and the platform's angular deviations if $H = 3 \times 10^{-3}$ N.m.s, $J = 10^{-5}$ kg m^2, $k = 0.1$ N.m/rad, $c = 1.5 \times 10^{-3}$ N.m.s/rad, $K = 100$ V/rad. The servomotor transfer function is given by:

$$F(s) = \frac{10^4}{s^2 + 200s + 10^4} \text{ N.m/V}.$$

Chapter 3
Control Design Techniques

3.1 Aims and Objectives

- To introduce linear control system concepts of transfer function, impulse response, step response, and frequency response, as well as the measures of control system performance and robustness
- To present an overview of design techniques for single-variable and multivariable linear control systems
- To highlight useful control theory concepts

3.2 Transfer Function and Singularity Inputs

A linear time-invariant (LTI), *single-input, single-output* (SISO) system's response to a specified input is most efficiently analyzed and designed with the concept of *transfer function*. Consider an LTI, SISO system, with input, $u(t)$, and output, $y(t)$, initially at rest and not experiencing the input initially, i.e., $u(t) = 0, y(t) = 0$, $(t < 0)$. All the time derivatives of the input and output are also zeros for $t < 0$. In the following discussion, the system must have the property of *causality*, i.e., it produces a nonzero output only when acted upon by a nonzero input for $t \geq 0$. The transfer function, $G(s)$, of such a system is then defined as the ratio of the output's Laplace transform, $Y(s)$, to that of the input, $U(s)$, subject to the zero initial condition,

$$G(s) \doteq \frac{Y(s)}{U(s)}. \tag{3.1}$$

The zero initial condition is commonly denoted by $u(0_-) = 0, y(0_-) = 0$, where $t = 0_-$ indicates the time immediately before the application of the input. By the definition of Laplace transform [11] and noting the causality of the system, we write

$$U(s) = \mathcal{L}\{u(t)\} = \int_0^\infty \mathrm{e}^{-st} u(t)\mathrm{d}t; \quad Y(s) = \mathcal{L}\{y(t)\} = \int_0^\infty \mathrm{e}^{-st} y(t)\mathrm{d}t. \tag{3.2}$$

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_3, © Springer Science+Business Media, LLC 2011

Table 3.1 Basic Laplace transforms and properties

$f(t)$	$F(s) = \mathcal{L}\{f(t)\}$
$\delta(t) = \begin{cases} \infty & (t = 0) \\ 0 & (t \neq 0) \end{cases}$	1
$u_s(t) = \begin{cases} 1 & (t \geq 0) \\ 0 & (t < 0) \end{cases}$	$\frac{1}{s}$
$tu_s(t)$	$\frac{1}{s^2}$
$t^n u_s(t);\ \ (n = 1, 2, \ldots)$	$\frac{n!}{s^{n+1}}$
$\mathrm{e}^{at} u_s(t)$	$\frac{1}{s-a}$
$\sin(\omega t) u_s(t)$	$\frac{\omega}{s^2+\omega^2}$
$\cos(\omega t) u_s(t)$	$\frac{s}{s^2+\omega^2}$
$\mathrm{e}^{at} h(t)$	$H(s-a);\ \ H(s) = \mathcal{L}\{h(t)\}$
$h(t-a)u_s(t-a)$	$\mathrm{e}^{-as} H(s);\ \ H(s) = \mathcal{L}\{h(t)\}$
$-th(t)$	$\frac{\mathrm{d}H(s)}{\mathrm{d}s};\ \ H(s) = \mathcal{L}\{h(t)\}$
$\frac{h(t)}{t}$	$\int_s^\infty H(p)\mathrm{d}p;\ \ H(s) = \mathcal{L}\{h(t)\}$
$\frac{\mathrm{d}h(t)}{\mathrm{d}t}$	$sH(s) - h(0_-);\ \ H(s) = \mathcal{L}\{h(t)\}$
$\int_0^t h(\tau)\mathrm{d}\tau$	$\frac{H(s)}{s};\ \ H(s) = \mathcal{L}\{h(t)\}$

The existence of the Laplace transforms requires that the integrals in (3.2) should converge to finite values for a given complex variable, s. If $U(s)$ and $Y(s)$ exist then they are unique. It can be shown [11] that a Laplace integral, $\mathcal{L}\{f(t)\}$, converges if and only if the function, $f(t)$, is piecewise continuous (i.e., any time interval, however large, can be broken up into a finite number of subintervals over each of which $f(t)$ is continuous, and at the ends of each subinterval, $f(t)$ is finite) and bounded by an exponential (i.e., there exists a positive, real constant, a, such that $|\ \mathrm{e}^{-at} f(t)\ |$ is finite at all times). Practical Laplace transforms of functions defined for $t \geq 0$ (along with some important properties) are listed in Table 3.1. Most of the commonly applied inputs are Laplace transformable, and the definition of the transfer function by (3.1) requires that the output of the LTI system to such an input is also Laplace transformable (which is true for all physical systems).

Example 3.1. Find the transfer function of a causal system described by the following governing differential equation:

$$\frac{\mathrm{d}^3 y}{\mathrm{d}t^3} - 4\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} + \frac{\mathrm{d}y}{\mathrm{d}t} - 2y = 0.1\frac{\mathrm{d}u}{\mathrm{d}t} - 0.5u,$$

where $u(t)$ is the input that begins acting at $t = 0$ and $y(t)$ the resulting output.

We apply the property of Laplace transform of a time derivative (Table 3.1) with zero initial conditions on both input and output, and write

$$\mathcal{L}\left\{\frac{\mathrm{d}y}{\mathrm{d}t}\right\} = sY(s); \quad \mathcal{L}\left\{\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\right\} = s^2Y(s); \quad \mathcal{L}\left\{\frac{\mathrm{d}^3y}{\mathrm{d}t^3}\right\} = s^3Y(s),$$

and

$$\mathcal{L}\left\{\frac{\mathrm{d}u}{\mathrm{d}t}\right\} = sU(s),$$

where $U(s) = \mathcal{L}\{u(t)\}$ and $Y(s) = \mathcal{L}\{y(t)\}$. Thus, taking the Laplace transform of both sides of the governing differential equation yields:

$$(s^3 - 4s^2 + s - 2)Y(s) = (0.1s - 0.5)U(s),$$

or

$$G(s) = \frac{Y(s)}{U(s)} = \frac{0.1s - 0.5}{s^3 - 4s^2 + s - 2}.$$

This example indicates that an LTI, SISO system's transfer function is a ratio of polynomials, called a *rational function*, in the Laplace variable, s. Note that here the degree of the numerator polynomial (i.e., the highest power of s) is less than that of the denominator polynomial.

For a general LTI, SISO system with the governing differential equation,

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} + a_n \frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}} + \cdots + a_2 \frac{\mathrm{d}y}{\mathrm{d}t} + a_1 y = b_{m+1} \frac{\mathrm{d}^m u}{\mathrm{d}t^m} + b_m \frac{\mathrm{d}^{m-1} u}{\mathrm{d}t^{m-1}} + \cdots + b_2 \frac{\mathrm{d}u}{\mathrm{d}t} + b_1 u, \tag{3.3}$$

the transfer function is the following:

$$G(s) = \frac{b_{m+1}s^m + b_m s^{m-1} + \cdots + b_2 s + b_1}{s^n + a_n s^{n-1} + \cdots + a_2 s + a_1}. \tag{3.4}$$

Here, n is the order of the system. If the degree of the numerator polynomial of $G(s)$ is either less than or equal to that of the denominator polynomial ($m \leq n$) then the LTI system is said to be *proper*. If $m < n$ then we have no direct connection between input and output, and the system is called *strictly proper*.

We can express the LTI, SISO system given by (3.3) in the following state-space form (Chap. 1):

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathsf{A}\mathbf{x}(t) + \mathsf{B}u(t) \\ y(t) &= \mathsf{C}\mathbf{x}(t) + Du(t), \end{aligned} \tag{3.5}$$

where $\mathsf{A}, \mathsf{B}, \mathsf{C}$ are constant coefficient matrices, D a constant scalar, and $\mathbf{x}(t)$ is the state vector. For a strictly proper system, $D = 0$. By taking Laplace transform of both sides of the state and output equations subject to zero initial condition, $\mathbf{x}(0_-) = \mathbf{0}$, we have

$$\begin{aligned} s\boldsymbol{\mathcal{X}}(s) &= \mathsf{A}\boldsymbol{\mathcal{X}}(s) + \mathsf{B}U(s) \\ Y(s) &= \mathsf{C}\boldsymbol{\mathcal{X}}(s) + \mathsf{D}U(s), \end{aligned} \tag{3.6}$$

or

$$Y(s) = \left[\mathsf{C}(s\mathsf{I} - \mathsf{A})^{-1} + D\right] U(s). \tag{3.7}$$

Comparing (3.1) and (3.7) we have the following relationship between the transfer function and the state-space coefficient matrices:

$$G(s) = \mathsf{C}(s\mathsf{I} - \mathsf{A})^{-1} + D. \tag{3.8}$$

As

$$(s\mathsf{I} - \mathsf{A})^{-1} = \frac{\text{adj}(s\mathsf{I} - \mathsf{A})}{|s\mathsf{I} - \mathsf{A}|} \tag{3.9}$$

we see that the denominator polynomial of $G(s)$ is the same as the characteristic polynomial, $|s\mathsf{I} - \mathsf{A}|$. Thus, the roots of the denominator polynomial, called the *poles* of the transfer function,

$$|s\mathsf{I} - \mathsf{A}| = s^n + a_n s^{n-1} + \cdots + a_2 s + a_1 = 0, \tag{3.10}$$

are the same as the eigenvalues of A, and determine the LTI system's characteristics. The roots of the numerator polynomial of the transfer function, $b_{m+1}s^m + b_m s^{m-1} + \cdots + b_2 s + b_1 = 0$ are called the *zeros* of the system. Clearly, the constant coefficients of the denominator polynomial, $a_1, \ldots, a_n$, have an important role to play in stability and performance of an LTI system, while the coefficients $b_1, \ldots, b_{m+1}$ also have an influence on the system's response to applied inputs.

Example 3.2. Find the poles and zeros of the system given in Example 3.1 and determine its stability.

The poles are the roots of the denominator polynomial of $G(s)$, $s^3 - 4s^2 + s - 2 = 0$, which are determined numerically by the use of MATLAB operator *roots.m* as follows:

```
>> a=[1 -4 1 -2];
>> roots(a)

ans =
   3.8751
   0.0624 + 0.7157i
   0.0624 - 0.7157i
```

Thus, the poles of the third-order system are $s_1 = 3.8751$ and $s_{2,3} = 0.0624 \pm 0.7157$i. By the stability criteria of Chap. 1, the system is unstable (presence of positive real parts). The system has only one zero, i.e., the root of the numerator polynomial of $G(s)$, $0.1s - 0.5 = 0$, or $s = 5$.

3.2.1 Impulse Response

The transfer function of an LTI, SISO system is also the Laplace transform of its output, $y(t)$, when the initial condition is zero and the input is a *unit impulse* (or Dirac delta) function applied at $t = 0$, $\delta(t)$, and given by:

$$\delta(t) = \begin{cases} \infty & (t = 0) \\ 0 & (t \neq 0) \end{cases}, \tag{3.11}$$

such that

$$\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1. \tag{3.12}$$

The unit impulse function, $\delta(t)$, can be visualized as a rectangular pulse of width, ϵ, and height, $1/\epsilon$, centered at $t = 0$, by taking the limit, $\epsilon \to 0$. The output of an LTI system with zero initial condition to a unit impulse input is called its *impulse response*, $g(t)$, and is given by:

$$g(t) = \mathcal{L}^{-1}\{G(s)\} \quad (t \geq 0). \tag{3.13}$$

A useful method of calculating the impulse response is by the *partial fraction expansion* of the transfer function as follows:

$$\begin{aligned} G(s) = &\frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \cdots + \frac{r_n}{s - p_n} \\ &+ \frac{r_{m1}}{s - p_m} + \frac{r_{m2}}{(s - p_m)^2} + \cdots + \frac{r_{m(k-1)}}{(s - p_m)^{k-1}} + \frac{r_{mk}}{(s - p_m)^k}, \end{aligned} \tag{3.14}$$

where $p_1, p_2, \ldots, p_n$ are n distinct poles, and p_m a pole of multiplicity k, of $G(s)$. The numerator coefficients of the series, $r_1, r_2, \ldots, r_n$, and $r_{m1}, \ldots, r_{mk}$, are called the *residues* of $G(s)$ and are determined by the following scheme:

$$\begin{aligned} r_1 &= (s - p_1)G(s)|_{s=p_1} \\ r_2 &= (s - p_2)G(s)|_{s=p_2} \\ r_n &= (s - p_n)G(s)|_{s=p_n} \\ r_{mk} &= (s - p_m)^k G(s)\big|_{s=p_m} \end{aligned} \tag{3.15}$$

$$r_{m(k-1)} = \left.\frac{\mathrm{d}(s-p_m)^k G(s)}{\mathrm{d}s}\right|_{s=p_m}$$

$$r_{m2} = \left.\frac{\mathrm{d}^{k-2}(s-p_m)^k G(s)}{\mathrm{d}s^{k-2}}\right|_{s=p_m}$$

$$r_{m1} = \left.\frac{\mathrm{d}^{k-1}(s-p_m)^k G(s)}{\mathrm{d}s^{k-1}}\right|_{s=p_m}$$

Example 3.3. Calculate the impulse response of a system with the following transfer function:

$$G(s) = \frac{s+3}{(s+1)^2(s+2)}.$$

The poles of the system are all real – a single pole $p_1 = -2$, and a double pole, $p_1 = -1$. Thus, we write

$$G(s) = \frac{r_1}{s+2} + \frac{r_2}{s+1} + \frac{r_3}{(s+1)^2},$$

where the residues are calculated as follows:

$$r_1 = (s+2)G(s)|_{s=-2} = \frac{-2+3}{(-2+1)^2} = 1$$

$$r_3 = (s+1)^2 G(s)\big|_{s=-1} = \frac{-1+3}{(-1+2)} = 2 \tag{3.16}$$

$$r_2 = \left.\frac{\mathrm{d}(s+1)^2 G(s)}{\mathrm{d}s}\right|_{s=-1} = \left.\frac{s+2-(s+3)}{(s+2)^2}\right|_{s=-1} = -1. \tag{3.17}$$

Thus, we have

$$G(s) = \frac{1}{s+2} - \frac{1}{s+1} + \frac{2}{(s+1)^2},$$

or, by Table 3.1,

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = \left(\mathrm{e}^{-2t} - \mathrm{e}^{-t} + 2t\mathrm{e}^{-t}\right) u_{\mathrm{s}}(t).$$

In case of complex poles (which always occur in conjugate pairs), the residues are also complex conjugates; hence, the partial fraction expansion involving a complex conjugate pair of poles can be combined to produce a quadratic (or second-order) subsystem.

Example 3.4. Calculate the impulse response of the system given in Example 3.2.

$$G(s) = \frac{0.1s - 0.5}{s^3 - 4s^2 + s - 2}.$$

The third-order system with distinct poles is expanded as follows:

$$G(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \frac{r_3}{s - p_3}.$$

Example 3.2 showed that the system had a real pole, $p_1 = 3.8751$, and a complex conjugate pair, $p_{2,3} = 0.0624 \pm 0.7157\mathrm{i}$. Thus, we expect the residue, r_1, to be real and the residues, $r_{2,3}$, to be complex conjugates. The complex terms are combined by writing $r_{2,3} = a \pm \mathrm{i}b$ and $p_{2,3} = c \pm \mathrm{i}d$, resulting in

$$\begin{aligned}\frac{r_2}{s - p_2} + \frac{r_3}{s - p_3} &= \frac{a + ib}{s - (c + id)} + \frac{a - ib}{s - (c - id)} \\ &= \frac{(a + ib)(s - c + id) + (a - ib)(s - c - id)}{(s - c)^2 + d^2} \\ &= \frac{2(as - bd - ac)}{(s - c)^2 + d^2}.\end{aligned}$$

Hence, we have

$$G(s) = \frac{r_1}{s - p_1} + \frac{2(as - bd - ac)}{(s - c)^2 + d^2},$$

and we first calculate r_1 by

$$r_1 = (s - p_1)G(s)|_{s=p_1} = \left.\frac{0.1s - 0.5}{(s - c)^2 + d^2}\right|_{s=p_1} = -0.00747479.$$

Next, r_1 is substituted into expanded $G(s)$, and the complex residue parts, a, b, calculated by comparing the coefficients of the numerator polynomial:

$$a = -\frac{r_1}{2} = 0.003737395,$$

and

$$b = -\frac{r_1 c + ac + ap_1 + 0.05}{d} = -0.089772745$$

Finally, we take the inverse Laplace transform of the expanded transfer function according to Table 3.1, yielding

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = \left[r_1 \mathrm{e}^{p_1 t} + 2\mathrm{e}^{ct}\left(\frac{1}{a}\cos(dt) - \left\{b + \frac{ac}{d}\right\}\sin(dt)\right)\right] u_\mathrm{s}(t),$$

which, by substituting the values computed earlier, becomes

$$\begin{aligned}g(t) = \big[&- 0.0075\mathrm{e}^{3.8751t} + 2\mathrm{e}^{0.0624t}\,(267.566\cos(0.7157t) \\ &+ 0.0894467\sin(0.7157t))\big] u_\mathrm{s}(t).\end{aligned}$$

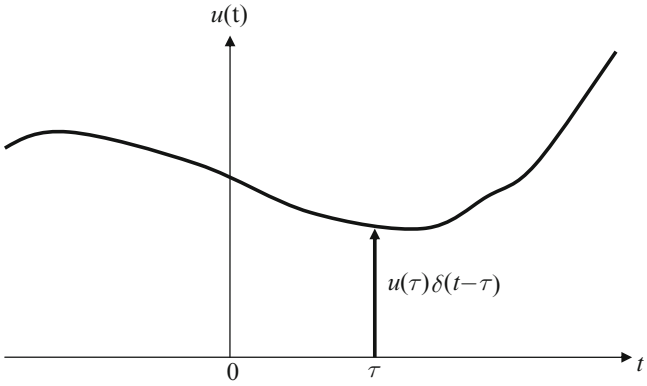


Fig. 3.1 The functional value, $u(\tau)$, represented by an impulse at $t = \tau$

The impulse response of a linear system (thus the transfer function) is a useful mathematical construct, as it helps us in deriving the system's response to an arbitrary input by linear superposition (Chap. 1). Consider a piecewise continuous, arbitrary input, $u(t)$, represented by series of impulse inputs, $\delta(t - \tau)$, applied at various times, $-\infty < \tau \leq t$, and scaled by the current input magnitude, $u(\tau)$, as shown in Fig. 3.1. By the *mean value theorem* of integral calculus [11], (3.12) yields the following expression for input magnitude at time t, called the *sampling property* of the Dirac delta function:

$$u(t) = \int_{-\infty}^{\infty} u(\tau)\delta(t - \tau)\mathrm{d}\tau. \tag{3.18}$$

From the definition of the unit impulse function, it is clear that the limits of integration in (3.18) need not be infinite, but should only bracket the time instant, τ, at which the impulse is applied.

If the system is initially at rest (i.e., the output and all its time derivatives are zero at $t = 0_-$) then the system's response at a subsequent time, t, is simply given by the summation of the individual impulse responses, scaled by the current input magnitude, $u(\tau)$:

$$y(t) = \int_{-\infty}^{\infty} u(\tau)g(t - \tau)\mathrm{d}\tau. \tag{3.19}$$

The integral on the right-hand side of (3.19) is called the *convolution integral*, which is expressed as $(u * g)(t)$, and is symmetric in $u(.)$ and $g(.)$ such that we can write

$$y(t) = (u * g)(t) = \int_{-\infty}^{\infty} u(t - \tau)g(\tau)\mathrm{d}\tau. \tag{3.20}$$

In the common control application, the input, $u(t)$, begins acting at $t = 0$, and since $g(t-\tau) = 0$ for $t < \tau$, we have

$$y(t) = (u * g)(t) = \int_0^t u(\tau)g(t-\tau)\mathrm{d}\tau = \int_0^t u(t-\tau)g(\tau)\mathrm{d}\tau. \tag{3.21}$$

Note that (3.21) can be alternatively derived by taking the inverse Laplace transform of $Y(s) = G(s)U(s)$ and using (3.13). Numerical evaluation of (3.21) is easily carried out by quadrature [11] for an arbitrary, piecewise continuous input, $u(t)$. (The concept of convolution can be extended beyond impulse response. Any two functions that satisfy the convolution property given by (3.21) are said to *convolve* with each other, and the Laplace transform of the convolution integral is a product of the Laplace transforms of the two functions.)

Example 3.5. Consider the single-axis rotation of a rigid spacecraft with the following equation of motion:

$$J\ddot{\theta} = u, \tag{3.22}$$

where $\theta(t)$ is the angle of rotation, J the constant moment of inertia, and $u(t)$ is the applied torque input. The spacecraft is equipped with a pair of rocket thrusters that can apply a torque impulse of magnitude, $\pm u_0$, whenever required.[1] It is desired to rotate the spacecraft by angle, θ_f, beginning from rest at $t = 0$. Find the final time, t_f, for achieving the desired rotation by application of only two impulses.

Let us apply the first impulse at $t = 0$ to initiate the rotation in the desired direction, and the second impulse in the opposite direction at $t = t_\mathrm{f}$ in order to stop the rotation when the desired angle is reached. Thus, the input profile is given by:

$$u(t) = u_0\left[\delta(t) - \delta(t-t_\mathrm{f})\right]. \tag{3.23}$$

Substitution of (3.23) into (3.22) and integration in time yields

$$\dot{\theta}(t) = \frac{1}{J}\int_0^t u(\tau)\mathrm{d}\tau = \begin{cases} \frac{u_0}{J} & (0 \le t < t_\mathrm{f}) \\ 0 & (t \ge t_\mathrm{f}) \end{cases} \tag{3.24}$$

$$\theta(t) = \begin{cases} \frac{u_0}{J}t & (0 \le t < t_\mathrm{f}) \\ \frac{u_0}{J}t_\mathrm{f} & (t \ge t_\mathrm{f}) \end{cases}. \tag{3.25}$$

By equating θ_f with $u_0 t_\mathrm{f}/J$, we have $t_\mathrm{f} = J\theta_f/u_0$. The larger the magnitude of the impulse, u_0, the smaller would be the required time for achieving the desired angle. Such an open-loop control by the application of equal and opposite impulses separated by a given interval is called *bang–bang* control, as depicted in Fig. 3.2.

[1] Note that a torque impulse of magnitude u_0 translates into an input profile (torque vs. time) of area u_0 N m s applied instantaneously, i.e., in zero time.

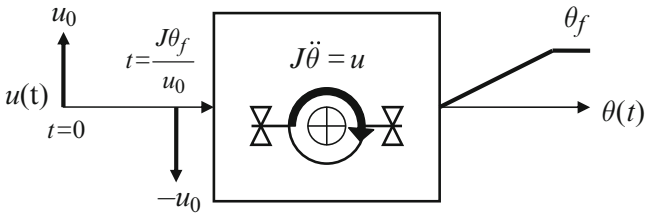


Fig. 3.2 Bang–bang rotation of a rigid spacecraft

If the impulses are replaced by pulses of the same area, u_0, the required time of rotation would be larger than $t_f = J\theta_f/u_0$, as seen later. In fact, of all the possible input profiles of a given amplitude, the smallest time of rotation is achieved by a bang–bang profile. This forms the basis of *time-optimal control*.

3.2.2 Step Response

Related to the transfer function are responses to other singularity inputs, such as the *unit step* and *unit ramp* functions. It is clear from the integral property of Laplace transform (last row of Table 3.1) that the unit step function, u_s, is the time integral of the unit impulse function, $\delta(t)$,

$$u_s(t) = \mathcal{L}^{-1}\left\{\frac{1}{s}\right\} = \int_0^t \delta(\tau)\mathrm{d}\tau \quad (t > 0). \tag{3.26}$$

Similarly, the *step response*, $y_s(t)$, of an LTI system, $G(s)$, defined as the output when the input is a unit step function applied at $t = 0$ with a zero initial condition, is the time integral of the impulse response:

$$y_s(t) = \mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\} = \int_0^t g(\tau)\mathrm{d}\tau \quad (t > 0). \tag{3.27}$$

The step response (also called indicial response) is valuable in studying a stable LTI system's *performance* when a sudden change is desired in the output, and can be derived in a closed-form by partial fraction expansion. The time integral of the step response is the *ramp response*, which is useful in such applications as tracking an object moving with a constant velocity.

Example 3.6. For the single-axis rotation of a rigid spacecraft (Example 3.5), consider the application of two equal and opposite torque pulses of area u_0 N m s (width, a, and height, u_0/a) as shown in Fig. 3.3. The torque profile is represented as follows:

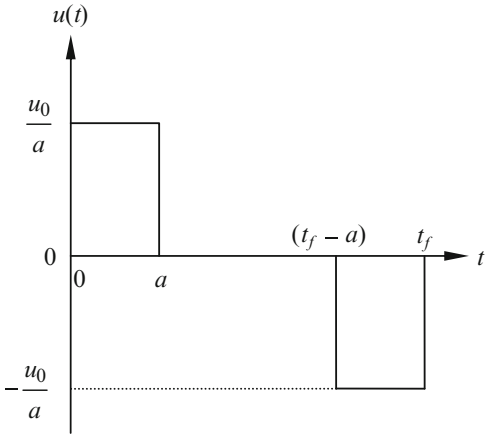


Fig. 3.3 Torque pulses of width a and height u_0/a for rigid spacecraft rotation

$$u(t) = \frac{u_0}{a}\left[u_s(t) - u_s(t-a) - u_s(t-a-t_f) + u_s(t-t_f)\right]. \tag{3.28}$$

Substitution of (3.28) into (3.22) and integration in time yields

$$\dot{\theta}(t) = \frac{1}{J}\int_0^t u(\tau)\mathrm{d}\tau = \begin{cases} \dfrac{u_0 t}{aJ} & (0 \le t \le a) \\ \dfrac{u_0}{J} & (a < t \le t_f - a) \\ \dfrac{u_0}{J}(t_f - t) & (t_f - a < t \le t_f) \\ 0 & (t > t_f) \end{cases} \tag{3.29}$$

$$\theta(t) = \begin{cases} \dfrac{u_0 t^2}{2Ja} & (0 \le t \le a) \\ \dfrac{u_0}{J}\left(t - \dfrac{3}{2}a\right) & (a < t \le t_f - a) \\ \dfrac{u_0}{Ja}\left[a\,(t_f - 2a) - \dfrac{1}{2}(t_f - t)^2\right] & (t_f - a < t \le t_f) \\ \dfrac{u_0}{J}\,(t_f - 2a) & (t > t_f) \end{cases}. \tag{3.30}$$

Therefore, the final rotation produced by the torque pulses is $\theta_f = u_0(t_f - 2a)/J$. Thus, for a given rotation angle, θ_f, the required time is

$$t_f = 2a + \frac{J}{u_0}\theta_f,$$

which is larger than the time required by a bang–bang impulse profile, $J\theta_f/u_0$, in Example 3.5.

Example 3.7. Many SISO systems are of second order, governed by the following differential equation:

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = u(t), \tag{3.31}$$

where the constants *m,c,k* are referred to as *inertia*, *damping*, and *stiffness* parameters, respectively. The transfer function of the second-order system is

$$\frac{Y(s)}{U(s)} = \frac{1}{ms^2 + cs + k}, \tag{3.32}$$

or in the traditional form,

$$\frac{Y(s)}{U(s)} = \frac{1}{m(s^2 + 2\zeta\omega_n s + \omega_n^2)}, \tag{3.33}$$

where $\omega_n \doteq \sqrt{\frac{k}{m}}$ is called the *natural frequency* and $\zeta \doteq \frac{c}{2m\omega_n}$ is the *damping-ratio*. It is clear that the stability of the second-order system depends upon the roots of the characteristic equation $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$, which are written as

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}. \tag{3.34}$$

Applying the linear stability criteria of Chapter 1, it is evident that the system is stable only for $\zeta \geq 0$. Furthermore, if $0 \leq \zeta < 1$, the eigenvalues are complex conjugates,

$$s_{1,2} = -\zeta\omega_n \pm \mathrm{i}\omega_n\sqrt{1 - \zeta^2}, \tag{3.35}$$

and the unforced system's initial response displays an oscillatory behavior with time, while $\zeta \geq 1$ is the case of real, negative eigenvalues, representing a purely exponential behavior (similar to a first-order LTI system). Most second-order control systems are designed with $0 \leq \zeta < 1$, because the overdamped case of $\zeta \geq 1$ generally requires large control input magnitudes.

The performance of a stable second-order control system is analyzed by its step response, derived as follows.

$\underline{0 \leq \zeta < 1}$:

$$Y(s) = \frac{(1/m)U(s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1/m}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}, \tag{3.36}$$

which is expanded by partial fractions into real and complex conjugate poles,

$$Y(s) = \frac{r_1}{s} + \frac{as + b}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{3.37}$$

where

$$r_1 = sY(s)|_{s=0} = \frac{1}{m\omega_n^2} = \frac{1}{k}.$$

By comparing the numerator polynomials of (3.36) and (3.37), we have

$$a = -\frac{1}{k}; \qquad b = -\frac{2\zeta\omega_n}{k}.$$

Thus, we write

$$Y(s) = \frac{1}{k}\left[\frac{1}{s} - \frac{s + 2\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_n^2(1 - \zeta^2)}\right], \tag{3.38}$$

or, taking the inverse Laplace transform (Table 3.1):

$$y_s(t) = \frac{1}{k}\left[1 - \mathrm{e}^{-\zeta\omega_n t}\left\{\cos(\omega_\mathrm{d} t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_\mathrm{d} t)\right\}\right] u_\mathrm{s}(t), \tag{3.39}$$

where $\omega_\mathrm{d} \doteq \omega_n\sqrt{1-\zeta^2}$ is the *damped natural frequency* of the system.

$\underline{\zeta = 1}$:

$$Y(s) = \frac{1}{ks} - \frac{1}{m\omega_n}\left[\frac{1}{\omega_n(s + \omega_n)} + \frac{1}{(s + \omega_n)^2}\right], \tag{3.40}$$

or, taking the inverse Laplace transform (Table 3.1):

$$y_s(t) = \left[\frac{1}{k}\left(1 - \mathrm{e}^{-\omega_n t}\right) - \frac{t}{\sqrt{km}}\mathrm{e}^{-\omega_n t}\right] u_\mathrm{s}(t). \tag{3.41}$$

$\underline{\zeta > 1}$:

$$Y(s) = \frac{1}{ks} + \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2}, \tag{3.42}$$

which, by taking the inverse Laplace transform (Table 3.1), becomes the step response,

$$y_s(t) = \frac{1}{k}\left(1 + r_1\mathrm{e}^{p_1 t} + r_2\mathrm{e}^{p_2 t}\right) u_\mathrm{s}(t), \tag{3.43}$$

where

$$p_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1},$$

and

$$r_{1,2} = \frac{1}{2\sqrt{\zeta^2 - 1}\left(\mp\zeta + \sqrt{\zeta^2 - 1}\right)}.$$

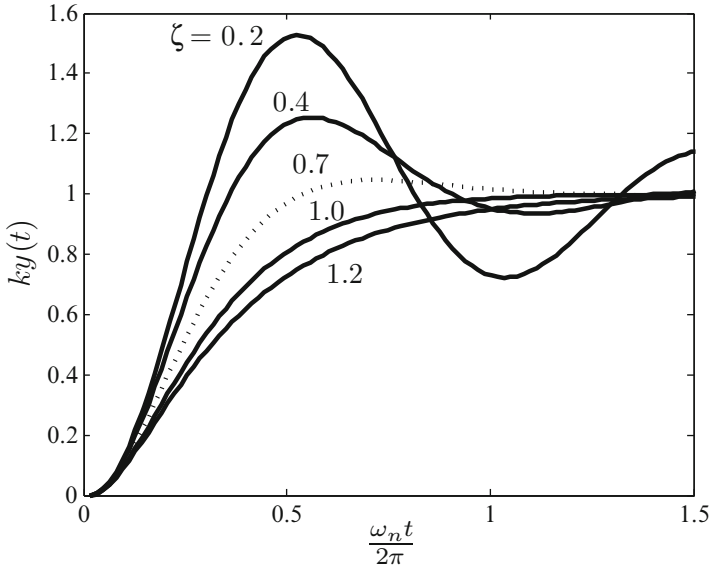


Fig. 3.4 Step response of a second-order, SISO, LTI system

Figure 3.4 shows a plot of $k\,y_s(t)$ against the nondimensional time, $\omega_n t/(2\pi)$, for different values of ζ in the stable range. As ζ is increased, the system takes a longer time to reach the desired value of unity for the first time. This time is called the *rise time* and indicates the speed (or alacrity) of the system's response to a step input. On the other hand, the time taken by the system to achieve an equilibrium (or steady) state after being disturbed by a step input is indicated by the *settling-time*, which is defined as the time taken by the amplitude of the step response to settle within $\pm 2\%$ of the steady-state value. The settling-time and rise time are nearly the same in the nonoscillatory response of system with $\zeta \geq 1$, which is a behavior identical to a first-order system. On the other hand, for $\zeta < 1$, the system does not stop at the steady state value when it reaches it, but instead crosses it. Such a crossing of the steady state is called an *overshoot*. Repeated crossings of the steady-state value results in an oscillatory response, which decays in amplitude for $\zeta > 0$. The maximum overshoot becomes larger and larger as the damping ratio, ζ, is reduced as seen in Fig. 3.4, causing an increase in the settling-time.

The following important approximation for the settling-time, t_s, of step response of a stable second-order system can be obtained by noting that the system essentially behaves like a first-order system[2] in the neighborhood of the steady state (Fig. 3.3):

[2] An asymptotically stable first-order system with transfer function,

$$\frac{Y(s)}{U(s)} = \frac{y_0}{s+k}\,; \qquad (k > 0),$$

has settling-time $t_s = \frac{4}{k}$.

$$e^{-\zeta\omega_n t_s} \simeq 0.02, \tag{3.44}$$

or,

$$t_s \simeq -\frac{\ln(0.02)}{\zeta\omega_n} \simeq \frac{4}{\zeta\omega_n}. \tag{3.45}$$

The rise time, maximum overshoot, and settling-time are important performance parameters of a system. It is evident from Fig. 3.4 that the increase of speed (rise time) and decrease of maximum overshoot are contradictory design requirements. A trade-off between the two can be obtained by selecting a value of ζ which is neither too large (causing a sluggish response) nor too small (leading to a large overshoot). In most design applications, $\zeta = \frac{1}{\sqrt{2}} = 0.707$ (shown as dotted line in Fig. 3.4) is considered ideal in striking a compromise between the speed and the maximum overshoot.

The concept of first- and second-order system analysis can be extended to higher-order systems, which have one or two poles (eigenvalues) nearest the imaginary axis and the other poles further away from it. Because the system's transient response is primarily governed by the poles that have the smallest real part magnitude, such poles are called the *dominant poles* of the system. Most higher-order physical systems can be approximated by a dominant second-order system (i.e., a pair of complex conjugate poles), called the primary (or predominant) *mode*. The other poles occurring either singly or in pairs are treated as less dominant (or secondary) modes that have a smaller contribution to the response, in proportion to the magnitudes of their respective real parts. Such an approximation is commonly applied to linear dynamical systems and is called *modal analysis*. For this reason, the second-order system is a valuable analysis tool, and terms such as damping-ratio and natural frequency can be used to denote either primary or secondary modes represented by each pair of complex conjugate poles.

3.2.3 Frequency Response

Apart from the singularity functions, some smooth test functions can be applied as inputs for analyzing an unknown physical system. The response of an LTI system to a simple harmonic input at a particular frequency, subject to zero initial condition, is called *frequency response*, and reveals important control system properties such as the natural frequencies, and the behavior in the presence of high-frequency, unmodeled disturbances, called *noise*. Consider the applied harmonic input,

$$u(t) = u_0 \cos(\omega t) u_s(t), \tag{3.46}$$

where ω is called the frequency of excitation. As the LTI system is initially at rest, and the input begins acting at $t = 0$, the system's output at a large time, $t \geq T > 0$, is also harmonic and is given by the frequency response,

$$y_f(t) = y_0 \cos\left(\omega\{t - T\}\right) u_s(t - T), \tag{3.47}$$

provided the system is asymptotically stable, i.e., the transient response due to the step input, $u_0 u_s(t)$, decays to zero in the limit $t \to \infty$, or practically vanishes at $t = T$. This implies shifting the zero initial condition from $t = 0$ to $t = T$, and taking the Laplace transform of the frequency response with the time-shift property (ninth row, Table 3.1):

$$Y_f(s) = \mathcal{L}\{y_f(t)\} = y_0 \mathrm{e}^{Ts} \mathcal{L}\{\cos(\omega t) u_s(t)\}. \tag{3.48}$$

As the Laplace transform of the harmonic input is given by $U(s) = u_0 \mathcal{L}\{\cos(\omega t) u_s(t)\}$, we write

$$G(s) = \frac{Y_f(s)}{U(s)} = \frac{y_0 \mathrm{e}^{Ts}}{u_0}, \tag{3.49}$$

or, in the harmonic limit, $s = \mathrm{i}\omega$, we have

$$G(\mathrm{i}\omega) = \frac{Y_f(\mathrm{i}\omega)}{U(\mathrm{i}\omega)} = \frac{y_0 \mathrm{e}^{\mathrm{i}\omega T}}{u_0}. \tag{3.50}$$

Here, we introduce the *gain* of the frequency response, y_0/u_0, as the magnitude of $G(\mathrm{i}\omega)$,

$$\frac{y_0}{u_0} = \mid G(\mathrm{i}\omega) \mid, \tag{3.51}$$

and the *phase angle*, $\phi(\omega) = \omega T$, as the phase of $G(\mathrm{i}\omega)$:

$$\phi = \omega T = \tan^{-1} \frac{\Im\{G(\mathrm{i}\omega)\}}{\Re\{G(\mathrm{i}\omega)\}}. \tag{3.52}$$

Thus, the frequency response of an LTI, SISO system is given by the transfer function on the imaginary axis of the Laplace domain, $s = \mathrm{i}\omega$. In phasor representation, the gain and phase of the frequency response are related to the harmonic transfer function by

$$G(\mathrm{i}\omega) = \mid G(\mathrm{i}\omega) \mid \mathrm{e}^{\mathrm{i}\phi(\omega)}. \tag{3.53}$$

Example 3.8. Consider the response of the spring, mass, and damper system of Example 3.7 with $0 \leq \zeta < 1$ to the simple harmonic input, (3.46). From Table 3.1, for the zero initial condition, we have

$$Y(s) = \frac{(1/m)U(s)}{(s + \zeta\omega_n)^2 + \omega_\mathrm{d}^2} = \frac{(u_0/m)s}{[(s + \zeta\omega_n)^2 + \omega_\mathrm{d}^2](s^2 + \omega^2)}, \tag{3.54}$$

where, $\omega_\mathrm{d} = \omega_n \sqrt{1 - \zeta^2}$. Expansion of the response into two quadratic denominator factors yields the following expression:

$$Y(s) = \frac{u_0}{m} \left[\frac{a/s + b}{(s + \zeta\omega_n)^2 + \omega_\mathrm{d}^2} + \frac{d/s + e}{s^2 + \omega^2} \right]. \tag{3.55}$$

Comparing the coefficients of the numerator polynomial, we have

$$\begin{aligned} a &= -d\alpha^2 \\ b &= -e = \frac{2\omega_n\zeta}{\omega^2(\alpha^2-1)}d \\ d &= \frac{1-\alpha^2}{(1-\alpha^2)^2+4\alpha^2\zeta^2}, \end{aligned}$$

where

$$\alpha = \omega_n/\omega.$$

By taking the inverse Laplace transform of (3.55) we obtain the following expression for the output in the time domain:

$$y(t) = \frac{u_0}{m}\left[\frac{a}{\omega_n^2} - \mathrm{e}^{-\zeta\omega_n t}\left\{\frac{a}{\omega_n^2}\cos(\omega_\mathrm{d}t) + \left(\frac{a\zeta - b\omega_n}{\omega_n\omega_\mathrm{d}}\right)\sin(\omega_\mathrm{d}t)\right\}\right.$$
$$\left. + \frac{c}{\omega^2}\{1-\cos(\omega t)\} - \frac{b}{\omega}\sin(\omega t)\right]u_\mathrm{s}(t). \tag{3.56}$$

Note that the term in the first row on the right-hand side of (3.56) decays to au_0/k in the limit $t \to \infty$ (or practically for $t > T$), thereby leaving the term in the second row on the right-hand side as the only function of time. Furthermore, the zero initial condition is easily seen to be satisfied,

$$y(0) = \frac{u_0}{m}\left(\frac{a}{\omega_n^2} + \frac{c}{\omega^2}\right) = 0. \tag{3.57}$$

Hence, the frequency response of the system, $y_f(t)$, expressed in the time domain is the following:

$$y_f(t) = -\frac{u_0}{m}\left[\frac{c}{\omega^2}\cos(\omega t) + \frac{b}{\omega}\sin(\omega t)\right]u_\mathrm{s}(t-T). \tag{3.58}$$

Comparison with (3.47),

$$y_f(t) = y_0[\cos(\omega t)\cos(\omega T) + \sin(\omega t)\sin(\omega T)]u_\mathrm{s}(t-T), \tag{3.59}$$

reveals the phase,

$$\tan\phi = \tan(\omega T) = \frac{b\omega}{d} = \frac{2\zeta\omega_n}{\omega_n^2 - \omega^2}, \tag{3.60}$$

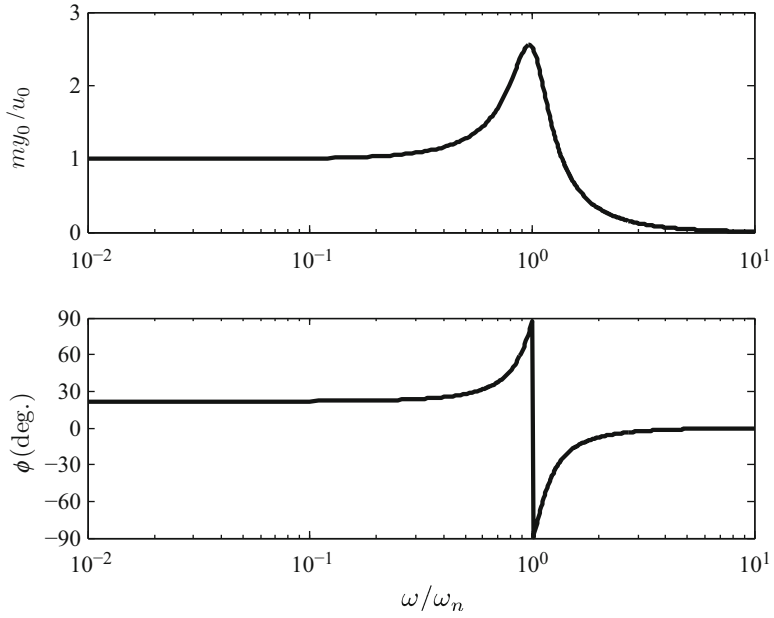


Fig. 3.5 Frequency response of a spring, mass, and damper system for $\zeta = 0.2$

and the gain,

$$\frac{y_0}{u_0} = \frac{1}{m\omega^2}\sqrt{d^2 + \left(\frac{2d\zeta\omega\omega_n}{\omega_n^2 - \omega^2}\right)^2}. \tag{3.61}$$

Clearly, the gain, $|G(\mathrm{i}\omega)|$, attains a maximum value at $\omega = \omega_n$, for a given damping ratio, $0 \leq \zeta < 1$. This is the well known condition of *resonance* where the excitation at the natural frequency builds up the response to a maximum amplitude by continually pumping-in energy. Furthermore, it is evident from (3.60) that the phase angle, $\phi(\omega)$, changes by $\pm 180°$ at $\omega = \omega_n$. While resonance is a second-order behavior, it can be used to isolate the modes of a higher-order system (provided they are not clustered together). Thus, by exciting a stable, higher-order system at various frequencies and studying its gain and phase, we can identify its distinct natural frequencies. Such an approach of system identification is called *spectrum analysis*. Figure 3.5 shows a plot of my_0/u_0 and the phase of the frequency response for $\zeta = 0.2$ in the frequency range, $0.01 \leq (\omega/\omega_n) \leq 10$. Note the peak corresponding to the natural frequency, the rapid decay of gain at higher frequencies, and the change of phase by $-180°$ at the natural frequency.

Typically, the gain of the frequency response is expressed on a logarithmic *decibel* scale:

$$\text{Gain } = 20\log_{10}|G(\mathrm{i}\omega)| \text{ (dB)}. \tag{3.62}$$

A plot of gain (dB) and phase (°) against a logarithmic frequency scale is termed a *Bode plot*. Bode plots are commonly used for studying stability, performance, and robustness properties of control systems. Each second-order subsystem corresponding to a quadratic factor in the characteristic polynomial generally appears as a peak in the gain plot and a phase change by $\pm 180°$ (unless it is affected by the numerator polynomial). Thus, the associated natural frequencies of the respective modes can be easily identified in the Bode plot. The gain at zero frequency, $|\ G(0)\ |$, is called *DC gain* of the system, and is useful in calculating the steady state response to a step input. The range of frequencies ($0 \leq \omega \leq \omega_b$) over which the gain stays above 0.707 $|G(0)|$ is called the system's *bandwidth* and is denoted by the frequency, ω_b. The bandwidth indicates the highest frequency signal to which the system has an appreciable response. For higher frequencies ($\omega > \omega_b$), the gain of a strictly proper system generally declines rapidly with increasing frequency, called *roll-off*. As noise is typically a high-frequency signal, a system with a large roll-off has a reduced sensitivity to noise – a desirable property for a robust system.

The roll-off and phase of a linear system in the high-frequency limit, $\omega \to \infty$, tell us about the difference in the degree of numerator and denominator polynomials of the transfer function. For example, the roll-off is approximately $-20(n - m)$ dB/decade, where n is the degree of the denominator (characteristic) polynomial and m is that of the numerator polynomial of the transfer function, $G(s)$.

Example 3.9. Consider the longitudinal dynamics of an aircraft described by the following transfer function:

$$G(s) = -\frac{(s + 0.01)(s + 0.5)}{(s^2 + 0.001s + 0.005)(s^2 + 1.5s + 1)}. \tag{3.63}$$

The computation of the numerator and denominator polynomials, followed by the poles and their corresponding natural frequencies and damping-ratios, is carried out using basic MATLAB commands:

```
>> n1=[-1 -.01];n2=[1 .5];num=conv(n1,n2) % numerator polynomial

num =   -1.0000   -0.5100   -0.0050

>> d1=[1 .001 .005];d2=[1 1.5 1];den=conv(d1,d2) % denominator polynomial

den =    1.0000    1.5010    1.0065    0.0085    0.0050

>> pole=roots(den) % poles of the system
pole =
  -0.7500 + 0.6614i
  -0.7500 - 0.6614i
  -0.0005 + 0.0707i
  -0.0005 - 0.0707i

>> wn=abs(pole), zeta=-real(pole)./wn % natural freq., damping-ratio
wn =
  1.0000e+000
  1.0000e+000
  7.0711e-002
  7.0711e-002
```

```
zeta =
  7.5000e-001
  7.5000e-001
  7.0711e-003
  7.0711e-003
```

Hence, the system has two modes, each corresponding to a pair of complex conjugate poles. The lower frequency mode at $\omega_n = 0.707$ rad/s is lightly damped ($\zeta = 0.007$), while the higher frequency mode at $\omega_n = 1$ rad/s is well damped ($\zeta = 0.75$). The Bode plot of the plant in the frequency range $10^{-4} \leq \omega \leq 100$ rad/s is generated using the following basic MATLAB commands:

```
>> w=logspace(-4,2);
>> N=num(:,1)*(i*w).^2+num(:,2)*(i*w)+num(:,3)*ones(size(w,1));
>> D=den(:,1)*(i*w).^4+den(:,2)*(i*w).^3+den(:,3)*(i*w).^2 ...
    +den(:,4)*(i*w)+den(:,5)*ones(size(w,1));
>> G=N./D;
>> gain=20*log10(abs(G));
>> phase=angle(G)*180/pi+180;
>> subplot(212),semilogx(w,phase),grid,xlabel('Frequency (rad/s)'),
   ylabel('Phase (deg.)'),
   subplot(211),semilogx(w,gain),grid,ylabel('Gain (dB)')
```

Alternatively, the Bode plot can be created more easily using the following MATLAB-Control Systems Toolbox (CST) commands:

```
>> sys=tf(num,den)

Transfer function:
            -s^2 - 0.51 s - 0.005
-----------------------------------------------
s^4 + 1.501 s^3 + 1.007 s^2 + 0.0085 s + 0.005
>> bode(sys)
```

The resulting Bode plot is shown in Fig. 3.6. Note the 180° phase change corresponding to the mode of natural frequency 0.0707 rad/s, which we will call the first mode. A sharp peak in the gain plot is also evident at $\omega = 0.0707$ rad/s signifying the dominance of the first mode. The second mode at $\omega = 1$ rad/s is well damped and the corresponding pole is close to the system's zeros, hence the corresponding peak and 180° phase change are absent. This indicates that the second mode does not contribute significantly to the system's frequency response for the concerned output variable. The DC gain is 0 dB and the bandwidth is seen to be approximately 1.05 rad/s. A roll-off of −40 dB per decade is evident in Fig. 3.6, indicating low noise sensitivity (high robustness) at high frequencies. As expected, the phase tends to −180° in the limit $\omega \to \infty$.

The step response of the system,

```
>> t=0:0.1:100;
>> step(sys,t)
```

plotted in Fig. 3.7, confirms the dominance of the first mode in the system's response with a characteristic time period of approximately $2\pi/(.0707) = 89$ s and the light damping causing an amplitude decay of about 5% per cycle. This indicates the fact that the given transfer function can be approximated by only the first mode.

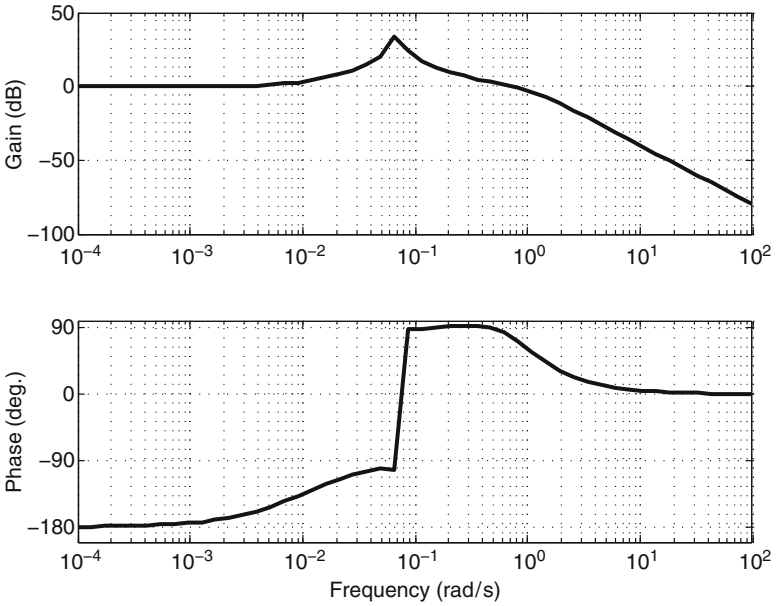


Fig. 3.6 Frequency response of aircraft longitudinal dynamics

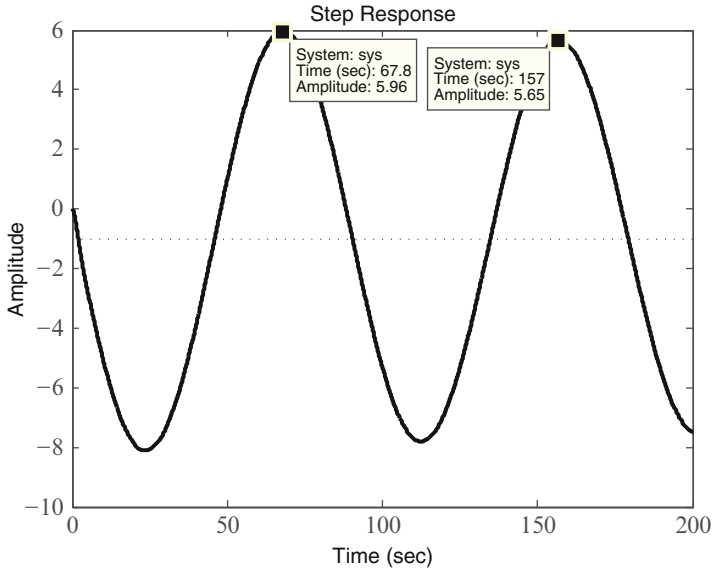


Fig. 3.7 Step response of aircraft longitudinal dynamics

3.3 Single Variable Design

The basic closed-loop system consists of a plant in a feedback loop with a controller, as shown in Fig. 3.8. As we are interested in controlling single-input, single-output, linear time-invariant plants, the same characteristics are usually present in the controller, and both plant and controller are modeled by their respective transfer functions, $G(s)$ and $H(s)$. However, the plant model is never exact and actually contains some unmodeled dynamics that we regard as random process noise. Furthermore, the plant output, $y(t)$, is neither sensed exactly nor fed back to the controller in a pristine form, but generally contains an additive random error called the measurement noise.

The design objective of the control system is to maintain the plant output, $y(t)$, close to the desired (or commanded) output, $y_d(t)$, by driving the error, $e(t) = y_d(t) - y(t)$, through the application of control input, $u(t)$, in the presence of unmodeled disturbance inputs, $p(t)$ (process noise), and $m(t)$ (measurement noise). All the variables are assumed to be Laplace transformable [20]. The Laplace transforms are indicated by upper-case symbols of the respective time-domain variables. In the basic controller design, we ignore the presence of the disturbance inputs; thus from Fig. 3.8, we have

$$\begin{aligned} E(s) &= Y_d(s) - Y(s) \\ Y(s) &= G(s)U(s) \\ U(s) &= H(s)E(s), \end{aligned} \tag{3.64}$$

which yield the following transfer function of the closed-loop system without disturbances:

$$\frac{Y(s)}{Y_d(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} \tag{3.65}$$

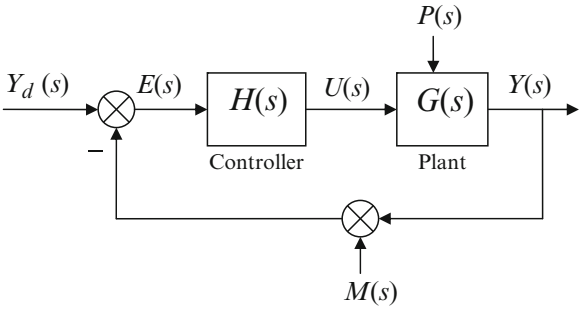


Fig. 3.8 Closed-loop, SISO, LTI control system

The closed-loop poles are the roots of the following characteristic equation:

$$1 + G(s)H(s) = 0, \tag{3.66}$$

If the command signal, $y_d(t)$, is zero the control system is called a regulator; otherwise, the control system is referred to as a tracking system. As a tracking system is expected to respond to a variety of command signals (including step inputs), it must have asymptotic stability as its primary feature. Therefore, the closed-loop poles must all lie in the left-half s-plane. Other closed-loop design features can be derived from the performance and robustness requirements.

3.3.1 Steady-State Error

In order to assess the closed-loop performance, we must know what are the likely desired outputs (command signals) the system is expected to respond to. For a given desired output, $y_d(t)$, the *steady-state error*, e_{ss}, can be obtained by applying the *final-value theorem* of the Laplace transform [20]:

$$e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} \frac{sY_d(s)}{1 + G(s)H(s)}. \tag{3.67}$$

The steady-state error crucially depends upon the *type* of the control system shown in Fig. 3.8. The system's type is defined as the number of pure integrators (or poles at $s = 0$) of the open-loop transfer function, $G(s)H(s)$. Thus, a control system having one pole of $G(s)H(s)$ at origin is called Type I system, the one with two poles at origin is said to be of Type II, and so on. As the most common control application involves tracking a step command, $Y_d(s) = y_0/s$, the steady-state error in such a case is given by:

$$e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} \frac{sy_0Y_d(s)/s}{1 + G(s)H(s)} = \frac{y_0}{1 + G(0)H(0)}. \tag{3.68}$$

Clearly, if the system is of Type I (or higher), its response to a step command will result in a zero steady-state error. For a ramp command, $Y_d(s) = y_0/s^2$, a control system of Type II (or higher) will produce a zero steady-state error. If the open-loop plant, $G(s)$, has an insufficient number of poles at origin then its type can be increased by putting poles at $s = 0$ in the controller transfer function, $H(s)$.

Example 3.10. A plant has the following transfer function:

$$G(s) = \frac{s + 1}{s - 1}. \tag{3.69}$$

Devise a controller in the feedback configuration of Fig. 3.8 such that the closed-loop response to a unit step input has zero steady-state error with a settling-time of 3 s.

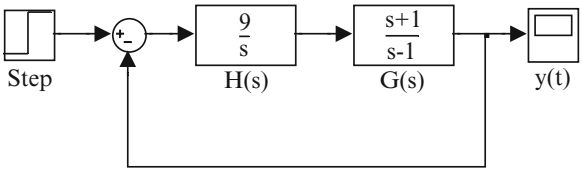


Fig. 3.9 Simulink block-diagram of the control system of Example 3.10

The plant is unstable due to the pole, $s = 1$. Furthermore, it is of Type 0 because there are no poles at the origin. Hence, we select a controller transfer function for both stabilizing and increasing the type of the control system as follows:

$$H(s) = \frac{k}{s}, \tag{3.70}$$

where the pole $s = 0$ adds integral action to the forward path. The closed-loop transfer function is then given by:

$$\frac{Y(s)}{Y_d(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{k(s+1)}{s^2 + (k-1)s + k}. \tag{3.71}$$

Thus, the closed-loop characteristic equation is the following:

$$1 + G(s)H(s) = s^2 + (k-1)s + k = 0,$$

which implies a natural frequency, $\omega_n = \sqrt{k}$, and damping ratio,

$$\zeta = \frac{k-1}{2\sqrt{k}}.$$

For meeting the settling-time requirement, we choose $k = 9$, which yields, $\omega_n = 3$ rad/s, $\zeta = 4/3$. The system's step response is simulated with the Simulink model of Fig. 3.9, and the result plotted in Fig. 3.10. Note the maximum overshoot of 17.3%. A reduction in the value of k increases both the maximum overshoot and the settling-time. For having a better control over the closed-loop performance, one must be able to change both the natural frequency and damping ratio in an independent manner, which requires an additional controller constant.

3.3.2 *Proportional-Integral-Derivative Control*

For the fedback control system of Fig. 3.8, the simplest control law is *proportional control*,

$$u = ke = k(y_d - y), \tag{3.72}$$

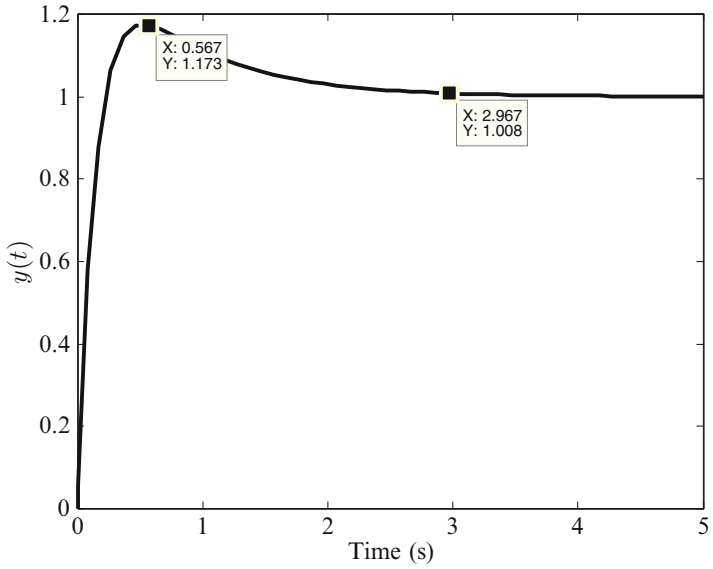


Fig. 3.10 Closed-loop step response of the control system of Example 3.10

where k is a positive constant, called the *feedback gain*. With a suitable choice of the gain, an acceptable closed-loop performance may be obtained. However, this is not always guaranteed, as the proportional feedback can amplify even small errors (perhaps due to the measurement noise), leading to a high sensitivity to disturbances. Thus, the proportional controller often acts analogous to a spring. In order to damp out the oscillations caused by the proportional gain, either the plant must have inherent damping or an artificial damping mechanism must be provided by the controller. Therefore, a *proportional-derivative* (PD) control,

$$u = k_1 e + k_2 \dot{e}, \tag{3.73}$$

is often applied instead of proportional control. A properly designed PD controller produces a satisfactory transient behavior, but can cause a large steady-state error, $e_{ss} \doteq \lim_{t\to\infty} e(t)$, in plants which do not have a sufficient number of poles at the origin ($s = 0$). In such cases, it becomes imperative to add a controller pole at the origin, which translates into an integral action in the time domain and increased type of the system. The resulting control law is termed *proportional-integral-derivative* (PID), and is given by:

$$u = k_1 e + k_2 \dot{e} + k_3 \int e dt. \tag{3.74}$$

The controller gains, k_1, k_2, k_3, are suitably chosen by a design process called *PID tuning* in order to achieve an acceptable response, as well as a low sensitivity

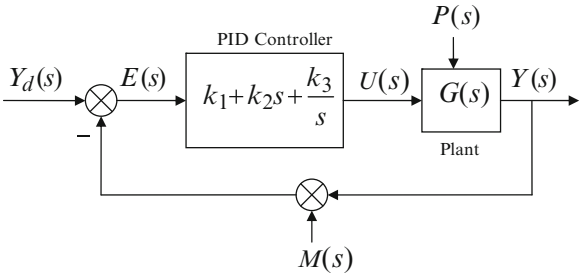


Fig. 3.11 Block diagram of a proportional-integral-derivative (PID) control system

to unmodeled disturbances. Due to their excellent closed-loop performance and ease of design, PID controllers are commonly used in SISO control systems. They are especially useful in controlling a second-order plant as the controller gains, k_1, k_2, k_3, provide the necessary parameters for suitably placing the poles of the resulting third-order control system. A block diagram of PID controller is shown in Fig. 3.11.

Example 3.11. As an application of the PID control system, consider the aircraft long-period (phugoid) longitudinal dynamics (Chap. 4) involving a slow oscillation in airspeed, $y(t)$ m/s, caused by elevator deflection, $u(t)°$ with the following transfer function:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{-s^2 + 99.5s + 50}{2{,}000s^2 + 2s + 10}. \tag{3.75}$$

It is desired to put in place a controller that enables a unit step change in airspeed by elevator deflection in about 30 s.

Since the plant does not have a pole at origin, we require integral action by the controller in order to make the steady-state error vanish for a step command. Furthermore, the plant has very large settling-time (8,000 s) due to low damping. Hence, we choose a proportional-integral (PI) controller with the transfer function

$$H(s) = k_1 + \frac{k_2}{s} \tag{3.76}$$

in order to both reduce the settling-time (with some help from the numerator polynomial of G(s)) and the steady-state error. The resulting closed-loop system is of third-order with transfer function given by:

$$\begin{aligned} \frac{Y(s)}{Y_d(s)} &= \frac{G(s)H(s)}{1 + G(s)H(s)} \\ &= \frac{(-s^2 + 99.5s + 50)(k_1 s + k_2)}{(2{,}000 - k_1)s^3 + (99.5k_1 - k_2 + 2)s^2 + (50k_1 + 99.5k_2 + 10)s + 50k_2}. \end{aligned} \tag{3.77}$$

Clearly, we cannot place all three poles at desired locations with only two controller constants. Happily, we don't need to do so as we have only the settling-time requirement, which can be easily met by properly selecting only the *real parts* of the closed-loop poles. To this end, we write the desired closed-loop characteristic polynomial as follows:

$$1 + G(s)H(s) = (s + a)(s^2 + 2\zeta\omega_n s + \omega_n^2), \tag{3.78}$$

and note that the real parts of the closed-loop poles are a and $-\zeta\omega_n$. Since we want the dominant closed-loop dynamics to be of second-order (as the plant), the real pole, $s = -a$, must be placed further into the left-half s-plane than the complex conjugate poles, $s_{2,3} = -\zeta\omega_n \pm \mathrm{i}\omega_n\sqrt{1-\zeta^2}$. Thus, we specify $\zeta\omega_n = 4/30$ ($\approx 4/t_s$) and $a = 1/3$. Upon comparing the desired characteristic polynomial with the numerator polynomial of the closed-loop transfer function (3.77) we have

$$\begin{aligned} k_1 &= 12.05425 \\ k_2 &= 8.633789 \\ \omega_n &= \sqrt{\frac{150k_2}{2000 - k_1}} = 0.8071 \text{ rad/s} \\ \zeta &= 0.1652. \end{aligned} \tag{3.79}$$

In order to verify our design, we compute the closed-loop step response by the following MATLAB-CST commands:

```
>> num=[-1 99.5 50];den=[2000 2 10];sys=tf(num,den) % Plant
>> k2=8.633789; k1=12.05425; % Controller constants
>> numb=conv(num,[k1 k2]);denb=[2000-k1 99.5*k1-k2+2 50*k1+99.5*k2+10 50*k2];
   sysb=tf(numb,denb)
% Closed-loop system

Transfer function: -12.05 s^3 + 1,191 s^2 + 1,462 s + 431.7
---------------------------------------
 1,988 s^3 + 1,193 s^2 + 1,472 s + 431.7

 >> [y,t]=step(sysb);

 >> syserr=tf(conv([1 0],den),denb)

Transfer function:
      2,000 s^3 + 2 s^2 + 10 s
-----------------------------------
1988 s^3 + 1,193 s^2 + 1,472 s + 431.7

>> [ye,te]=step(syserr); % System error
```

From the system error calculated earlier, we can estimate the required control input using the Simulink block diagram of Fig. 3.12. The closed-loop step response and control input are plotted in Fig. 3.13. The settling-time is quite close to the desired value of 30 s, while the maximum elevator deflection is slightly greater than 10° – an acceptable magnitude for a linear control system.

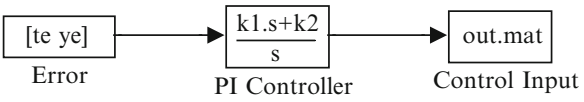


Fig. 3.12 Simulink block diagram for PI control input simulation

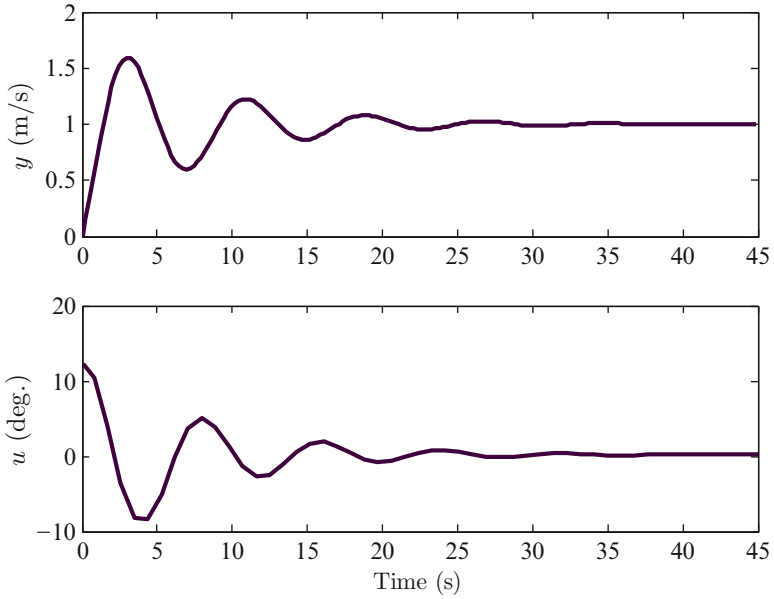


Fig. 3.13 Closed-loop step response for aircraft with PI airspeed tracking system

3.3.2.1 Roll Autopilot for Aircraft and Banking Missiles

Rolling motion of high-speed aircraft and bank-to-turn missiles requires automatic control that is often designed using single-input techniques. In such applications, the wings/fins are of much smaller span than a conventional aircraft, hence the aerodynamic coupling among roll, yaw, and sideslip can be considered negligible. Even when there is a significant roll-yaw-sideslip coupling due to larger span wings, the pure rolling motion is well damped and has a much smaller time-scale than the yaw-sideslip dynamics. Thus, a predominantly rolling mode generally exists – called the *pure rolling mode* – and a roll autopilot exercises control over it for quickly achieving a desired bank angle. Consider the pure roll dynamics of the vehicle represented by:

$$\begin{aligned} J_{xx}\ddot{\sigma} &= -L_p p + L_{\mathrm{A}}\delta_{\mathrm{A}} \\ p &= \dot{\sigma} \end{aligned} \tag{3.80}$$

resulting in the following transfer function:

$$\frac{\Sigma(s)}{\Delta_A(s)} = \frac{L_{\mathrm{A}}}{J_{xx}s^2 - L_p s}. \tag{3.81}$$

As the roll dynamics plant lacks asymptotic stability, a PD feedback control is required to ensure a well damped response to a step banking command, $\sigma_d(t)$, resulting in the following closed-loop transfer function:

$$\frac{\Sigma(s)}{\Sigma_d(s)} = \frac{\frac{L_{\mathrm{A}}}{J_{xx}}(K_P + K_D s)}{s^2 + \left(K_D \frac{L_{\mathrm{A}}}{J_{xx}} - \frac{L_p}{J_{xx}}\right)s + \frac{L_{\mathrm{A}}}{J_{xx}}K_P}. \tag{3.82}$$

A suitable choice of the controller gains, K_P, K_D, results in the desired settling-time and maximum overshoot. Due to the open-loop transfer function having a pole at the origin, the steady-state error to a step command is zero. The bank angle, $\sigma(t)$, is sensed and fedback through a vertical gyro whose gain is included in the PD roll controller.

In addition to the roll dynamics, the aileron's rotation about its hinge is modeled by the following second-order dynamics:

$$J_A \ddot{\delta}_A = H_{\dot{\delta}_{\mathrm{A}}} \dot{\delta}_{\mathrm{A}} + H_{\delta_{\mathrm{A}}} \delta_{\mathrm{A}} + K_A u, \tag{3.83}$$

where $u(t)$ is the applied input voltage to the DC motor whose dynamics is modeled simply by the gain, K_A. As the aileron hinge-moment damping derivative, $H_{\dot{\delta}_{\mathrm{A}}}$, is seldom sufficient to produce acceptable transient characteristics, a closed-loop control system – called *aileron servo* – is invariably required. The servo must also provide integral action for ensuring zero steady-state error to a step command, $\delta_{Ad}(t)$. Hence, a PID control with gains k_1, k_2, k_3 is selected for the aileron servo, which has the following closed-loop transfer function:

$$\frac{\Delta_A(s)}{\Delta_{Ad}(s)} = \frac{\frac{K_A}{J_A}(k_3 + k_1 s + k_2 s^2)}{s^3 + \left(k_2 \frac{K_A}{J_A} - \frac{H_{\dot{\delta}_{\mathrm{A}}}}{J_A}\right)s^2 + \left(\frac{K_A}{J_A}k_1 - \frac{H_{\delta_{\mathrm{A}}}}{J_A}\right) + \frac{K_A}{J_A}k_3}. \tag{3.84}$$

The servo must have a faster transient response than that of the roll control loop. A likely choice for the gains, k_1, k_2, k_3, is such that the servo dynamics is ten times faster than the closed-loop roll dynamics (3.82), resulting in the following servo poles:

$$\begin{aligned} s_1 &= -\zeta\omega \\ s_{2,3} &= -\zeta\omega \pm \omega\sqrt{1-\zeta^2}, \end{aligned} \tag{3.85}$$

where $\zeta = 1/\sqrt{2}$ and

$$\omega = 10\sqrt{\frac{L_A}{J_{xx}}K_P}. \tag{3.86}$$

There is usually a limit imposed on the motor input voltage, $|u(t)| \leq u_{\max}$. Such a limit – called *saturation* – renders the control system nonlinear, eventhough its design is based upon an LTI system. Alternatively, saturation limits may be individually imposed on the aileron deflection, $\delta_A(t)$, and its rate, $\dot{\delta}_A(t)$. Input saturation as well as process and measurement noise cause a departure of the autopilot from its nominal design performance that should be carefully analyzed before implementation.

3.3.3 Feedforward/Feedback Tracking

An alternative to PID control is feedforward/feedback (FF/FB) tracking of command signals wherein the integral action of the PID control is replaced by a suitable feedforward path in order to reduce the steady-state error. Consider the block diagram of a SISO FF/FB control system shown in Fig. 3.14, with output, $y(t)$, and error, $e(t) = y_d(t) - y(t)$. The plant transfer function, $G(s)$, does not have an adequate number of poles at the origin in order to make $e_{ss} \to 0$ for a given command signal (desired output), $y_d(t)$. Furthermore, the feedback controller, $H(s)$, lacks integral action, to compensate for which a feedforward path, $F(s)$ is added. The closed-loop transfer function is thus the following:

$$\frac{Y(s)}{Y_d(s)} = \frac{G(s)[F(s) + H(s)]}{1 + G(s)H(s)}, \tag{3.87}$$

and the error is given by

$$\frac{E(s)}{Y_d(s)} = \frac{1 - G(s)F(s)}{1 + G(s)H(s)}. \tag{3.88}$$

Clearly, the closed-loop characteristic polynomial, $1 + G(s)H(s)$, is unaffected by the feedforward path. However, the steady-state error,

$$e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0}\left\{sY_d(s)\frac{1 - G(s)F(s)}{1 + G(s)H(s)}\right\}, \tag{3.89}$$

is greatly influenced by $F(s)$ since $G(s)H(s)$ does not have an adequate number of poles at $s = 0$ for e_{ss} to vanish identically. Therefore, the primary function of the feedforward path is to reduce the steady-state error, by providing a similar effect as that of a pole at origin in $G(s)H(s)$.

The feedforward path requires that the command signal, $y_d(t)$, be measured separately from the output, $y(t)$. This requirement is usually more stringent than the

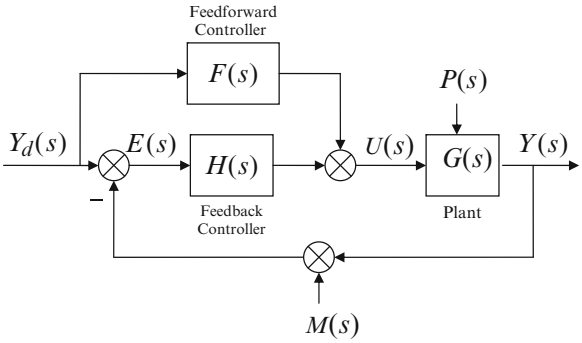


Fig. 3.14 Block diagram of a feedforward/feedback (FF/FB) tracking system

error measurement required in a pure feedback control. Typically, a command signal is either operator specified or generated as an output of another control system; hence, it can often be measured quite easily.

Example 3.12. An air-to-air missile autopilot is tasked to track a step normal acceleration signal, $Y_d(s) = 0.1/s$ g, commanded by an outer guidance loop. The transfer function of the missile's rigid body, short-period longitudinal dynamics with control-surface deflection input is given by:

$$G(s) = -\frac{0.05(s^2 - 50)}{s^2 + s + 10} \text{ g/deg.} \tag{3.90}$$

A proportional feedback controller, $H(s) = 19.75$, is selected in order to have a closed-loop settling-time of 0.1 s, with the attendant damping ratio, $\zeta = 0.58$. However, as neither $G(s)$ nor $H(s)$ has a pole at origin, the steady-state error for a step command is nonzero. Hence, a proportional feedforward controller, $F(s) = K$, is added such that $e_{ss} = 0$, which requires

$$e_{ss} = \lim_{s\to 0} sE(s) = \frac{1 - KG(0)}{1 + 19.75G(0)} = 0, \tag{3.91}$$

or

$$K = \frac{1}{G(0)} = 4. \tag{3.92}$$

The responses of the plant, the pure feedback control system, and the FF/FB system are compared in Fig. 3.15 generated using the following MATLAB-CST commands:

```
>> num=-.05*[1 0 -50];den=[1 1 10];sys=tf(num,den) % Plant

Transfer function:
-0.05 s^2 + 2.5
---------------
 s^2 + s + 10
```

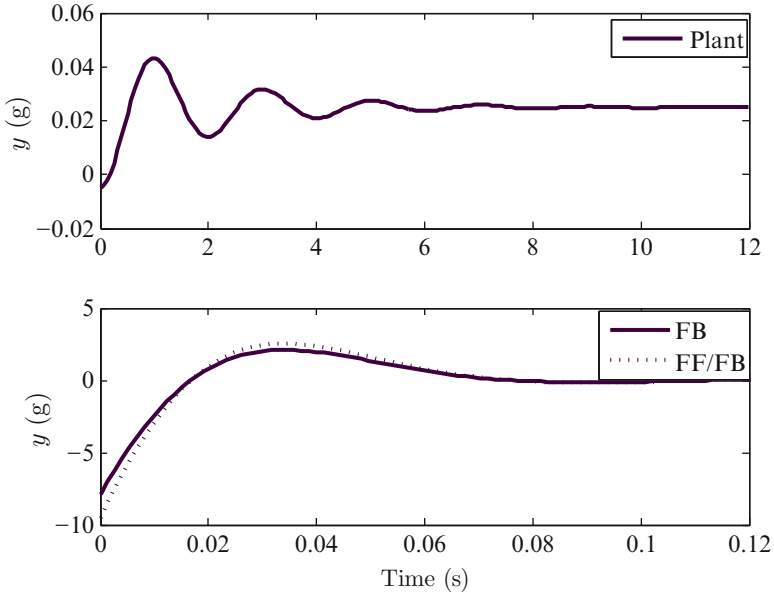


Fig. 3.15 Step response of an air-to-air missile longitudinal dynamics plant compared with a pure feedback (FB) controller and a feedforward/feedback (FF/FB) tracking system

```
>> numfb=19.75*num;denfb=[(1-0.05*19.75) 1 (10+2.5*19.75)];
   sysfb=tf(numfb,denfb)% FB system

Transfer function:
 -0.9875 s^2 + 49.38
-----------------------
0.0125 s^2 + s + 59.38

>> numffb=(4+19.75)*num;sysffb=tf(numffb,denfb) % FF/FB system

Transfer function:
  -1.188 s^2 + 59.38
-----------------------
0.0125 s^2 + s + 59.38

>> [y,t]=step(sys);
>> [yb,tb]=step(sysfb);
>> [yf,tf]=step(sysffb);
>> subplot(211),plot(t,y/10),subplot(212),plot(tb,yb/10,tf,yf/10)
```

The steady state outputs of the plant, the FB and the FF/FB systems to the applied command $Y_d(s) = 0.1/s$ g are the following:

$$y_{ss} = \lim_{s\to 0} sY_d(s)G(s) = 0.025 \text{ g} \qquad \text{(Plant)}$$

$$y_{ss} = \lim_{s\to 0} sY_d(s)\frac{G(s)H(s)}{1+G(s)H(s)} = (0.1)\frac{49.38}{59.38} = 0.0832 \text{ g} \qquad \text{(FB)}$$

$$y_{ss} = \lim_{s\to 0} sY_d(s)\frac{G(s)[F(s)+H(s)]}{1+G(s)H(s)} = (0.1)\frac{59.38}{59.38} = 0.1 \text{ g} \qquad \text{(FF/FB)}$$

Thus, only the FF/FB system has a zero steady-state error, as expected.

3.3.4 Robustness Analysis from Frequency Response

Robustness of a control system is related to its sensitivity to unmodeled dynamics, i.e., part of the behavior of an actual control system, which is not included in the mathematical model of the control system. As we have to deal with actual control systems in which it is difficult to mathematically model all relevant physical processes, we are always left with the question: will the control system based upon a mathematical model really work? We have already discussed the presence of random disturbances in the form of process and measurement noise. If a control system meets its performance and stability objectives in the presence of such disturbances then the control system is said to be robust. Hence, robustness is a desirable property that dictates whether a control system is immune to uncertainties in its mathematical model. More specifically, robustness can be subdivided into *stability robustness* and *performance robustness*, depending upon whether we are interested in range of controller parameters for which either the stability, or the performance of the control system is guaranteed.

Taking the case of a feedback control system with, we can write the closed-loop characteristic equation as follows:

$$1 + G(s)H(s) = 0, \tag{3.93}$$

which becomes the following on the imaginary axis of the Laplace domain (frequency domain), $s = \mathrm{i}\omega$:

$$1 + G(\mathrm{i}\omega)H(\mathrm{i}\omega) = 0. \tag{3.94}$$

Note that (3.94) satisfies the closed-loop system's characteristic equation *if and only if* any of the roots of (3.93) falls on the imaginary axis, $s = \mathrm{i}\omega$. As the closed-loop system would become unstable if any of its poles crosses over into the right-half s-plane (Chap. 1), the margin of stability (i.e., stability robustness) is measured by how far the poles are from crossing the imaginary axis, or in other words, how shy is the locus of the open-loop frequency response, $G(\mathrm{i}\omega)H(\mathrm{i}\omega)$, from the point -1. In the Laplace domain, the point -1 is denoted by the gain of unity and a phase of $\pm 180°$. Thus, the stability robustness is determined by the proximity of the open-loop frequency response from a gain of unity and a phase of $\pm 180°$. The gain of the frequency response when its phase is $\pm 180°$ is thus called *gain margin* and indicates the factor by which the controller gain, $|H(\mathrm{i}\omega)|$, can be increased before the closed-loop system becomes unstable. Similarly, the phase difference of

$G(\mathrm{i}\omega)H(\mathrm{i}\omega)$ from $\pm 180°$ when its gain is unity is called the *phase margin*. Note that a negative gain margin indicates an unstable closed-loop system. The frequency at which the phase of $G(\mathrm{i}\omega)H(\mathrm{i}\omega)$ crosses the $\pm 180°$ line is called the *phase crossover frequency*, while the frequency at which $|G(\mathrm{i}\omega)H(\mathrm{i}\omega)|$ becomes unity is called the *gain crossover frequency*.

Example 3.13. Let us find the gain and phase margins of the aircraft longitudinal dynamics plant of Example 3.9. From the frequency response shown in Fig. 3.6, we note that the phase of $\pm 180°$ is never crossed, which implies an infinite gain margin. The gain of unity (0 dB) is crossed at the gain crossover frequency of 0.74 rad/s, at which the phase is approximately 77°. Therefore, the phase margin is $180 - 77 = 103°$.

Another important measure of a control system's robustness is its ability to reject high-frequency process and measurement noise. A system that has a rapid decline of the gain, $|G(\mathrm{i}\omega)|$, with frequency (roll-off) beyond the bandwidth of the system, is little affected by random, high-frequency noise inputs. An adequate roll-off is considered to be about -20 dB per decade of freqency in rad/s.

3.3.4.1 Lead and Lag Compensation for Robust Design

Often the controller transfer function must be compensated for the unmodeled dynamics of a control system, such as higher (neglected) modes of the plant and actuator nonlinearities, which are invariably present in the actual system. Such a compensation may involve placing a simple, linear system (called a *filter*) in series at the controller output. Since the open-loop DC gain (that has already been selected for a small steady-state error) must be unaffected by a filter, its transfer function should be such that only a phase change occurs in a desired frequency range. In order to increase or decrease the speed of the closed-loop step response, the filter must introduce either a phase-lead or phase-lag, respectively. Hence, such a filter is appropriately called either a lead or a lag compensator and can be used to increase the gain and phase margins or the roll-off of the linear control system. The compensating filters are generally chosen to be of first-order to avoid resonant peaks associated with second- and higher-order filters.

A first-order filter has the following transfer function:

$$F(s) = \frac{s+b}{as+b}, \tag{3.95}$$

where a, b are real coefficients. For a lag compensator, we have $a > 1$, whereas $a < 1$ for the lead compensator. A lag compensator is useful in reducing the steady-state error of Type 0 plants (i.e., plants without any pure integrators), and decreasing the gain of the closed-loop system at high frequencies (which is desirable for reducing the sensitivity to the measurement noise). However, lag compensation slows down the closed-loop transient response. Lag compensation is relatively

simple to use, because the passive circuit through which it can be implemented is quite inexpensive. On the other hand, a lead compensator is useful in increasing the speed of the closed-loop response, and increasing the phase margin of the closed-loop system, which also results in smaller overshoots of the transient response. Lead compensation usually requires amplification of error signals through a relatively expensive electrical circuit. Also, lead compensation increases the gain at high frequencies, which is undesirable due to increased sensitivity to measurement noise.

In order to combine the useful features of both lead and lag compensators, one can have a lead–lag compensator with the following transfer function:

$$F(s) = \left(\frac{s + b_1}{s + b_1/\alpha}\right)\left(\frac{s + b_2}{s/\alpha + b_2}\right), \tag{3.96}$$

where $\alpha > 1$. Note that the lead–lag compensator applies a phase reduction in the approximate frequency range, $b_1/\alpha \leq \omega \leq b_1$, and also increases the DC gain of the system, while the increasing the phase in the approximate frequency range $b_2 \leq \omega \leq \alpha b_2$. In the limit $\alpha \to \infty$, the lead–lag compensator becomes a PID controller. Hence, one can approximate PID controller by a passive lead–lag electrical network with a large value of the positive constant, α [20].

3.4 Multivariable Control Design

The concept of transfer function ceases to be useful for multi-input, multi-output (MIMO) plants, for which the state-space methods are more appropriate. The design methodology for MIMO control systems is thus carried out directly in the time domain, rather than in the Laplace (or frequency) domain. Consider the plant dynamics expressed in a linear, state space form as follows:

$$\dot{\mathbf{x}} = \mathsf{A}\mathbf{x} + \mathsf{B}\mathbf{u}, \tag{3.97}$$

where $\mathbf{x}(t)$ is the *state vector*, $\mathbf{u}(t)$ the control input vector, and $\mathsf{A}(t), \mathsf{B}(t)$ are the state space coefficient matrices that could be time varying. However, for the time being, we will confine our attention to a linear plant with constant coefficient matrices, A, B, i.e., linear time-invariant (LTI) system.

3.4.1 Regulator Design by Eigenstructure Assignment

Design of a linear, state feedback regulator for the LTI plant of (3.97) with the control law,

$$\mathbf{u} = -\mathsf{K}\mathbf{x}, \tag{3.98}$$

is possible by assigning a structure for the eigenvalues and eigenvectors of the closed-loop dynamics matrix, $\mathsf{A}-\mathsf{BK}$. In case of single-input plants, this merely involves selecting the locations for the closed-loop poles (*pole-placement*) by the following *Ackermann's formula* that yields the desired closed-loop characteristics [20]:

$$\mathsf{K} = (\mathbf{a_d} - \mathbf{a})(\mathsf{PP}')^{-1}, \tag{3.99}$$

where a is the row vector formed by the coefficients, a_i, of the plant's characteristic polynomial in *descending order* [$\mathbf{a} = (a_n, a_{n-1}, \ldots, a_2, a_1)$]:

$$|s\mathsf{I} - \mathsf{A}| = s^n + a_n s^{n-1} + a_{n-1}s^{n-2} + \cdots + a_2 s + a_1, \tag{3.100}$$

a_d is the row vector formed by the characteristic coefficients of the closed-loop system in descending order [$\mathbf{a_d} = (a_{dn}, a_{d(n-1)}, \ldots, a_{d2}, a_{d1})$]:,

$$|s\mathsf{I} - \mathsf{A} + \mathsf{BK}| = s^n + a_{dn}s^{n-1} + a_{d(n-1)}s^{n-2} + \cdots + a_{d2}s + a_{d1}, \tag{3.101}$$

P is the controllability test matrix of the plant, and P' is the following upper triangular matrix:

$$\mathsf{P}' = \begin{pmatrix} 1 & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \\ 0 & 1 & a_{n-1} & \cdots & a_3 & a_2 \\ 0 & 0 & 1 & \cdots & a_4 & a_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & a_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{3.102}$$

Of course, this requires that the plant must be controllable, $|\mathsf{P}| \neq 0$. A popular choice of the closed-loop poles is the *Butterworth pattern* [20] wherein all the poles are equidistant from the origin, $s = 0$. Such a pattern generally requires the least control effort for a given bandwidth, and is thus considered to be optimal placement of poles.

Example 3.14. Find the regulator gain matrix that can move the poles of the following single-input plant to $s_{1,2} = -1 \pm \mathrm{i}$, $s_{3,4} = -2 \pm 2\mathrm{i}$, and $s_{5,6} = -3 \pm 3\mathrm{i}$:

$$\mathsf{A} = \begin{pmatrix} 0 & -1 & 3 & 0 & 1 & -2 \\ -1 & 0.5 & 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 2 & -1 & 1 \\ 0 & -2 & -1 & 0 & 0.5 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}; \quad \mathsf{B} = \begin{pmatrix} -1 \\ 0 \\ 1 \\ 2 \\ 1 \\ -0.5 \end{pmatrix}.$$

We begin by checking the controllability of the plant as follows:

```
>> A=[0 -1 3 0 1 -2; -1 0.5 0 -1 0 1; 0 1 -1 2 -1 1;
      0 -2 -1 0 0.5 1; -1 0 0 0 1 0; 0 -1 0 0 0 1];
>> B=[-1 0 1 2 1 -0.5]';

>> P=ctrb(A,B) % controllability test matrix

P =
   -1.0000    5.0000    9.0000  -22.2500   18.8750   58.9375
         0   -1.5000   -5.2500  -12.6250    4.6875  -12.9063
    1.0000    1.5000   -7.5000   10.2500   30.3750   13.4375
    2.0000   -1.0000    2.0000   17.5000   15.2500  -15.7500
    1.0000    2.0000   -3.0000  -12.0000   10.2500   -8.6250
   -0.5000   -0.5000    1.0000    6.2500   18.8750   14.1875

>> rank(P) % rank of controllability test matrix

ans =     6
```

Thus, P is of full-rank and one can design a regulator by pole-placement.

```
% characteristic coefficients of the plant (in descending order):
>> a=poly(A);a=a(2:7)

%    a_6       a_5       a_4       a_3       a_2       a_1
a =  -1.5000    0.5000   -1.0000   -4.5000   15.0000   -6.0000

>> ad=poly([-1-i,-1+i,-2-2i,-2+2i,-3-3i,-3+3i]);ad=ad(2:7)

% desired characteristic coefficients (in descending order):
%     a_d6  a_d5  a_d4  a_d3   a_d2  a_d1
ad =  288   528   484   240    72    12

% Upper triangular matrix follows:
>> Pprime=[1 a(1) a(2) a(3) a(4) a(5); 0 1 a(1) a(2) a(3) a(4);
           0 0 1 a(1) a(2) a(3); 0 0 0 1 a(1) a(2);
           0 0 0 0 1 a(1); 0 0 0 0 0 1]
Pprime =
    1.0000   -1.5000    0.5000   -1.0000   -4.5000   15.0000
         0    1.0000   -1.5000    0.5000   -1.0000   -4.5000
         0         0    1.0000   -1.5000    0.5000   -1.0000
         0         0         0    1.0000   -1.5000    0.5000
         0         0         0         0    1.0000   -1.5000
         0         0         0         0         0    1.0000

>> K=(ad-a)*inv(P*Pprime) % regulator gain matrix

K =   53.7609   16.3270   52.2065   49.8244  -96.4548  -23.7209
```

It is prudent to check whether the closed-loop poles are indeed at the desired locations, which we do as follows:

```
>> eig(A-B*K) % eigenvalues of the closed-loop system

ans =
  -3.0000 + 3.0000i
  -3.0000 - 3.0000i
  -2.0000 + 2.0000i
  -2.0000 - 2.0000i
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
```

Alternatively, one can employ the MATLAB-Control Systems Toolbox (CST) command *acker* for computing the gains by a single statement:

```
>> K=acker(A,B,[-1-i,-1+i,-2-2i,-2+2i,-3-3i,-3+3i])

K =   53.7609   16.3270   52.2065   49.8244  -96.4548  -23.7209
```

The pole-placement method is inapplicable to multi-input plants, which have many more controller gains to be found than the number of equations available from the pole locations. In such a case, we need additional equations that can be derived from the shape of the eigenvectors using the method of *eigenstructure assignment*. A popular method in this regard is the *robust pole assignment method* of [10] wherein the eigenvectors, $\mathbf{v_i}$, $i = 1, 2, \ldots, n$, corresponding to the eigenvalues, λ_i, respectively, and satisfying the eigenvalue problem,

$$(\mathsf{A} - \mathsf{BK})\mathbf{v_i} = \lambda_i \mathbf{v_i}, \tag{3.103}$$

are chosen such that the modal matrix,

$$\mathsf{V} = (\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}), \tag{3.104}$$

is as well-conditioned[3] as possible. An eigenstructure assignment algorithm of [10] is coded in the MATLAB-CST command called *place*.

Example 3.15. Repeat the regulator design of Example 3.14 if the matrix B is replaced by the following:

$$\mathsf{B} = \begin{pmatrix} -1 & 0 \\ 0 & -0.5 \\ 1 & 0 \\ 0 & -1 \\ 2 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

The eigenstructure assignment is carried out by the MATLAB-CST command, *place*, as follows:

```
>> B=[-1  0;0 -0.5;1  0;0  -1;2  0;0  0.5];
>> [K,prec]=place(A,B,[-1-i,-1+i,-2-2i,-2+2i,-3-3i,-3+3i])

K =
   11.0966  -30.3800    3.6203   18.3561    7.6079   -0.5582
   -8.7649   20.7055   -4.3927  -13.6522   -4.1298    4.9219

prec = 13 % indicates precision of pole-placement (13th decimal place)
```

[3]Conditioning of a square matrix refers to how close the matrix is from being singular (i.e., has a zero determinant). A scalar measure called the *condition number* is assigned to the matrix that reflects its conditioning. A large condition number implies the matrix is close to being singular.

```
>> eig(A-B*K) % eigenvalues of the closed-loop system

ans =
  -3.0000 + 3.0000i
  -3.0000 - 3.0000i
  -2.0000 + 2.0000i
  -2.0000 - 2.0000i
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
```

3.4.2 *Linear, Quadratic Regulator*

An alternative design strategy to eigenstructure assignment for MIMO control systems is the method of linear optimal control. An optimal state feedback regulator with gain matrix, K, is designed with the control law,

$$\mathbf{u} = -\mathsf{K}\mathbf{x}, \tag{3.105}$$

for minimizing the following quadratic, infinite time objective function with symmetric cost parameter matrices, (Q, R),

$$\mathcal{J} = \frac{1}{2}\int_0^\infty \left[\mathbf{x}^{\mathrm{T}}(\tau)\mathsf{Q}\mathbf{x}(\tau) + \mathbf{u}^{\mathrm{T}}(\tau)\mathsf{R}\mathbf{u}(\tau)\right] \mathrm{d}\tau, \tag{3.106}$$

subject to the state equation constraint, (3.97). The stabilizing steady state solution for the feedback gain matrix is given by (Appendix A)

$$\mathsf{K} = \mathsf{R}^{-1}\mathsf{B}^{\mathrm{T}}\mathsf{S}, \tag{3.107}$$

where S is the constant, symmetric, positive semidefinite solution of the following *algebraic Riccati equation*:

$$\mathsf{S}\mathsf{A} + \mathsf{A}^{\mathrm{T}}\mathsf{S} - \mathsf{S}\mathsf{B}\mathsf{R}^{-1}\mathsf{B}^{\mathrm{T}}\mathsf{S} + \mathsf{Q} = 0. \tag{3.108}$$

A sufficient condition for the existence of a unique, positive semidefinite solution to (3.108) is that the plant (A, B) is controllable, R is positive definite, and Q is positive semidefinite (Appendix A). A regulator designed in such a way is termed linear, quadratic regulator (LQR).

Example 3.16. Re-design the regulator for the multi-input plant of Example 3.15 by LQR method with the cost matrices, $\mathsf{Q} = \mathsf{I}_{6\times 6}, \mathsf{R} = \mathsf{I}_{2\times 2}$.

The design is carried out by the MATLAB-CST command *lqr* as follows:

```
>> [K,S,E]=lqr(A,B,eye(6),eye(2)) % LQR design

K =
    6.0852  -17.1092    2.2125   10.2854    5.2204   -2.0313
   -6.3399   14.1600   -3.1633  -10.0378   -2.9616    4.5359
```

```
% Positive semi-definite solution to algebraic Riccati equation:

S =
   18.5603  -46.3295    8.2108   29.7079    8.2174    0.4066
  -46.3295  122.9885  -18.5601  -77.0397  -22.4393   -2.7710
    8.2108  -18.5601    4.8149   12.8668    2.8041    0.8470
   29.7079  -77.0397   12.8668   49.9388   13.5633    2.7622
    8.2174  -22.4393    2.8041   13.5633    5.3168   -1.2358
    0.4066   -2.7710    0.8470    2.7622   -1.2358   11.8252

% Eigenvalues of the closed-loop system:

E =
  -2.3539 + 0.9632i
  -2.3539 - 0.9632i
  -1.2325 + 1.9804i
  -1.2325 - 1.9804i
  -1.5605 + 0.5196i
  -1.5605 - 0.5196i
```

Thus, the new regulator design results in closed-loop eigenvalues, $s_{1,2} = -2.3539 \pm 0.9632i$, $s_{3,4} = -1.2325 \pm 1.9804i$, and $s_{5,6} = -1.5605 \pm 0.5196i$.

A variation of the state-weighted LQR design is the output weighted LQR (or LQRY) problem with the following objective function:

$$\mathcal{J} = \frac{1}{2}\int_0^\infty \left[\mathbf{y}^{\mathrm{T}}(\tau)\bar{\mathsf{Q}}\mathbf{y}(\tau) + \mathbf{u}^{\mathrm{T}}(\tau)\bar{\mathsf{R}}\mathbf{u}(\tau)\right]\mathrm{d}\tau, \tag{3.109}$$

where $\mathbf{y}(t)$ satisfies the following output equation:

$$\mathbf{y} = \mathsf{C}\mathbf{x} + \mathsf{D}\mathbf{u}. \tag{3.110}$$

The LQRY problem is more general than a corresponding LQR problem, because it results in a cross-weighting term between state and control variables in its objective function (Appendix A):

$$\mathbf{y}^{\mathrm{T}}\bar{\mathsf{Q}}\mathbf{y} + \mathbf{u}^{\mathrm{T}}\bar{\mathsf{R}}\mathbf{u} = \mathbf{x}^{\mathrm{T}}\mathsf{Q}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathsf{L}\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathsf{L}^{\mathrm{T}}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathsf{R}\mathbf{u}. \tag{3.111}$$

However, the LQRY problem being more general requires a greater care in selecting the cost parameters for a positive semidefinite Riccati solution, compared to those of the LQR method. Reference [20] discusses the derivation of the algebraic Riccati equation for the LQRY problem, while the MATLAB-CST contains the dedicated *lqry* command for solving the same.

3.4.2.1 Frozen LQR Design for a Time-Varying Plant

While we have derived the LQR gain matrix for the LTI plant, the same approach can be extended to a slowly time varying plant. In such a case, we allow a variation of the cost coefficients, Q, R, with time and express the cost function as follows:

$$\mathcal{J}_\infty(t) = \frac{1}{2}\int_0^\infty \left[\mathbf{x}^{\mathrm{T}}(\tau)\mathsf{Q}(\tau)\mathbf{x}(\tau) + \mathbf{u}^{\mathrm{T}}(\tau)\mathsf{R}(\tau)\mathbf{u}(\tau)\right]\mathrm{d}\tau, \tag{3.112}$$

subject to the state constraint equation

$$\dot{\mathbf{x}}(t) = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)\mathbf{u}(t) \qquad \mathbf{x}(0) = \mathbf{x_0}, \tag{3.113}$$

and the linear, full-state feedback control law

$$\mathbf{u}(t) = -\mathsf{K}_\infty(t)\mathbf{x}(t). \tag{3.114}$$

The general, linear time-varying optimal control problem results in a matrix Riccati differential equation, which must be integrated in time by a nonlinear programming method (Appendix A). However, the derivation of the quasi-steady gain matrix, $\mathsf{K}_\infty(t)$, at a given time involves the assumption that it can stabilize the "time-frozen" plant $\mathsf{A}(t), \mathsf{B}(t)$, at each time instant in the same way as the constant feedback gain matrix would stabilize the LTI plant. Therefore, we have

$$\mathsf{K}_\infty(t) = \mathsf{R}^{-1}(t)\mathsf{B}^{\mathrm{T}}(t)\mathsf{P}(t), \tag{3.115}$$

where $\mathsf{P}_\infty(t)$ is the symmetric, positive semidefinite, quasi-steady solution of the following algebraic Riccati equation:

$$\mathsf{P}_\infty(t)\mathsf{A}(t) + \mathsf{A}^{\mathrm{T}}(t)\mathsf{P}_\infty(t) - \mathsf{P}_\infty(t)\mathsf{B}(t)\mathsf{R}^{-1}(t)\mathsf{B}^{\mathrm{T}}(t)\mathsf{P}_\infty(t) + \mathsf{Q}(t) = \mathsf{0}. \tag{3.116}$$

Thus, the quasi-steady approximation is tantamount to finding a constant, steady state solution to (3.116) at each time instant, provided the variation of $\mathsf{A}, \mathsf{B}, \mathsf{Q}, \mathsf{R}$ with time is much slower than the settling-time of all the transients of the closed-loop dynamics matrix, $\mathsf{A} - \mathsf{BK}$. An efficient choice of Q, R in such a case is by constant matrices. We shall have the occasion to apply the frozen LQR design to spacecraft and rocket plants.

Great caution must be exercised while attempting to apply the frozen LQR approach to a rapidly varying plant, as the closed-loop system could be unstable eventhough the eigenvalues of $\mathsf{A}(t) - \mathsf{B}(t)\mathsf{K}(t)$ are always in the left-half s-plane.

3.4.3 Linear Observers and Output Feedback Compensators

Control systems with output (rather than state) feedback require observers that can reconstruct the missing information about the system's states from the input applied to the plant, and the output fed back from the plant. In other words, an observer mimics the plant by generating an estimated state vector, $\mathbf{x}_o$, instead of the actual plant state vector, $\mathbf{x}$, and supplies it to the regulator. A control system that contains both an observer and a regulator is called a *compensator*. Due to a

decoupling of the observer and plant states in the control system, it is possible to design the regulator and observer separately from each other by what is known as the *separation principle* [20].

3.4.3.1 Full-Order Observer

The output equation,

$$\mathbf{y} = \mathsf{C}\mathbf{x} + \mathsf{D}\mathbf{u}, \tag{3.117}$$

is used in the design of a *full-order observer* with the following state equation:

$$\dot{\mathbf{x}}_o = (\mathsf{A} - \mathsf{LC})\,\mathbf{x}_o + (\mathsf{B} - \mathsf{LD})\,\mathbf{u} + \mathsf{L}\mathbf{y}, \tag{3.118}$$

where $\mathbf{x}_o(t)$ is the estimated state and L the observer gain matrix, provided the plant (A, C) is observable (Chap. 1). The observer gain matrix, L, can be selected in a manner similar to the regulator gain, K, by either eigenstructure assignment for the observer dynamics matrix, $\mathsf{A} - \mathsf{LC}$, or linear, quadratic, optimal control where A is replaced by A^{T}, and B by C^{T}.

The closed-loop control system dynamics with a desired state, $\mathbf{x}_d(t)$, and linear feedforward/feedback control with output feedback (Chap. 1),

$$\mathbf{u} = \mathsf{K}_d\mathbf{x}_d + \mathsf{K}\left(\mathbf{x}_d - \mathbf{x}_o\right), \tag{3.119}$$

is given by the state equation

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_o \end{Bmatrix} = \begin{pmatrix} \mathsf{A} & -\mathsf{BK} \\ \mathsf{LC} & \mathsf{A} - \mathsf{BK} - \mathsf{LC} \end{pmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{x}_o \end{Bmatrix} + \begin{pmatrix} \mathsf{B}(\mathsf{K} + \mathsf{K}_\mathsf{d}) \\ \mathsf{B}(\mathsf{K} + \mathsf{K}_\mathsf{d}) \end{pmatrix} \mathbf{x}_d, \tag{3.120}$$

where K, L are separately designed (by the separation principle) and the feedforward gain matrix, K_d, is selected to ensure that the closed-loop error dynamics is independent of the desired state, $\mathbf{x}_d(t)$, with a given state equation

$$\dot{\mathbf{x}}_d = \mathbf{f}_d(\mathbf{x}_d), \tag{3.121}$$

such that

$$(\mathsf{A} + \mathsf{BK}_\mathsf{d}\mathbf{x}_d) - \mathbf{f}_d(\mathbf{x}_d) = \mathbf{0}. \tag{3.122}$$

Thus, one can design a tracking system for a plant that is both controllable and observable with the available inputs and outputs, respectively, as well as satisfies (3.122) with its desired state vector.

Example 3.17. Design a full-order observer for the plant of Example 3.14 with the output coefficient matrix,

$$\mathsf{C} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

such that the observer's poles are roughly twice as deep in the left-half s-plane as those of the regulator.

The design is carried out by the MATLAB-CST command *lqr* with $\mathsf{Q} = 20\mathsf{I}_{6\times 6}$, $\mathsf{R} = \mathsf{I}_{2\times 2}$ as follows:

```
>> C=[-1 0 0 0 0 0;0 0 0 0 0 1];
>> rank(ctrb(A',C')) % rank of the observability test matrix

ans = 6

% LQR design of full-order observer:
>> [Lp,S,E]=lqr(A',C',10*eye(6),eye(2)); L=Lp'

L =
  -10.2781    1.2284
   13.5273  -11.2823
    7.2149    2.9955
  -33.3984   14.1360
  -54.1484    1.7663
   -1.2284    7.4850

>> eig(A-L*C) % poles of the observer

ans =
  -3.6643 + 1.4221i
  -3.6643 - 1.4221i
  -3.6780
  -2.1044 + 1.5877i
  -2.1044 - 1.5877i
  -1.0478
```

MATLAB-CST has the command *lqe* is dedicated to observer design by the LQR method. The observer designed by LQR method is also referred to as a *Kalman filter*.

3.4.3.2 Reduced-Order Observer

When a part of the plant's state vector can be directly obtained from the output vector, it is unnecessary to estimate the entire state vector by a full-order observer. Consider a plant whose state vector is partitioned as follows:

$$\mathbf{x} = (\mathbf{x_1}^{\mathrm{T}}, \mathbf{x_2}^{\mathrm{T}})^{\mathrm{T}}, \tag{3.123}$$

such that

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathsf{A}_{11}\mathbf{x_1} + \mathsf{A}_{12}\mathbf{x_2} + \mathsf{B}_1\mathbf{u} \\ \dot{\mathbf{x}}_2 &= \mathsf{A}_{21}\mathbf{x_1} + \mathsf{A}_{22}\mathbf{x_2} + \mathsf{B}_2\mathbf{u}. \end{aligned} \tag{3.124}$$

The measurable part of the state vector, $\mathbf{x_1}$, can be directly obtained by inversion of the output equation with a square coefficient matrix, C:

$$\mathbf{y} = \mathsf{C}\mathbf{x_1}. \tag{3.125}$$

The unmeasurable part, $\mathbf{x_2}$, needs estimation by a *reduced-order observer*, and can be expressed as follows:

$$\mathbf{x}_{o2} = \mathsf{L}\mathbf{y} + \mathbf{z}, \tag{3.126}$$

where $\mathbf{z}$ is the state vector of the reduced-order observer with the following state equation:

$$\dot{\mathbf{z}} = \mathsf{F}\mathbf{z} + \mathsf{H}\mathbf{u} + \mathsf{G}\mathbf{y}, \tag{3.127}$$

whose coefficient matrices are determined from the requirement that the estimation error, $\mathbf{e}_o = \mathbf{x_2} - \mathbf{x}_{o2}$, should go to zero in the steady state, irrespective of the control input and the output [20]:

$$\begin{aligned} \mathsf{F} &= \mathsf{A}_{22} - \mathsf{L}\mathsf{C}\mathsf{A}_{12} \\ \mathsf{G} &= \mathsf{F}\mathsf{L} + (\mathsf{A}_{21} - \mathsf{L}\mathsf{C}\mathsf{A}_{11})\,\mathsf{C}^{-1} \\ \mathsf{H} &= \mathsf{B}_2 - \mathsf{L}\mathsf{C}\mathsf{B}_1, \end{aligned} \tag{3.128}$$

with the observer gain L selected by eigenstructure assignment or Kalman filter (see later) approach, such that all the eigenvalues of the observer dynamics matrix, F, are in the left-half s-plane.

Example 3.18. Design a reduced-order observer for the plant of Example 3.17.

We begin by partitioning the state-space coefficient matrices and continue with the aforementioned design steps as follows:

```
>> C=[C(1,1) C(1,6); C(2,1) C(2,6)]

C =
    -1     0
     0     1

>> A11=[A(1,1) A(1,6); A(6,1) A(6,6)]

A11 =
     0    -2
     0     1

>> A12=[A(1,2:5); A(6,2:5)]

A12 =
    -1     3     0     1
    -1     0     0     0

>> A21=[A(2:5,1) A(2:5,6)]

A21 =
    -1     1
     0     1
     0     1
    -1     0

>> A22=A(2:5,2:5)
```

```
A22 =
    0.5000         0   -1.0000         0
    1.0000   -1.0000    2.0000   -1.0000
   -2.0000   -1.0000         0    0.5000
         0         0         0    1.0000

>> B1=[B(1,:);B(2,:)]

B1 =
   -1.0000         0
         0   -0.5000

>> B2=B(2:5,:)

B2 =
         0   -0.5000
    1.0000         0
         0   -1.0000
    2.0000         0

% Reduced-order observer design by LQR method:
>> Lp=lqr(A22',A12'*C',eye(4),eye(2));L=Lp'

L =
    1.6824   -1.7849
    1.6870    0.0196
   -5.5490    1.6157
  -10.2132   -0.1612

>> F=A22-L*C*A12

F =
   -2.9673    5.0473   -1.0000    1.6824
   -0.6674    4.0609    2.0000    0.6870
    5.1648  -17.6471         0   -5.0490
   10.0521  -30.6396         0   -9.2132

>> eig(F) % Poles of reduced-order observer

ans =
  -3.0003
  -2.0452 + 1.4449i
  -2.0452 - 1.4449i
  -1.0290

>> G=F*L+(A21-L*C*A11)*inv(C)

G =
   -7.1115    2.9282
  -12.3865    1.9979
   30.4858    1.7323
   60.3197    3.5311

>> H=B2-L*C*B1

H =
   -1.6824   -1.3924
   -0.6870    0.0098
    5.5490   -0.1921
   12.2132   -0.0806
```

3.4.4 *Linear, Quadratic, Gaussian (LQG) Compensator*

For a multivariable plant, it is often advantageous to employ linear optimal control (Appendix A) for designing both the regulator and the observer. The resulting design is called a linear, quadratic, Gaussian (LQG) compensator, and is known for its good robustness properties. The plant linearized about the nominal trajectory is described by:

$$\dot{\mathbf{x}} = \mathsf{A}\mathbf{x} + \mathsf{B}\mathbf{u} + \mathsf{F}\mathbf{p}, \tag{3.129}$$

$$\mathbf{y} = \mathsf{C}\mathbf{x} + \mathsf{D}\mathbf{u} + \mathbf{w}, \tag{3.130}$$

where $\mathbf{y}(t)$ is the measured output, $\mathbf{p}(t)$ the process noise, and $\mathbf{w}(t)$ the measurement noise. A full-order observer designed by the steady state, linear optimal control[4] for minimizing a quadratic cost function of white process noise spectral density matrix, $\bar{\mathsf{P}}$, measurement noise spectral density matrix, W, and cross-spectral density matrix between process and measurement noise, Ψ, is called a *Kalman filter* [20] and has the following state equation:

$$\dot{\mathbf{x}}_o = (\mathsf{A} - \mathsf{LC})\,\mathbf{x}_o + \mathsf{B}\mathbf{u} + \mathsf{L}\mathbf{y}. \tag{3.131}$$

The steady state Kalman filter gain matrix, L, is given by:

$$\mathsf{L} = \left(\mathsf{V}\mathsf{C}^{\mathrm{T}} + \mathsf{F}\Psi\right)\mathsf{W}^{-1}, \tag{3.132}$$

where V is the *optimal covariance matrix* [20], the unique, symmetric and positive semidefinite solution to the following steady state, algebraic Riccati equation:

$$0 = \mathsf{A}_{\mathsf{G}}\mathsf{V} + \mathsf{V}\mathsf{A}_{\mathsf{G}}^{\mathrm{T}} - \mathsf{V}\mathsf{C}^{\mathrm{T}}\mathsf{W}^{-1}\mathsf{C}\mathsf{V} + \mathsf{F}\mathsf{P}_{\mathsf{G}}\mathsf{F}^{\mathrm{T}}, \tag{3.133}$$

and

$$\begin{aligned} \mathsf{A}_{\mathsf{G}} &= \mathsf{A} - \mathsf{F}\Psi\mathsf{W}^{-1}\mathsf{C} \\ \mathsf{P}_{\mathsf{G}} &= \bar{\mathsf{P}} - \Psi\mathsf{W}^{-1}\Psi^{\mathrm{T}}. \end{aligned} \tag{3.134}$$

Note the similarity between the Kalman filter and the LQRY regulator problem. When the process and measurement white noise are uncorrelated (as they usually are), we have $\Psi = 0$ and a considerable simplification takes place in the Kalman filter derivation. A sufficient condition for the existence of a unique, positive semidefinite solution to (3.133) is that the plant (A, C) is observable, P_{G} is positive semidefinite, and W is positive definite. Generally, the matrices $\bar{\mathsf{P}}, \mathsf{W}, \Psi$ are treated as design parameters for achieving a desired robustness through loop transfer recovery (LTR) [20]. Alternative design techniques such as $\mathcal{H}_\infty$-control and μ-synthesis [20] also achieve robustness objectives in a manner similar to the LQG/LTR method.

[4] The extension to a reduced-order observer is easily carried out; thus, there is no loss of generality.

3.5 Digital Control System

While we have assumed a continuous time (or *analog*) control system in the foregoing analysis, whenever a control system is implemented in practice, a time delay invariably occurs in processing the feedback signal and generating the corresponding control input by a controller. It is as if the output is held constant for an instant before being fed back to the plant in the form of the control input. Thus, we have a digitized (or *digital*) control system that works in discrete – rather than continuous – time steps. The process of handling continuous time signals in discrete time steps is called *analog-to-digital* (A/D) conversion, and consists of sampling and holding the data for a brief time interval. Every manned or automatic control system is inherently digital, eventhough the plant and controller themselves are analog systems. A modern digital computer with a fast processor and memory can quickly handle signals in a way that may appear to be continuous due to the very small time steps involved. On the other hand, a human operator who takes a few seconds to process a plant's output is very obviously a digital controller. The only difference between the two is the sampling rate.

The sampling rate at which an analog signal is processed is crucial for the successful implementation of a digital control system. The sampling rate must not be smaller than the bandwidth (i.e., the range of frequencies in the power spectrum) of the analog signal, otherwise a distorted digital output will be produced. In order to reduce signal distortion, the sampling rate should be at least twice the bandwidth of the analog signal. One half of the sampling rate is known as the *Nyquist frequency*, and is the approximate bandwidth of the digital system. Hence, we may regard the A/D converter as a *low-pass filter* [20] with the Nyquist frequency as the cut-off frequency.

When a digital controller sends a discrete time input to an analog plant, it must be first converted into a continuous signal by a *digital-to-analog* (D/A) convertor, which is the reverse of the sampling and hold process. An example of D/A convertor is playing a series of still photographs on a movie projector. By spinning the wheel of the projector, the time interval between any two shots is reduced to almost zero (compared with the time taken by the human brain to process individual pictures), thereby producing the sensation of continuous motion. The block-diagram of a digital control system is shown in Fig. 3.16.

Consider a first-order digital system with the following discrete time governing equation:

$$x_{k+1} = ax_k + bu_k, \tag{3.135}$$

where the subscript k denotes the state and input variables at the kth time step, while the subscript $k+1$ refers to the variables at the next time step, and a, b are constant coefficients. Clearly, (3.135) describes the evolution of the system in time (like the differential equation of a continuous time system) and is called a *difference equation*. If, instead of the subscripts, we express the system's future state by an

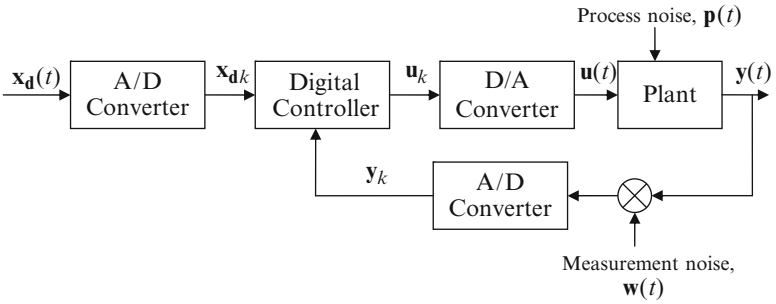


Fig. 3.16 A typical digital control system

operator, $\mathcal{Z}(.)$, that operates on the present state such that $\mathcal{Z}(x_{k+1}) = zx$, where z is a complex variable, the difference equation is said to be *z-transformed* as follows:

$$zx = ax + bu, \tag{3.136}$$

or

$$x = \frac{bu}{z - a}. \tag{3.137}$$

Clearly, the z-transform for a digital system is equivalent to Laplace transform for a continuous time system, and is useful in predicting the future state. A single-input, single-output (SISO) digital system is described by a *pulse transfer function*, $G(z)$, which is analogous to the transfer function of a continuous time system. Analog-to-digital conversion of a SISO system with transfer function $G(s)$ thus consists of taking a z-transform, $\mathcal{Z}(G(s)) = G(z)$. For more on z-transform and pulse transfer function refer to Chap. 8 of [20].

In order to derive difference equations of a multivariable plant, consider the LTI continuous system solution (Chap. 1) held for a sampling interval, $\Delta t = T$:

$$\mathbf{x}(t + T) = \mathrm{e}^{\mathsf{A}T}\mathbf{x}(t) + \int_t^{t+T} \mathrm{e}^{\mathsf{A}(t+T-\tau)}\mathsf{B}\mathbf{u}(\tau)\mathrm{d}\tau, \tag{3.138}$$

or, because $\mathbf{u}(t)$ is held during the sampling interval,

$$\mathbf{x}(t + T) = \mathrm{e}^{\mathsf{A}T}\mathbf{x}(t) + \mathrm{e}^{\mathsf{A}T}\left(\int_0^T \mathrm{e}^{-\mathsf{A}\tau}\mathrm{d}\tau\right)\mathsf{B}\mathbf{u}(t). \tag{3.139}$$

If one assumes that the interval T is small enough for the integrand, $\mathrm{e}^{-\mathsf{A}\tau}\mathrm{d}\tau$ to be constant during the sampling process, we have

$$\mathbf{x}(t + T) = \mathrm{e}^{\mathsf{A}T}\mathbf{x}(t) + T\mathsf{B}\mathbf{u}(t). \tag{3.140}$$

In any case, we can write the following difference equation:

$$\mathbf{x}_{k+1} = \bar{\mathsf{A}}\mathbf{x}_k + \bar{\mathsf{B}}\mathbf{u}_k, \tag{3.141}$$

with

$$\bar{\mathsf{A}} = \mathrm{e}^{\mathsf{A}T} \; ; \; \bar{\mathsf{B}} = \mathrm{e}^{\mathsf{A}T}\left(\int_0^T \mathrm{e}^{-\mathsf{A}\tau}\mathrm{d}\tau\right)\mathsf{B}. \tag{3.142}$$

The derivation of the digital state-space coefficient matrices from the analog ones is thus given by (3.142), which is also programmed in the inbuilt MATLAB-CST functions called *c2d* and *c2dm*. After conversion of the system into a digital form, one can apply techniques similar to those used for the continuous time systems for designing feedback regulators and observers [20].

Example 3.19. Derive the digital equivalent of the continuous time plant of Example 3.17 with $\mathsf{D} = 0$ using a zero-order hold and a sampling interval of $T = 0.1$ s.

We employ the inbuilt MATLAB-CST function *c2dm* as follows:

```
>> A=[0 -1 3 0 1 -2; -1 0.5 0 -1 0 1; 0 1 -1 2 -1 1;
      0 -2 -1 0 0.5 1; -1 0 0 0 1 0; 0 -1 0 0 0 1];
>> B=[-1  0;0 -0.5;1  0;0  -1;2  0;0  0.5];
>> C=[-1 0 0 0 0 0;0 0 0 0 0 1];D=[0 0;0 0];
>> [Ad,Bd,Cd,Dd] = c2dm(A,B,C,D,0.1,'zoh')

Ad =
    0.9996   -0.0799    0.2844    0.0333    0.0908   -0.1987
   -0.1026    1.0610   -0.0098   -0.1035   -0.0076    0.1130
    0.0008    0.0724    0.8957    0.1856   -0.0949    0.1139
    0.0077   -0.2150   -0.0944    1.0009    0.0578    0.0888
   -0.1052    0.0045   -0.0150   -0.0012    1.1003    0.0103
    0.0053   -0.1081    0.0003    0.0053    0.0003    1.0996


Bd =
   -0.0761   -0.0040
    0.0043   -0.0436
    0.0852   -0.0088
    0.0004   -0.0925
    0.2147    0.0001
   -0.0002    0.0550

Cd =
    -1     0     0     0     0     0
     0     0     0     0     0     1

Dd =
     0     0
     0     0
```

A digital Bode plot of the analog (solid line) and corresponding digital (circle) systems' transfer function from the first input to the first output – obtained as follows – is shown in Fig. 3.17:

```
>> bode(A,B(:,1),C(1,:),D(1,1),0.1) % Bode plot of analog system
>> hold on,dbode(Ad,Bd(:,1),Cd(1,:),Dd(1,1),0.1)%Bode plot of digital system
```

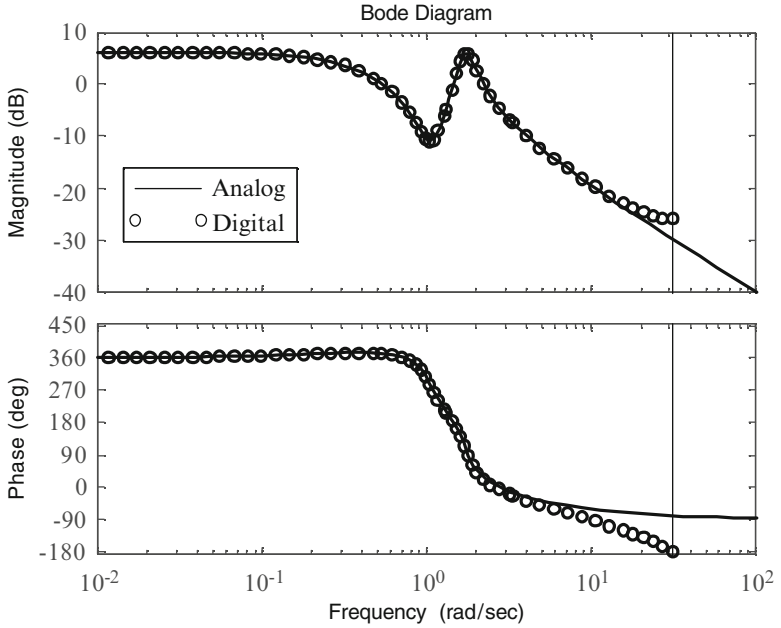


Fig. 3.17 Bode plots from the first input to the first output for analog (*solid line*) and digital (*circle*) systems

Note the vertical line at $\omega = \pi/T = 31.4159$ rad/s (or 5 Hz) indicating the cut-off (Nyquist) frequency. Thus, the digital system's response is limited to below 5 Hz. Furthermore, at the Nyquist frequency, the digital system's phase is shifted to $-180°$ from the $-90°$ for the analog system. On the other hand, the gain of the digital system at the Nyquist frequency is increased by about 5 dB compared with that of the analog system.

3.6 Summary

Flight control systems are often linear and can be either of single- or multivariable type, which are designed by the transfer function or state-space methods, respectively. While the step response can provide a measure of the single-input, single-output (SISO) system's performance, the frequency response reveals its stability and robustness. The practical SISO designs include the PID and feedforward/feedback tracking systems, which can be compensated for actuator dynamics by using lead or lag compensators. Multivariable regulators and observers can be designed separately by either eigenstructure assignment or the linear, optimal

(LQR) method that requires solving an algebraic Riccati equation. Linear observers designed by the linear, optimal control method are called Kalman filters, and have a guaranteed robustness for white noise disturbances. Digital control systems involve time discretization and require difference equations and z-transform for design and analysis.

Exercises

3.1. Derive an expression for the impulse response of a spring, mass, and damper system described by the differential equation,

$$m\ddot{y} + c\dot{y} + ky = u,$$

where $u(t)(=\delta(t))$ is the applied input and $y(t)(=g(t))$ the output, for underdamped ($0 \leq \zeta < 1$), critically damped, ($\zeta = 1$), and overdamped ($\zeta > 1$) cases.

3.2. Derive the ramp response for the system of Exercise 3.1 for underdamped ($0 \leq \zeta < 1$), critically damped, ($\zeta = 1$), and overdamped ($\zeta > 1$) cases.

3.3. A plant has the following transfer function:

$$G(s) = \frac{100}{s-4}.$$

(a) Derive an expression for the plant's step response. Is the plant stable?
(b) When the desired output is a unit step function, design a closed-loop controller for the plant such that the output has a settling-time of 1 second, damping ratio 0.7, and a zero steady-state error.
(c) Derive the expression for the closed-loop step response for the system designed in Part (b).
(d) Derive the expression for the closed-loop frequency response for the system designed in Part (b).
(e) Find a state-space representation for the plant.

3.4. Design a feedback compensator for the roll dynamics of a fighter aircraft described by the transfer function, $G(s) = 1{,}000/[s(s+5)]$, such that closed-loop step response settles down in 1 second with a zero steady-state error, and a damping ratio of 0.7.

3.5. For the aircraft longitudinal dynamics transfer function given in Example 3.9, find the approximate transfer function by neglecting the higher-frequency mode. Compare the frequency response and step response of the approximate transfer function with those given in the example.

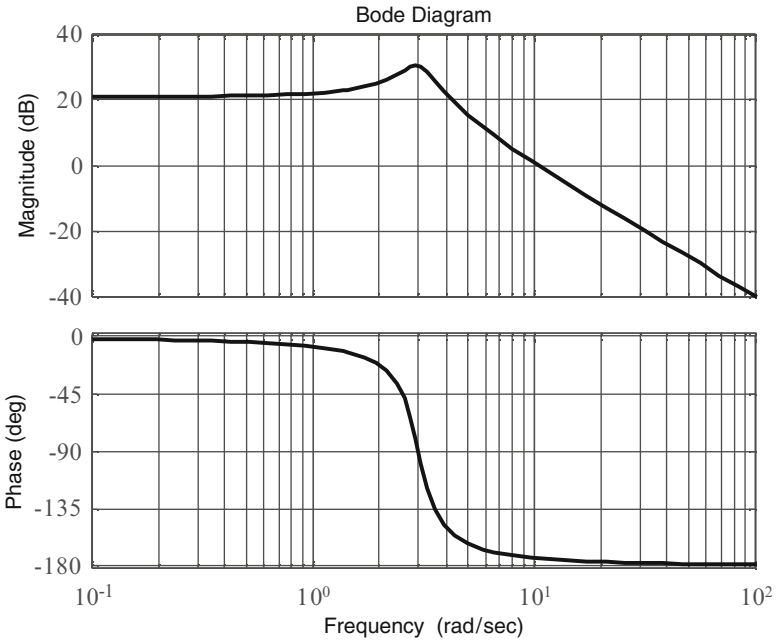


Fig. 3.18 Second-order, open-loop frequency response for the control system of Exercise 3.7

3.6. A closed-loop control system of Fig. 3.8 without any process or measurement noise has the following plant's transfer function:

$$G(s) = \frac{2s - 1}{s^2 + 2s - 3}.$$

(a) Can a compensator with transfer function, $H(s) = (s - 1)/(s + 1)$, stabilize this plant?
(b) If the compensator of Part (a) is used, what are the locations of the closed-loop poles?
(c) Can a proportional-integral-derivative (PID) compensator transfer function, $H(s)$, be selected such that the closed-loop step response settles down with a zero steady-state error? Why?

3.7. A closed-loop control system of Fig. 3.8 without any process or measurement noise has a second-order, open-loop frequency response plotted in Fig. 3.18. Determine the following:

(a) The open-loop transfer function.
(b) The gain and phase margins of the closed-loop system.

(c) The steady-state error of the closed-loop system to a unit step desired output, $Y_d(s)$.
(d) What modification do you propose in the control system in order to make the steady-state error in Part (c) zero?

3.8. Consider the control system shown in Fig. 3.8, where the plant's transfer function is $G(s) = 1/(s^2 + 0.3s + 0.02)$. It is intended to use a PID compensator for controlling this plant.

(a) What are the values of the PID compensator constants, k_1, k_2, k_3, for achieving a zero steady-state error with a closed-loop pole at $s = -1$, and a pair of complex conjugate poles with damping ratio 0.707, and natural frequency, 1 rad/s?
(b) Derive the closed-loop transfer function, $Y(s)/Y_d(s)$, with the compensator in part (a), and compute the closed-loop system step response. What is the maximum percentage overshoot, settling-time, and steady-state error of the closed-loop step response?
(c) Plot the locus of the closed-loop poles as the integral gain, k_3, varies from 0 to 10, with k_1, k_2 remaining constant at the values calculated in part(a). What are the values of k_3 for which the closed-loop system is stable?
(d) Draw a Bode plot of the closed-loop system of part (a), and determine the gain and phase margins, as well as the respective crossover frequencies.

3.9. Consider the following transfer function of an aircraft's longitudinal dynamics:

$$G(s) = -\frac{s^2 + 0.6s + 0.01}{s^4 + s^3 + s^2 + 0.01s + 0.009}.$$

(a) What are the poles of the system? Is the system stable?
(b) Using modal decomposition, resolve the denominator polynomial of $G(s)$ into two quadratic factors (modes), each corresponding to a pair of complex poles determined in Part (a). What are the respective natural frequencies and damping ratios of the two modes?
(c) Determine the step response by the MATLAB-CST command *step*.
(d) Make a Bode plot of the system using the MATLAB-CST command *bode*.

3.10. A system is described by the following transfer function:

$$G(s) = \frac{10(s+1)}{s^2(s-2)}.$$

(a) Is the system stable?
(b) Design a state feedback regulator for the system such that the closed-loop poles satisfy the characteristic equation,

$$s^3 + 2s^2 + 2s = -1.$$

3.11. Is it possible to design a tracking system for the plant given by Exercise 1.8 such that the desired state satisfies the following dynamic equation:

$$\dot{\mathbf{x}}_\mathbf{d} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 3 & -2 & -6 \end{pmatrix} \mathbf{x}_\mathbf{d}?$$

3.12. Consider an LTI system with the following state-space coefficient matrices:

$$\mathsf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \; ; \; \mathsf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}.$$

(a) Design a full-order observer for this plant using $\mathsf{C} = (1, 0, -1)$ with poles at $s_1 = -2$, $s_{2,3} = -2 \pm 2\mathrm{i}$.
(b) Based upon the direct measurement of any two state variables, design a reduced-order observer for this plant with pole at $s = -2$.
(c) Can a tracking system be designed for this plant to track a constant desired state of $\mathbf{x}_\mathbf{d} = (0, 1, 0)^\mathrm{T}$? Explain.

3.13. Design a linear, quadratic, regulator for a spring-mass system with the following governing equation:

$$\ddot{x} + 100x = u,$$

where $x(t)$ is the displacement from equilibrium in meters, and $u(t)$ is the applied input acceleration in $\mathrm{m/s}^2$. The optimal regulator must bring the displacement to zero with a settling-time of one second in the presence of an initial disturbance of $\dot{x}(0) = 1$ m/s.

3.14. For an aircraft with the following lateral dynamics:

$$\mathsf{A} = \begin{pmatrix} -15 & 0 & -15 & 0 & 0 \\ 0 & -0.8 & 10 & 0 & 0 \\ 0 & -1 & -0.8 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} ; \; \mathsf{B} = \begin{pmatrix} 25 & 3 \\ 0 & -3.5 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Design a bi-input linear, quadratic regulator with $\mathsf{Q} = \mathsf{I}_{5\times5}$ and $\mathsf{Q} = \mathsf{I}_{2\times2}$. Calculate and plot the initial response of the regulated system for the initial condition, $\mathbf{x}(0) = (0.5, 0, 0, 0, 0)^\mathrm{T}$. What are the settling-time and the maximum overshoot of the closed-loop initial response? What are the largest input magnitudes required for the closed-loop initial response?

3.15. Using any two state variables as the outputs, design a Kalman filter for the aircraft lateral dynamics of Exercise 3.14 with $\mathsf{F} = \mathsf{I}$, $\mathsf{W} = \mathsf{C}\mathsf{C}^\mathrm{T}$, $\Psi = 0$, and $\bar{\mathsf{P}} = 0.001\mathsf{I}$. What are the poles of the Kalman filter?

Chapter 4
Automatic Control of Aircraft

4.1 Aims and Objectives

- To derive linear aircraft flight dynamics plant for use in automatic flight control system design
- To decouple aircraft dynamics into longitudinal and lateral-directional plants, and to briefly study the stability and control derivatives
- To consider modeling of servo actuators for aircraft's engine and control surfaces
- To present single- and multi-variable automatic flight control systems (stability augmentation systems and autopilots) for both longitudinal and lateral-directional dynamics
- To design and analyze aircraft control systems using both transfer function and state-space methods

4.2 Aircraft Dynamics

Aircraft must operate at much lower altitudes when compared to other flight vehicles because they require adequate atmospheric density for generating lift and airbreathing thrust. Furthermore, in order to have a relatively small drag in the denser region of the atmosphere, aircraft flight speed is restricted to only a fraction of the orbital speed. Therefore, flight-path curvature and Coriolis acceleration terms are generally insignificant in the translational equations of motion. Hence, aircraft's translational kinematics – being of a much larger time scale than either rotational or translational kinetics – does not have too much impact on its dynamic behavior. Consequently, one can separate translational kinematics (to be used for long-range navigation purposes) from the much faster kinetic processes, and thus approximate the local horizon frame as an inertial reference frame for stability and control applications.

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_4, © Springer Science+Business Media, LLC 2011

4.2.1 Rotational Kinematics

Since aircraft flight typically takes place in a manner, that angular attitude about one principal body axis is confined to less than $\pm 90°$, its orientation can be described using Euler angles. As noted above, the local horizon frame is taken to be the inertial reference frame, with the axes, $\mathbf{I} = \mathbf{i}_\delta$ (North), $\mathbf{J} = \mathbf{i}_\lambda$ (East), and $\mathbf{K} = -\mathbf{i}_\mathbf{r}$ (Down).

The orientation of an aircraft's body-fixed frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, relative to the local horizon (inertial) frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, is represented by the Euler angle sequence, $(\Psi)_3, (\Theta)_2, (\Phi)_1$, (Chap. 2) as follows:

$$\begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix} = \mathsf{C} \begin{Bmatrix} \mathbf{I} \\ \mathbf{J} \\ \mathbf{K} \end{Bmatrix}, \tag{4.1}$$

where C is the following rotation matrix (Chap. 2):

$$\mathsf{C} = \begin{pmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ \sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi & \sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi & \sin\Phi\cos\Theta \\ \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi & \cos\Phi\cos\Theta \end{pmatrix}.$$

Here Φ is the *roll* (or *bank*) angle, Θ the *pitch* angle, and Ψ the *yaw* angle. The angular velocity of the rigid aircraft about its center of mass is that of the body-fixed frame relative to the reference frame, and is given by

$$\boldsymbol{\omega} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}, \tag{4.2}$$

where P is the *roll rate*, Q the *pitch rate*, and R the *yaw rate*. By the Euler angle representation, $(\Psi)_3, (\Theta)_2, (\Phi)_1$, shown in Fig. 4.1, we have the following rotational kinematics equation:

$$\boldsymbol{\omega}(t) = \dot{\Phi}\mathbf{i} + \dot{\Theta}\mathbf{J}' + \dot{\Psi}\mathbf{K}. \tag{4.3}$$

From the individual elementary rotations, (Ψ, Θ, Φ), constituting (4.1), we have

$$\boldsymbol{\omega}(t) = \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} = \begin{Bmatrix} \dot{\Phi} - \dot{\Psi}\sin\Theta \\ \dot{\Theta}\cos\Phi + \dot{\Psi}\sin\Phi\cos\Theta \\ -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Phi\cos\Theta \end{Bmatrix}. \tag{4.4}$$

For determining the Euler angles from the body rates, (P, Q, R), it is more appropriate to invert the relationship of (4.4) to yield the following rotational kinematics equations:

$$\begin{Bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{Bmatrix} = \frac{1}{\cos\Theta} \begin{pmatrix} \cos\Theta & \sin\Phi\sin\Theta & \cos\Phi\sin\Theta \\ 0 & \cos\Phi\cos\Theta & -\sin\Phi\cos\Theta \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix}. \tag{4.5}$$

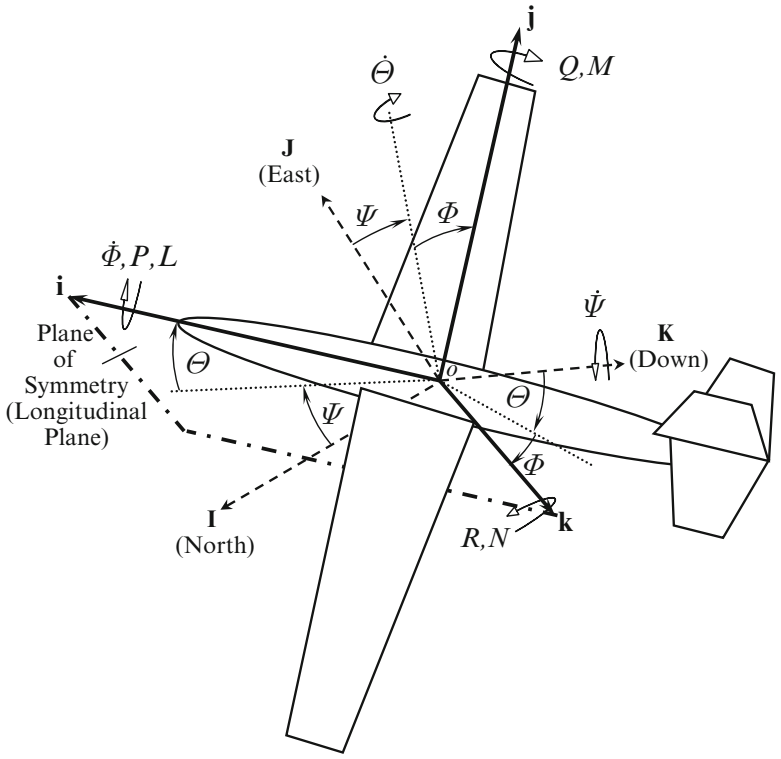


Fig. 4.1 Plane of symmetry (the longitudinal plane) and the longitudinal motion, ($\Phi = P = R = V = 0$)

From (4.5), it is clear that the Euler angle representation $(\Psi)_3, (\Theta)_2, (\Phi)_1$ is singular (or undefined) for $\Theta = \pm \pi/2$. Fortunately, most aircraft normally do not approach the vertical attitude. Therefore, (4.5) can be applied to conventional aircraft stability and control applications. When an exception arises – such as a vertical loop performed by a high-performance aircraft – it is treated by an alternative, nonsingular, attitude representation, like the quaternion (Chap. 2).

4.2.2 Translational Kinetics

For an aircraft, the external force, $\mathbf{F}(t)$, is the sum of the gravity force, $m\mathbf{g}$, the aerodynamic force, $\mathbf{F_a}(t)$, and the propulsive force (thrust), $\mathbf{T}(t)$. The vector sum of aerodynamic force and thrust is expressed as follows[1] (Chap. 2):

$$\mathbf{F_a} + \mathbf{T} = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}. \tag{4.6}$$

[1]Since aircraft are almost always powered by airbreathing engines that produce thrust by aerodynamic means, the aerodynamic and propulsive forces are always clubbed together.

The gravity force resolved in the body axes, becomes

$$m\mathbf{g} = mg\mathbf{K} = mg\left(-\sin\Theta\mathbf{i} + \sin\Phi\cos\Theta\mathbf{j} + \cos\Phi\cos\Theta\mathbf{k}\right). \tag{4.7}$$

For aircraft stability and control, the inertial reference frame $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, is approximated by the local horizon frame, which can thus be regarded as being instantaneously fixed to the atmosphere. Thus, the velocity of the center of mass, $\mathbf{v}(t)$, is taken to be the aircraft's linear velocity relative to the atmosphere (called the *relative velocity* in Chap. 2), and can be resolved in the body-fixed frame as follows:

$$\mathbf{v} = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}. \tag{4.8}$$

Therefore, the translational kinetics equation can be expressed as

$$\mathbf{F} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = m\dot{\mathbf{v}} + m\left(\boldsymbol{\omega}\times\mathbf{v}\right), \tag{4.9}$$

or, collecting the individual components, we have

$$\begin{aligned} X - mg\sin\Theta &= m(\dot{U} + QW - RV)\\ Y + mg\sin\Phi\cos\Theta &= m(\dot{V} + RU - PW)\\ Z + mg\cos\Phi\cos\Theta &= m(\dot{W} + PV - QU). \end{aligned} \tag{4.10}$$

Equations (4.10) are useful in stability analysis as well as in autopilot design. The acceleration terms on the right-hand side of (4.10) should be compared with those derived in Chap. 2 for a planet in fixed frame that rotates with the planet. Note that planetary rotation and curvature terms are neglected in deriving (4.10). Moreover, the mass m is taken to be a constant in the small time scale for stability and control analysis, rather than the varying mass model of the navigational plant.

4.2.3 Rotational Kinetics

All aircraft are designed to operate in equilibrium with a zero net external torque. However, when perturbed from equilibrium, the vehicle experiences a non-zero aerodynamic torque due to flow perturbations. A perturbing torque is also generated by aerodynamic control surfaces, and in some cases, by thrust deflection (Chap. 2). If we club all aerodynamic and propulsive torques into the external torque vector, $\boldsymbol{\tau}_\mathbf{a}$, we have

$$\boldsymbol{\tau}_\mathbf{a} = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}, \tag{4.11}$$

where L is the *rolling moment*, M the *pitching moment*, and N the *yawing moment* (Fig. 4.1). In order to have a zero aerodynamic torque in an equilibrium flight condition, all aircraft are designed with a plane of symmetry, oxz, denoted by the unit vector, $(\mathbf{i}, \mathbf{k})$ (Fig. 4.1).

The inertial coupling between pitch, roll, yaw, and the three translatory degrees of freedom (U, V, W) is evident in the translational and rotational dynamics of a general flight vehicle (Chap. 2). Such a coupling is further strengthened by aerodynamic forcing terms on the left-hand side of the equations of motion. However, the presence of the plane of symmetry for an aircraft and the assumption of small disturbances causes a separation of aerodynamic effects into those created by *longitudinal* motion (i.e., motion in the plane of symmetry) and *lateral-directional* motion (i.e., that occurring outside the plane of symmetry).

Due to the plane of symmetry, we have $J_{xy} = 0 = J_{yz}$, hence the rotational kinetics equation (Chap. 2) of an aircraft becomes

$$\boldsymbol{\tau}_{\mathbf{a}} = \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = \mathsf{J}\dot{\boldsymbol{\omega}} + \mathsf{S}(\boldsymbol{\omega})\mathsf{J}\boldsymbol{\omega}, \tag{4.12}$$

where

$$\mathsf{S}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{pmatrix}, \tag{4.13}$$

and

$$\mathsf{J} = \begin{pmatrix} J_{xx} & 0 & J_{xz} \\ 0 & J_{yy} & 0 \\ J_{xz} & 0 & J_{zz} \end{pmatrix}. \tag{4.14}$$

Equations (4.12)–(4.14) can be expressed as the following scalar equations of rotational kinetics:

$$L = J_{xx}\dot{P} + J_{xz}(\dot{R} + PQ) + \left(J_{zz} - J_{yy}\right) QR, \tag{4.15}$$

$$M = J_{yy}\dot{Q} + J_{xz}(R^2 - P^2) + (J_{xx} - J_{zz})\, PR, \tag{4.16}$$

$$N = J_{zz}\dot{R} + J_{xz}(\dot{P} - QR) + \left(J_{yy} - J_{xx}\right) PQ. \tag{4.17}$$

Here we have approximated the aircraft by a single rigid body by neglecting gyroscopic terms due to the fixed angular momenta of engines.[2] Equations (4.15)–(4.17) are nonlinear, coupled ordinary differential equations that must be solved

[2]Early fighter aircraft of World War-I era – like the British *Sopwith Camel* – were quite small and light-weight in construction, but had relatively big rotary engines. Hence, such aircraft encountered a significant coupling between the pitching motion (rotation about **j**), and the yawing motion (rotation about **k**) caused by the constant angular momentum of the engine. Thus, a pitching moment applied by the pilot invariably produced an unintended yaw (and roll), making the aircraft very difficult to control. It is believed that many crashes during fighting (extreme maneuvering) were caused by this inertial coupling. However, engine angular momentum is negligible for a conventional modern aircraft.

along with the translational kinetics equations, (4.10), and the rotational kinematics equations, (4.5), due to the dependence of (L, M, N) on (U, V, W) (and their time derivatives), which in turn, depend upon (Ψ, Θ, Φ). Furthermore, we require aerodynamic relationships governing (L, M, N, X, Y, Z) in terms of the motion variables, $(U, V, W, \dot{U}, \dot{V}, \dot{W}, \Psi, \Theta, \Phi, P, Q, R)$ in order to solve the combined set of translational and rotational dynamics equations.

4.3 Longitudinal Stability and Control

There exists a special solution to the aircraft dynamics equations, (4.5), (4.10), (4.15)–(4.17), for which $\Phi = P = R = V = 0$, yielding

$$\begin{aligned} \dot{\Theta} &= Q \\ \dot{\Psi} &= 0 \\ X - mg\sin\Theta &= m(\dot{U} + QW) \\ Y &= 0 \\ Z + mg\cos\Theta &= m(\dot{W} - QU) \\ L &= 0 \\ M &= J_{yy}\dot{Q} \\ N &= 0. \end{aligned} \tag{4.18}$$

Such a motion – called the *longitudinal dynamics* – is confined to the plane of symmetry and can be entirely represented by only four motion variables, $\Theta(t), U(t), W(t), Q(t)$, with $\Psi = \text{const}$. For this very reason, the plane of symmetry, oxz, is called the *longitudinal plane* (Fig. 4.1).

For aircraft stability and control analysis, the longitudinal motion is conveniently represented by a small displacement from an equilibrium flight condition in the longitudinal plane, denoted with a subscript e, $U_e, W_e, \Theta_e, Q_e, X_e, Z_e, M_e$. The most common aircraft equilibrium state is a straight line flight path for which $Q_e = 0$ and (4.18) yields

$$\begin{aligned} \dot{\Theta}_e &= 0 \\ X_e - mg\sin\Theta_e &= 0 \\ Z_e + mg\cos\Theta_e &= 0 \\ M_e &= 0. \end{aligned} \tag{4.19}$$

When the nominal trajectory is a steady maneuver, the aerodynamic force and moment are balanced by other forces, and thus have constant equilibrium values.

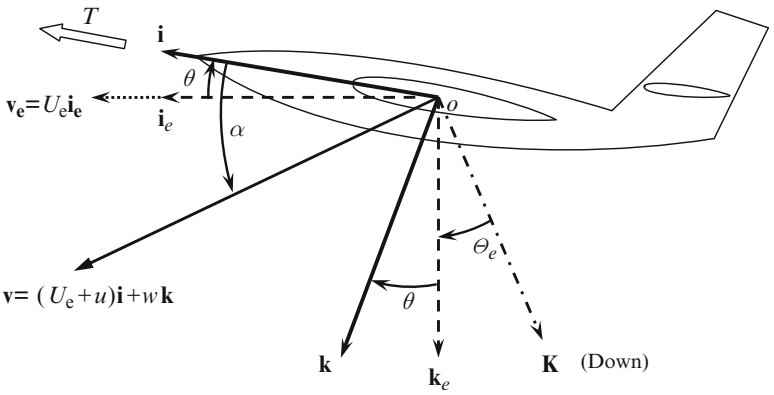


Fig. 4.2 The stability axes, $(\mathbf{i}_e, \mathbf{k}_e)$, for the longitudinal dynamics

In such a case, a special body-fixed frame – called *stability axes* – is generally employed having one axis initially aligned with the nominal flight direction (Chap. 2). Let $(\mathbf{i}_e, \mathbf{j}_e, \mathbf{k}_e)$, be the stability axes for a straight line equilibrium, such that the axis $\mathbf{i}_e$ is along the equilibrium flight direction, i.e., $W_e = 0$. Hence, a longitudinal perturbation can be described by the displaced body-fixed frame, $(\mathbf{i}, \mathbf{k})$ (Fig. 4.2), as well as perturbations in aerodynamic forces and moment given by:

$$\begin{aligned} \Theta &= \Theta_e + \theta \\ U &= U_e + u \\ W &= w \\ Q &= q \\ X &= X_e + \bar{X} \\ Z &= Z_e + \bar{Z} \\ M &= \bar{M}. \end{aligned} \tag{4.20}$$

The vertical velocity perturbation, w (called *downwash*), is represented as a perturbation in the *angle-of-attack* (Fig. 4.2) given by

$$\alpha = \tan^{-1} \frac{w}{U}. \tag{4.21}$$

The most practical method of analyzing stability of the aircraft in a particular equilibrium flight condition is to consider a small perturbation from equilibrium. By stability in the sense of Lyapunov (Chap. 1), if the perturbed motion tends to remain arbitrarily close to equilibrium, we say that the given equilibrium point is *stable*; if otherwise, the given equilibrium condition is termed *unstable*. The study of stability

is carried out for the *controls fixed* case, i.e., when no control inputs are applied despite the initial disturbance from equilibrium. On the other hand, controllability (Chap. 1) requires an arbitrary change in the motion variables by the application of control inputs. Such changes are caused by the variation of forces and moments due to the control inputs.

From the above given definition of stability, it is clear that the perturbation quantities, $\theta, u, w, q, \bar{X}, \bar{Z}, \bar{M}$, can be considered small to begin with. If the resulting equations of motion are such that the small perturbation quantities tend to grow with time, we will have an unstable flight equilibrium. However, in such a case, one need not consider large perturbations and merely the tendency for small perturbations to grow with time is sufficient to indicate instability. Thus, for all intents and purposes, a small perturbation assumption can be applied for carrying out a stability analysis. This is fortunate because aerodynamic forces and moments have a rather complex character in case of large perturbations, which can be very difficult to analyze in a closed form.

In case of a small perturbation from the equilibrium we have

$$\sin\Theta \simeq \sin\Theta_\mathrm{e} + \theta\cos\Theta_\mathrm{e},$$
$$\cos\Theta \simeq \cos\Theta_\mathrm{e} - \theta\sin\Theta_\mathrm{e},$$
$$u \ll U_\mathrm{e}; \qquad w \ll U_\mathrm{e}; \qquad \alpha \simeq \tan^{-1}\frac{w}{U_\mathrm{e}} \simeq \frac{w}{U_\mathrm{e}},$$

and the rotational kinematics equation (4.5) becomes

$$\dot{\theta} = q. \tag{4.22}$$

Since $U = U_\mathrm{e} + u \simeq U_\mathrm{e}$, we will henceforth drop the subscript on U_e.

With a small perturbation, the longitudinal aerodynamic forces and moments can be expressed as linear functions of the disturbed quantities. On expanding the aerodynamic forces and moments in a Taylor series about the equilibrium point and assuming small disturbances, we have the following linear approximations:

$$\begin{aligned}
X &= X^\mathrm{e} + X_u u + X_\alpha\alpha + X_{\dot{u}}\dot{u} + X_{\dot{\alpha}}\dot{\alpha} + X_q q + X_\delta\delta_\mathrm{E} + X_T\beta_\mathrm{T}\\
Z &= Z^\mathrm{e} + Z_u u + Z_\alpha\alpha + Z_{\dot{u}}\dot{u} + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_\delta\delta_\mathrm{E} + Z_T\beta_\mathrm{T}\\
M &= M^\mathrm{e} + M_u u + M_\alpha\alpha + M_{\dot{u}}\dot{u} + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_\delta\delta_\mathrm{E} + M_T\beta_\mathrm{T},
\end{aligned} \tag{4.23}$$

Here, the coefficients

$$X_u = \left(\frac{\partial\bar{X}}{\partial u}\right)_\mathrm{e}, \text{ etc.}$$

are due to the multivariable Taylor series expansion truncated to first-order. The coefficients related to the motion variables, $u, \alpha, \dot{u}, \dot{\alpha}, q,$ are called *stability*

derivatives as they influence the stability of the equilibrium point, while those related to the control inputs, δ_{E} (elevator deflection) and β_{T} (throttle deflection), are termed *control derivatives*.

4.3.1 Longitudinal Stability Derivatives

The stability of aircraft motion depends upon the magnitudes and signs of the stability derivatives. Stability derivatives are functions of the aircraft's geometry, size, airspeed, flow regime (i.e., nondimensional parameters describing the relative flow, such as equilibrium angle-of-attack, sideslip, Mach number, and Reynolds number) (Chap. 2), and atmospheric properties. These functional relationships can either be estimated by semi-empirical methods [7], or carefully obtained from wind-tunnel tests.

A useful method of analyzing stability derivatives is by making them nondimensional with respect to size, mass, atmospheric density and flight speed. In order to do so, we define the *wing planform area*, S, as the reference area, and $\bar{c}$ the wing's *mean aerodynamic chord* (MAC), as the characteristic length (Fig. 4.3):

$$S = \int_{-b/2}^{b/2} c(y)\mathrm{d}y; \qquad \bar{c} = \frac{1}{S}\int_{-b/2}^{b/2} c^2(y)\mathrm{d}y, \tag{4.24}$$

where b is the *wing span*, $c(y)$ the chord distribution, and y is measured along $\mathbf{j}$ from the plane of symmetry ($y = 0$) as shown in Fig. 4.3. The longitudinal stability derivatives are rendered nondimensional by dividing the forces by $\bar{q}S$, moments by $\bar{q}S\bar{c}$, and speed by U, where $\bar{q} = 1/2\rho U^2$ is the *dynamic pressure* of the free stream. The angular rates are traditionally expressed in a nondimensional time, $\hat{t} = t\frac{2U}{\bar{c}}$, which corresponds to the dimensional angular rates being multiplied by the factor, $\frac{\bar{c}}{2U}$.

The nondimensional forward and vertical force coefficients, C_x and C_z, respectively, can be obtained by nondimensionalizing the respective forces according to the standard NACA nomenclature:

$$C_x = \frac{X}{\bar{q}S} = \frac{X_{\mathrm{e}} + \bar{X}}{\bar{q}S}, \tag{4.25}$$

and

$$C_z = \frac{Z}{\bar{q}S} = \frac{Z_{\mathrm{e}} + \bar{Z}}{\bar{q}S}. \tag{4.26}$$

The aerodynamic force is resolved into components along the wind axes, namely the lift, $\mathcal{L}$ and the drag, D. The lift acts normal to the instantaneous velocity vector, $\mathbf{v} = (U_{\mathrm{e}} + u)\mathbf{i} + w\mathbf{k}$ (Fig. 4.2), while the drag is always opposed to $\mathbf{v}$. Since the engines are firmly bolted to the aircraft, the thrust, T, is along the perturbed body

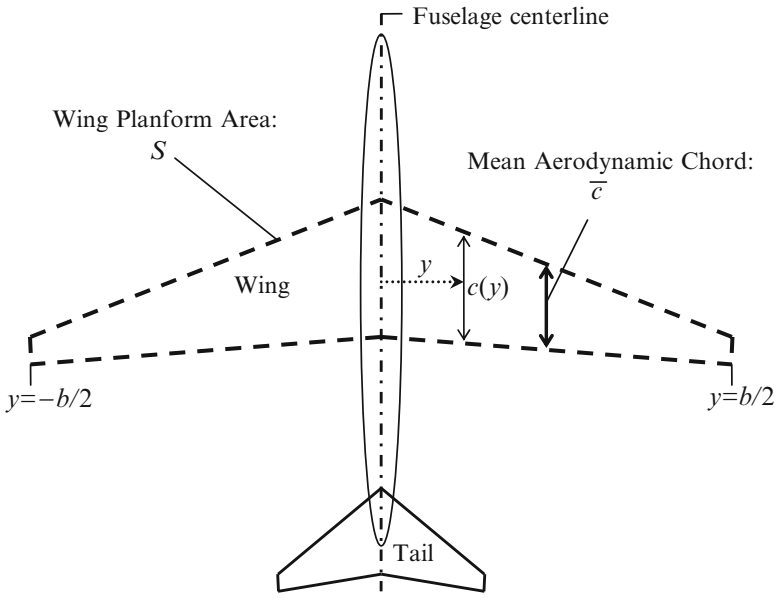


Fig. 4.3 The planform (*top view*) geometry, with wing planform area, S, tail planform area, S_t, and wing mean aerodynamic chord (MAC), $\bar{c}$

axis, **i**, as shown in Fig. 4.2. The perturbed velocity vector, **v**, makes an angle α with ox, which is the change in the angle-of-attack. Thus, with a small angle-of-attack perturbation, α, we have

$$\begin{aligned} C_x &\simeq C_{\mathrm{T}} - C_{\mathrm{D}} + C_{\mathrm{L}}\alpha \\ C_z &\simeq -C_{\mathrm{L}} - C_{\mathrm{D}}\alpha, \end{aligned} \tag{4.27}$$

where

$$C_{\mathrm{L}} = \frac{\mathcal{L}}{\bar{q}S} \tag{4.28}$$

is the lift coefficient,

$$C_{\mathrm{T}} = \frac{T}{\bar{q}S} \tag{4.29}$$

the thrust coefficient, and

$$C_{\mathrm{D}} = \frac{D}{\bar{q}S} \tag{4.30}$$

the drag coefficient. For a small angle-of-attack perturbation, α, the lift coefficient usually has a linear relationship with the angle-of-attack, given by

$$C_{\mathrm{L}} = C_{\mathrm{Le}} + C_{\mathrm{L}\alpha}\alpha, \tag{4.31}$$

where $C_{L\alpha}$ is the *lift-curve slope* and C_{Le} the lift coefficient at equilibrium. For a steady and straight-line equilibrium condition, we require

$$\begin{aligned} C_{Le} &= \frac{mg\cos\Theta_e}{\bar{q}S} \\ C_{Te} - C_{De} &= \frac{mg\sin\Theta_e}{\bar{q}S}. \end{aligned} \tag{4.32}$$

The drag coefficient is generally related to the lift coefficient by a *parabolic drag polar*, and given by

$$C_D = C_{D0} + KC_L^2, \tag{4.33}$$

where C_{D0} is the parasite drag coefficient and K the lift-dependent drag factor. The aerodynamic coefficients $C_{L\alpha}, C_{D0}, K$ depend upon the aircraft's geometrical shape as well as on the flow regime parameters (Mach number and Reynolds number) at equilibrium. Unless the compressibility effects are predominant (such as in the transonic flow regime), $C_{L\alpha}, C_{D0}, K$ are taken to be constants.

Derivatives of forward and vertical forces with airspeed can be expressed in terms of the nondimensional quantities as follows:

$$\begin{aligned} \frac{\partial X}{\partial u} &= \rho U S C_{xe} + \bar{q}S\frac{\partial C_x}{\partial \hat{u}} \\ \frac{\partial Z}{\partial u} &= \rho U S C_{ze} + \bar{q}S\frac{\partial C_z}{\partial \hat{u}}, \end{aligned} \tag{4.34}$$

where

$$\hat{u} = \frac{u}{U} \tag{4.35}$$

is the nondimensional airspeed and C_{xe}, C_{ze} are the nondimensional force coefficients in the equilibrium condition. Since $C_{xe} = C_{Le}\tan\Theta_e$ and $C_{ze} = -C_{Le}$ by (4.27) and (4.32), we write

$$\begin{aligned} C_{xu} &= \frac{\partial C_x}{\partial \hat{u}} = -2C_{Le}\tan\Theta_e + \frac{U}{\bar{q}S}\frac{\partial X}{\partial u} \\ C_{zu} &= \frac{\partial C_z}{\partial \hat{u}} = 2C_{Le} + \frac{U}{\bar{q}S}\frac{\partial Z}{\partial u}. \end{aligned} \tag{4.36}$$

Substituting in the first two equations of (4.23) into (4.25) and (4.26) with the nondimensional derivatives is defined above, we have

$$C_x = (C_{xu} + 2C_{Le}\tan\Theta_e)\,\hat{u} + C_{x\alpha}\alpha + C_{x\dot{\alpha}}\hat{\dot{\alpha}} + C_{xq}\hat{q}, \tag{4.37}$$

and

$$C_z = (C_{zu} - 2C_{Le})\,\hat{u} + C_{z\alpha}\alpha + C_{z\dot{\alpha}}\hat{\dot{\alpha}} + C_{zq}\hat{q}. \tag{4.38}$$

The force derivatives that do not involve differentiation with respect to speed, u, are simpler to nondimensionalize:

$$\begin{aligned}
C_{x\alpha} &= \frac{1}{\bar{q}S}\frac{\partial \bar{X}}{\partial \alpha} = \frac{\partial C_x}{\partial \alpha} \\
C_{x\dot{\alpha}} &= \frac{1}{\bar{q}S}\frac{2U}{\bar{c}}\frac{\partial \bar{X}}{\partial \dot{\alpha}} = \frac{\partial C_x}{\partial \hat{\dot{\alpha}}} \\
C_{xq} &= \frac{1}{\bar{q}S}\frac{2U}{\bar{c}}\frac{\partial \bar{X}}{\partial q} = \frac{\partial C_x}{\partial \hat{q}} \\
C_{z\alpha} &= \frac{1}{\bar{q}S}\frac{\partial \bar{Z}}{\partial \alpha} = \frac{\partial C_z}{\partial \alpha} \\
C_{z\dot{\alpha}} &= \frac{1}{\bar{q}S}\frac{2U}{\bar{c}}\frac{\partial \bar{Z}}{\partial \dot{\alpha}} = \frac{\partial C_z}{\partial \hat{\dot{\alpha}}} \\
C_{zq} &= \frac{1}{\bar{q}S}\frac{2U}{\bar{c}}\frac{\partial \bar{Z}}{\partial q} = \frac{\partial C_z}{\partial \hat{q}},
\end{aligned} \tag{4.39}$$

where

$$\hat{\dot{\alpha}} = \dot{\alpha}\left(\frac{\bar{c}}{2U}\right), \tag{4.40}$$

and

$$\hat{q} = q\left(\frac{\bar{c}}{2U}\right). \tag{4.41}$$

Similarly, the nondimensional pitching moment coefficient, C_{m}, can be derived from (4.23) as follows:

$$C_{\mathrm{m}} = \frac{M}{\bar{q}S\bar{c}} = C_{mu}\hat{u} + C_{m\alpha}\alpha + C_{m\dot{\alpha}}\hat{\dot{\alpha}} + C_{mq}\hat{q}, \tag{4.42}$$

where

$$\begin{aligned}
C_{mu} &= \frac{\partial C_{\mathrm{m}}}{\partial \hat{u}} = \frac{U}{\bar{q}S\bar{c}}\frac{\partial M}{\partial u} - 2C_{\mathrm{me}} = \frac{U}{\bar{q}S\bar{c}}\frac{\partial M}{\partial u} \\
C_{m\alpha} &= \frac{1}{\bar{q}S\bar{c}}\frac{\partial M}{\partial \alpha} = \frac{\partial C_{\mathrm{m}}}{\partial \alpha} \\
C_{m\dot{\alpha}} &= \frac{1}{\bar{q}S\bar{c}}\frac{2U}{\bar{c}}\frac{\partial M}{\partial \dot{\alpha}} = \frac{\partial C_{\mathrm{m}}}{\partial \hat{\dot{\alpha}}} \\
C_{mq} &= \frac{1}{\bar{q}S\bar{c}}\frac{2U}{\bar{c}}\frac{\partial M}{\partial q} = \frac{\partial C_{\mathrm{m}}}{\partial \hat{q}},
\end{aligned} \tag{4.43}$$

as $C_{\mathrm{me}} = 0$ for equilibrium.

4.3.1.1 Speed Derivatives

The first-order changes in aerodynamic forces and moments are due to a change in the forward speed, u, when other perturbations are zeros (i.e., $\alpha = \dot{\alpha} = q = 0$) are called the *speed derivatives*. The change in speed causes a change of Mach number, that influences the aerodynamic forces and moments by compressibility effect. In addition, there is a change in the thrust caused by the forward speed. The speed derivative of the axial force can be obtained as follows:

$$\begin{aligned} C_{xu} &= \frac{\partial C_x}{\partial \hat{u}} = \frac{\partial C_{\mathrm{T}}}{\partial \hat{u}} - \frac{\partial C_{\mathrm{D}}}{\partial \hat{u}} \\ &= \frac{U}{\bar{q}S}\left(\frac{\partial T}{\partial u}\right)_{\mathrm{e}} - \left(\frac{2T}{\bar{q}S}\frac{\partial U}{\partial u}\right)_{\mathrm{e}} - U\frac{\partial C_{\mathrm{D}}}{\partial \mathcal{M}}\frac{\partial \mathcal{M}}{\partial u} \\ &= \frac{U}{\bar{q}S}\left(\frac{\partial T}{\partial u}\right)_{\mathrm{e}} - 2C_{T\mathrm{e}} - \mathcal{M}\frac{\partial C_{\mathrm{D}}}{\partial \mathcal{M}}, \end{aligned} \tag{4.44}$$

where

$$\mathcal{M} = \frac{U}{a}$$

is the flight Mach number at equilibrium. Clearly, the thrust derivative with forward speed,

$$\left(\frac{\partial T}{\partial u}\right)_{\mathrm{e}},$$

is crucial in determining C_{xu}. For equilibrium flight condition, we have from (4.19)

$$C_{T\mathrm{e}} = C_{D\mathrm{e}} + C_{\mathrm{Le}} \tan \Theta_{\mathrm{e}}.$$

For a glider, $T = 0$, $C_{xu} = -\mathcal{M}\partial C_{\mathrm{D}}/\partial \mathcal{M}$. For an airplane powered by jet or rocket engines, the thrust is invariant with flight speed, thus

$$C_{xu} = -2\left(C_{D\mathrm{e}} + C_{\mathrm{Le}} \tan \Theta_{\mathrm{e}}\right) - \mathcal{M}\frac{\partial C_{\mathrm{D}}}{\partial \mathcal{M}}.$$

A propeller driven airplane has a constant thrust power, TU, thus

$$\left(\frac{\partial T}{\partial u}\right)_{\mathrm{e}} = -\frac{T}{U},$$

and

$$C_{xu} = -3\left(C_{D\mathrm{e}} + C_{\mathrm{Le}} \tan \Theta_{\mathrm{e}}\right) - \mathcal{M}\frac{\partial C_{\mathrm{D}}}{\partial \mathcal{M}}.$$

Similarly, the normal force speed derivative can be expressed as follows:

$$\begin{aligned} C_{zu} &= \frac{\partial C_z}{\partial \hat{u}} = -\frac{\partial C_{\mathrm{L}}}{\partial \hat{u}} \\ &= -U\left(\frac{\partial C_{\mathrm{L}}}{\partial u}\right)_{\mathrm{e}} \\ &= -\mathcal{M}\frac{\partial C_{\mathrm{L}}}{\partial \mathcal{M}}. \end{aligned} \tag{4.45}$$

The speed derivative, C_{mu}, is affected by both compressibility (i.e., $\mathcal{M}$) and static aeroelastic deformations (usually fuselage bending) that change the angle-of-attack at the tail. While the change in lift and drag by aeroelastic effect is usually negligible, the pitching moment can be appreciably affected by even a small change in the tail lift due to a large tail arm. Since the aeroelatic deformations are directly proportional to the flight dynamic pressure, $\bar{q} = 1/2\rho U^2$, we must take into account the change in the pitching moment caused by a modified dynamic pressure due to speed changes. Therefore, we write

$$C_{mu} = \frac{\partial C_{\mathrm{m}}}{\partial \hat{u}} = \mathcal{M}\frac{\partial C_{\mathrm{m}}}{\partial \mathcal{M}} + 2\bar{q}\frac{\partial C_{\mathrm{m}}}{\partial \bar{q}}. \tag{4.46}$$

The main compressibility effect on C_{mu} is an aft movement of the average center of pressure of the wing as the speed increases from subsonic to supersonic. Thus, the transonic speed range is responsible for the largest negative magnitude of C_{mu}. Approximate values of $\partial C_{\mathrm{m}}/\partial \mathcal{M}$ can be obtained from a potential flow aerodynamic theory. The aeroelastic effect is much more difficult to predict, requiring either a detailed mathematical modeling, or a wind-tunnel test of an aeroelastically scaled model.

4.3.1.2 Static Stability Derivatives

The stability derivatives due to a change in the angle-of-attack, $C_{x\alpha}, C_{z\alpha}, C_{\mathrm{m}\alpha}$, denote static aerodynamic phenomena and are thus called static (or aerodynamic stiffness) stability derivatives. The stability derivative $C_{\mathrm{m}\alpha}$ determines the longitudinal static stability of an aircraft.

By differentiating (4.27) with respect to α and putting $\alpha = 0$ for the equilibrium condition we have

$$\begin{aligned} C_{x\alpha} &= \left(\frac{\partial C_x}{\partial \alpha}\right)_{\mathrm{e}} = C_{\mathrm{Le}} - \left(\frac{\partial C_{\mathrm{D}}}{\partial \alpha}\right)_{\mathrm{e}} \\ &= C_{\mathrm{Le}} + 2KC_{\mathrm{L}\alpha}C_{\mathrm{Le}} = C_{\mathrm{Le}}\left(1 + 2KC_{\mathrm{L}\alpha}\right), \end{aligned} \tag{4.47}$$

$$C_{z\alpha} = -C_{\mathrm{L}\alpha} - C_{D\mathrm{e}} = -C_{\mathrm{L}\alpha} - C_{\mathrm{D0}} - KC_{\mathrm{Le}}^2, \tag{4.48}$$

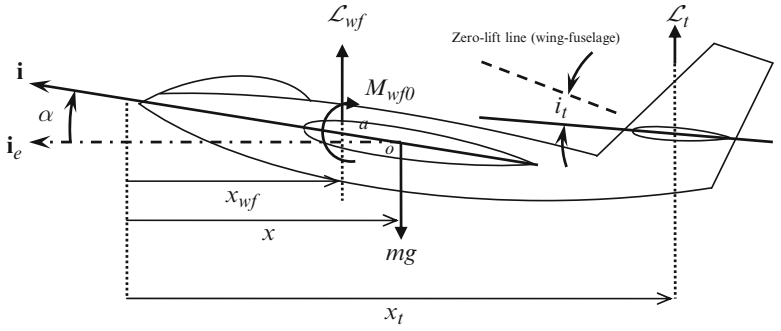


Fig. 4.4 The center of mass, o, the wing–fuselage aerodynamic center, a, and the tail setting angle, i_t, measured from the zero-lift line of wing–fuselage combination

where the thrust is considered invariant with the angle-of-attack (a good approximation for all engines with small α) and the parabolic drag polar of (4.33) is assumed.

The determination of the static stability derivative, $C_{m\alpha}$, involves the individual contributions of the wing, the fuselage, and the horizontal tail. The major influence on the pitching moment is that of the lifting surfaces (the wing and horizontal tail) as they have the largest normal force magnitudes, C_z, while the fuselage has a minor effect of re-distributing the aerodynamic pressure on the two lifting surfaces. Each lifting surface is associated with a unique point about which the pitching moment caused by the lifting surface is invariant with the angle-of-attack of the given surface. Such a point is called the *aerodynamic center*. Furthermore, each lifting surface has a unique relative flow direction for which it experiences a zero lift. This relative flow direction is called the *zero-lift line*. Consider an airplane with the wing–fuselage aerodynamic center location, x_{wf}, the center of mass location, x, and the horizontal tail's aerodynamic center, x_{t}, measured from a common datum as shown in Fig. 4.4. We reiterate that since $u = 0$ here, the angle α denotes the angle-of-attack perturbation from initial straight line equilibrium, and is thus the angle made by the perturbed longitudinal axis, $\mathbf{i}$, with the equilibrium flight direction, $\mathbf{i_e}$ (Fig. 4.4).

At equilibrium, we have $\alpha = 0$ and $C_{\mathrm{m}} = 0$. Let us nondimensionalize the various distances by dividing by the mean aerodynamic chord of the wing, $\bar{c}$,

$$\bar{x} = \frac{x}{\bar{c}}; \qquad \bar{x}_{\mathrm{wf}} = \frac{x_{\mathrm{wf}}}{\bar{c}}; \qquad \bar{x}_t = \frac{x_{\mathrm{t}}}{\bar{c}},$$

and write the nondimensional pitching moment equation in the perturbed condition as follows:

$$C_{\mathrm{m}} = C_{\mathrm{m}\alpha}\alpha = C_{\mathrm{mwf0}} + (\bar{x} - \bar{x}_{\mathrm{wf}})\, C_{\mathrm{Lwf}} - \eta \frac{S_{\mathrm{t}}}{S} (\bar{x}_t - \bar{x})\, C_{\mathrm{Lt}}. \tag{4.49}$$

Here

$$C_{\mathrm{Lwf}} = \frac{\mathcal{L}_{\mathrm{wf}}}{\bar{q}S}, \tag{4.50}$$

$$C_{\mathrm{Lt}} = \frac{\mathcal{L}_t}{\bar{q}'S_{\mathrm{t}}}, \tag{4.51}$$

C_{mwf0} is the pitching moment coefficient of the wing-body combination when $C_{\mathrm{Lwf}} = 0$,[3] S_{t} is the planform area of the horizontal tail, and $\bar{q}'$ the effective dynamic pressure seen by the tail, reduced due to the wing and fuselage from the freestream (flight) dynamic pressure by the following *tail efficiency factor*:

$$\eta = \frac{\bar{q}'}{\bar{q}} < 1.$$

The total airplane lift coefficient is given by

$$C_{\mathrm{L}} = C_{\mathrm{Lwf}} + \eta \frac{S_{\mathrm{t}}}{S} C_{\mathrm{Lt}}. \tag{4.52}$$

Before proceeding further, it is important to recognize that the tail is usually set at a negative angle with respect to the zero-lift line of wing–fuselage combination, called *tail setting angle*, i_{t}, as depicted in Fig. 4.4. This is because the tail must produce a negative lift for achieving moment equilibrium when $C_{\mathrm{Lwf}} = 0$. Because of the negative zero-lift pitching moment, $C_{\mathrm{mwf0}} < 0$, of a positively cambered wing airfoil. Furthermore, whenever there is an increase in the angle-of-attack, α, there is also an increase in the lift-induced *downwash angle*, ϵ, generated by the wing–fuselage combination at the tail, which is assumed to vary linearly with α as follows:

$$\epsilon = \frac{\partial \epsilon}{\partial \alpha}\alpha. \tag{4.53}$$

Thus, the change in the tail's angle-of-attack, α_t, due to a change in the wing–fuselage angle-of-attack, α is given by

$$\alpha_t = \alpha - \epsilon = \left(1 - \frac{\partial \epsilon}{\partial \alpha}\right)\alpha. \tag{4.54}$$

Since the tail usually has a symmetrical airfoil, the tail's lift coefficient is given by

$$C_{\mathrm{Lt}} = C_{\mathrm{Lte}} + C_{\mathrm{L}\alpha\mathrm{t}}\alpha_t = C_{\mathrm{Lte}} + C_{\mathrm{L}\alpha\mathrm{t}}\left(1 - \frac{\partial \epsilon}{\partial \alpha}\right)\alpha, \tag{4.55}$$

[3] Generally, $C_{\mathrm{mwf0}} < 0$ due to a positively cambered wing airfoil that is necessary for producing $C_{\mathrm{Lwf}} > 0$ when the wing–fuselage angle-of-attack is zero. In contrast, the tail airfoil is symmetrical to produce both positive and negative tail lift for stability and control.

substituting into (4.31) and (4.52) produces

$$C_{\mathrm{L}} = C_{\mathrm{Le}} + C_{\mathrm{L}\alpha\mathrm{wf}}\alpha + \eta\frac{S_{\mathrm{t}}}{S}C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right)\alpha, \tag{4.56}$$

where the equilibrium lift coefficient ($\alpha = 0$) is

$$C_{\mathrm{Le}} = C_{\mathrm{Lwfe}} + \eta\frac{S_{\mathrm{t}}}{S}C_{\mathrm{Lt_e}}. \tag{4.57}$$

On comparing (4.56) and (4.31), we have the *lift-curve slope* of the airplane

$$C_{\mathrm{L}\alpha} = C_{\mathrm{L}\alpha\mathrm{wf}} + \eta\frac{S_{\mathrm{t}}}{S}C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right), \tag{4.58}$$

where $C_{\mathrm{L}\alpha\mathrm{wf}}$ and $C_{\mathrm{L}\alpha\mathrm{t}}$ are the lift-curve slopes of the wing–fuselage and the tail, respectively.

The pitching moment equation can be alternatively written as follows by substituting (4.55) into (4.49):

$$\begin{aligned} C_{\mathrm{m}} = {} & C_{\mathrm{mwf0}} + (\bar{x}-\bar{x}_{\mathrm{wf}})\left(C_{\mathrm{Lwfe}} + C_{\mathrm{L}\alpha\mathrm{wf}}\alpha\right) \\ & - \eta\frac{S_{\mathrm{t}}}{S}(\bar{x}_t-\bar{x})\left[C_{\mathrm{Lte}} + C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right)\alpha\right], \end{aligned} \tag{4.59}$$

or by (4.49), we have

$$0 = C_{\mathrm{mwf0}} + (\bar{x}-\bar{x}_{\mathrm{wf}})\,C_{\mathrm{Lwfe}} - \eta\frac{S_{\mathrm{t}}}{S}(\bar{x}_t-\bar{x})\,C_{\mathrm{Lte}}, \tag{4.60}$$

and

$$\begin{aligned} C_{\mathrm{m}\alpha} &= (\bar{x}-\bar{x}_{\mathrm{wf}})\,C_{\mathrm{L}\alpha\mathrm{wf}} - \eta\frac{S_{\mathrm{t}}}{S}(\bar{x}_t-\bar{x})\,C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right) \\ &= (\bar{x}-\bar{x}_{\mathrm{wf}})\left[C_{\mathrm{L}\alpha\mathrm{wf}} + \eta\frac{S_{\mathrm{t}}}{S}C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right)\right] - \eta V_{\mathrm{t}}C_{\mathrm{L}\alpha\mathrm{t}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right), \end{aligned} \tag{4.61}$$

where

$$V_{\mathrm{t}} = \frac{S_{\mathrm{t}}}{S}(\bar{x}_t-\bar{x}_{\mathrm{wf}}), \tag{4.62}$$

is the *tail volume ratio*. Note that V_{t} only depends upon the fixed *tail arm*, $(\bar{x}_t-\bar{x}_{\mathrm{wf}})$, and is independent of the location of the center of mass, $\bar{x}$, that is a variable of loading and fuel distribution of the airplane. Substituting (4.58) into (4.61), we have

$$C_{\mathrm{m}\alpha} = C_{\mathrm{L}\alpha}\left[\bar{x}-\bar{x}_{\mathrm{wf}} - \eta V_{\mathrm{t}}\frac{C_{\mathrm{L}\alpha\mathrm{t}}}{C_{\mathrm{L}\alpha}}\left(1-\frac{\partial\epsilon}{\partial\alpha}\right)\right], \tag{4.63}$$

or,

$$C_{m\alpha} = C_{L\alpha}\left(\bar{x} - \bar{x}_n\right), \tag{4.64}$$

where

$$\bar{x}_n = \bar{x}_{wf} + \eta V_t \frac{C_{L\alpha t}}{C_{L\alpha}}\left(1 - \frac{\partial \epsilon}{\partial \alpha}\right) \tag{4.65}$$

is the *controls fixed neutral point*.[4] Clearly, the controls fixed neutral point is the location of the center of mass for which $C_{m\alpha} = 0$, therefore the neutral point is to the whole airplane what the aerodynamic center is for lifting surface. For static longitudinal stability, one must have

$$C_{m\alpha} < 0,$$

which implies $\bar{x} < \bar{x}_{wf}$, i.e., the center of mass must be forward of the wing–fuselage aerodynamic center. Due to its influence on static longitudinal stability, the nondimensional distance between the neutral point and the center of mass, $(\bar{x}_n - \bar{x})$, is called the *controls fixed static margin* and is commonly expressed as percentage of the mean aerodynamic chord.

By eliminating C_{Lwfe} from (4.57) and (4.60), we can derive the following expression for the equilibrium lift coefficient:

$$C_{Le} = \frac{C_{mwf0} - \eta V_t C_{Lte}}{\bar{x}_{wf} - \bar{x}}. \tag{4.66}$$

Example 4.1. An airplane of mass 10,000 kg, with a wing of planform area 50 m^2, wing's mean aerodynamic chord 3 m, and tail planform area 10 m^2 is flying straight having a standard sea level with a constant speed of 500 km/h. A wind-tunnel test carried out on a scaled model of the airplane reveals the lift-curve slopes of the wing–fuselage combination and the tail to be 0.1 per deg. and 0.08 per deg., respectively. The test also indicates that the tail efficiency factor is 0.96, the zero-lift pitching moment coefficient of the wing–fuselage combination is −0.09, the airplane's center of mass is located 10% of the wing's mean aerodynamic chord aft of the wing–fuselage aerodynamic center, and the derivative of downwash angle with angle-of-attack is 0.4. The distance between the aerodynamic centers of the wing and the tail is 8.5 m. Calculate:

(a) The equilibrium tail lift coefficient
(b) The equilibrium lift coefficient of wing–fuselage combination
(c) The controls fixed static margin
(d) Static stability derivative, $C_{m\alpha}$

[4] Often, $\bar{x}_n$ is called the *stick fixed neutral point* referring to a cockpit control lever named *stick*, manipulated by the pilot for longitudinal control.

We begin by computing the equilibrium lift coefficient as follows:

$$\bar{q} = \frac{1}{2}(1.225)\left(\frac{500 \times 1{,}000}{3{,}600}\right)^2 = 11815.2 \text{ N/m}^2$$

$$C_{\text{Le}} = \frac{10{,}000 \times 9.81}{11815.2 \times 50} = 0.166057.$$

Next, we calculate the equilibrium lift coefficient of the tail by (4.66) as follows:

$$V_{\text{t}} = \frac{10 \times 8.5}{50 \times 3} = 0.566667$$

$$C_{\text{Lte}} = \frac{C_{\text{mwf0}} + C_{\text{Le}}(\bar{x} - \bar{x}_{\text{wf}})}{\eta V_{\text{t}}}$$

$$= -\frac{0.09 + (0.166057)(0.1)}{(0.96)(0.566667)} = -0.19597.$$

The equilibrium lift coefficient of the wing–fuselage combination is thus calculated by following (4.57):

$$C_{\text{Lwfe}} = C_{\text{Le}} - \eta\frac{S_{\text{t}}}{S}C_{\text{Lt}} = 0.166057 - (0.96)\left(\frac{10}{50}\right)(-0.19597) = 0.20368.$$

The controls fixed static margin calculation is performed using (4.58) and (4.63) as follows:

$$C_{\text{L}\alpha} = 0.1 + (0.96)\left(\frac{10}{50}\right)(0.08)(1 - 0.4) = 0.10922 \text{ per deg.}$$

$$\bar{x}_n - \bar{x} = -0.1 + (0.96)(0.566667)\left(\frac{0.08}{0.10922}\right)(1 - 0.4) = 0.1391.$$

Thus the controls fixed static margin is 13.91% of the mean aerodynamic chord. Finally, the static stability derivative is computed to be the following:

$$C_{\text{m}\alpha} = -(0.10922)(0.1391) = -0.01519 \text{ per deg.}$$

4.3.1.3 Pitch Derivatives

The derivatives due to a change in the pitch rate, C_{zq} and C_{mq}, when there is no change in either the speed or the angle-of-attack, are called pitch derivatives.[5] They

[5] Since neither the thrust nor the drag is much affected by a small pitch rate, by differentiating the first of (4.27) with q and putting $\alpha = 0$, we have $C_{xq} = 0$.

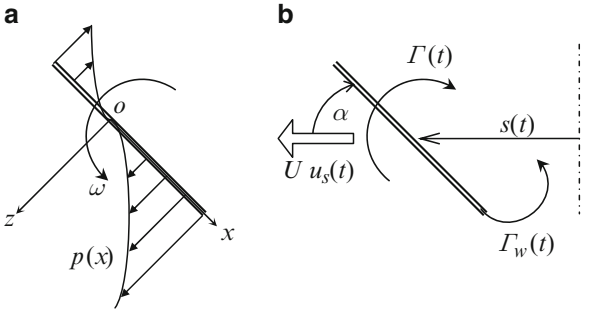


Fig. 4.5 Flat plate in incompressible flow undergoing (**a**) a steady rotation at rate ω about o, and (**b**) sudden acceleration at $t = 0$ from rest with a constant angle, α, to a speed, U

arise due to a flight path curvature in the longitudinal plane, which causes a change in the effective angle-of-attack of the wing and the horizontal tail. The derivative C_{mq} determines the change in the pitching moment due to pitch rate, and is termed *damping in pitch*.

Example 4.2. In order to understand the basic aerodynamics of changes due to rotation at a steady rate, consider a flat plate of chord c and an infinite span rotating at a constant rate, ω, about pitch axis passing through point, o, in an incompressible, inviscid medium (Fig. 4.5a). The resulting two-dimensional flow has the following governing equations in terms of velocity components, $u(x, z)$, $w(x, z)$, density, ρ, and pressure, $p(x, z)$:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0, \tag{4.67}$$

$$\rho\left(u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial p}{\partial x}, \tag{4.68}$$

$$\rho\left(u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial p}{\partial z}, \tag{4.69}$$

subject to the boundary condition, $w(x, 0) = \omega x$. From symmetry in the flowfield, we see that the equations have the following solution:

$$u(x, z) = \omega z \tag{4.70}$$

$$w(x, z) = \omega x \tag{4.71}$$

$$p(x, z) = -\frac{1}{2}\rho\omega^2(x^2 + z^2). \tag{4.72}$$

When integrated over the chord, we have the following net normal force and pitching moment per unit span:

$$Z/b = \int_{-a}^{c-a} p(x,0)\mathrm{d}x = -\frac{1}{6}\rho\omega^2\left[(c-a)^3 + a^3\right], \tag{4.73}$$

$$M/b = \int_{-a}^{c-a} xp(x,0)\mathrm{d}x = -\frac{1}{8}\rho\omega^2\left[(c-a)^4 - a^4\right], \tag{4.74}$$

where a is the distance of the rotation axis, o, from the leading edge. Clearly, for $a = c/2$ the normal force vanishes, but the pitching moment remains non-zero. Both pitching moment and normal force have their maximum magnitudes for rotation about the leading edge ($a = 0$). This simple flow model with pure rotation (zero freestream velocity) illustrates the dependence of Z and M on ω^2. However, for a flat plate rotating at a small rate in comparison with the velocity of translation (like an aircraft wing), these relationships are linear in ω. When real flow effects of viscosity and compressibility are added, the pressure distribution shown in Fig. 4.5a is greatly modified.

Due to a long tail arm, $\ell = x_\mathrm{t} - x$, a positive change in the pitch rate causes an appreciable increase in the effective angle-of-attack at the tail due to flow rotation, given by

$$\alpha' = \frac{q\ell}{U}, \tag{4.75}$$

thereby changing the tail lift coefficient by

$$C'_{\mathrm{Lt}} = C_{\mathrm{L}\alpha\mathrm{t}}\frac{q\ell}{U}, \tag{4.76}$$

the total lift coefficient by

$$C'_L = \eta\frac{S_\mathrm{t}}{S}C_{\mathrm{L}\alpha\mathrm{t}}\frac{q\ell}{U}, \tag{4.77}$$

and the total pitching moment coefficient by

$$C'_\mathrm{m} = -\eta V_\mathrm{t} C_{\mathrm{L}\alpha\mathrm{t}}\frac{q\ell}{U}. \tag{4.78}$$

The tail contributions effectively employ a quasi-steady flow approximation, wherein the unsteady flow over the tail, and that is due to wake vortices shed by the wing are neglected. This is a good approximation provided the tail arm is at least three or four times larger than the mean aerodynamic chord of the wing, thereby allowing sufficient time for the wing wake to develop to a steady state as it reaches the tail.

The contribution of the wing to pitch derivatives is much smaller and more difficult to estimate than that of the tail due to an unsteady change in the pressure

distribution of the wing caused by the pitching motion. A detailed wind-tunnel test may be the only procedure to accurately estimate the wing contribution. A swept-back wing has a larger contribution to pitch rate derivatives than a straight wing (due to a larger effective arm, $x - x_{wf}$). However, semi-empirical methods are commonly employed to modify the tail contribution by multiplying it by a constant factor, $\bar{K} > 1$, in order to account for the wing's contribution to the pitch derivatives.

The two pitch derivatives are estimated as follows:

$$C_{zq} = -\frac{\partial C_{\mathrm{L}}}{\partial \hat{q}} = -\bar{K}\frac{2U}{\bar{c}}\eta\frac{S_{\mathrm{t}}}{S}C_{\mathrm{L}\alpha\mathrm{t}}\frac{\ell}{U} \simeq -2\bar{K}V_{\mathrm{t}}C_{\mathrm{L}\alpha\mathrm{t}}$$

$$C_{\mathrm{m}q} = \frac{\partial C_{\mathrm{m}}}{\partial \hat{q}} = -\bar{K}\frac{2U}{\bar{c}}\eta V_{\mathrm{t}}C_{\mathrm{L}\alpha\mathrm{t}}\frac{\ell}{U} \simeq -2\bar{K}V_{\mathrm{t}}C_{\mathrm{L}\alpha\mathrm{t}}\frac{\ell}{\bar{c}},$$

where $\bar{K} \simeq 1.1$ for a straight wing, and $\bar{K} \simeq 1.3$ for a highly swept-back wing.

4.3.1.4 Aerodynamic Inertia Derivatives

The changes in aerodynamic force and moment do not occur instantaneously with the change in the angle-of-attack, but generally involve a time-lag due to both circulatory and noncirculatory flow over the lifting surfaces associated with the generation of lift. Unless the airplane's mass density is comparable to atmospheric density, the noncirculatory portion of lift – caused by accelerating a given mass of air vertically and called *apparent additional mass effect* – is negligible in comparison with the lift generated by circulation of air around the lifting surface. Hence, the aerodynamic time-lag, or *aerodynamic inertia*, is best represented by potential, unsteady aerodynamics of circulatory flow.

When a quasi-steady approximation is employed, we assume that all the aerodynamic changes are caused by a constant rate, $\dot{\alpha}$, which implies a plunging motion at a constant acceleration. In such a case, we have first-order coefficients in the form of $\dot{\alpha}$ stability derivatives.

Example 4.3. Consider a rectangular flat plate of chord c and infinite span inclined at a constant angle $\bar{\alpha}$, and suddenly accelerated forward from rest (Fig. 4.5b) at $t = 0$ in an incompressible medium to a constant speed, U. Such a motion corresponds to a step change in the angle-of-attack:

$$U(t) = U u_{\mathrm{s}}(t); \qquad \alpha(t) = \bar{\alpha} u_{\mathrm{s}}(t), \tag{4.79}$$

where $u_{\mathrm{s}}(t)$ is the unit step function applied at $t = 0$ (Chap. 1).

The unsteady lift per unit span of the plate has the following expression [9]:

$$\frac{\mathcal{L}(t)}{b} = \rho U \Gamma(t) u_{\mathrm{s}}(t), \tag{4.80}$$

where $\Gamma(t)$ is the circulation induced by a developing wake vortex of strength, $\Gamma_w(t)$ (Fig. 4.5b). The following is a simple analytical expression in terms of the nondimensional distance, $s(t) = Ut/c$, traveled forward by the plate from rest is due to Wagner [22]:

$$\Gamma(t) = \pi U c \left(1 - \frac{1}{2+s}\right) \sin\bar{\alpha}. \tag{4.81}$$

By substituting (4.81) into (4.80) we have the following unsteady lift per unit span:

$$\frac{\mathcal{L}(t)}{b} = \pi\rho U^2 c \left(1 - \frac{1}{2+s}\right) u_s(t) \sin\bar{\alpha}. \tag{4.82}$$

Thus, the lift per unit span starts off ($t = 0$) with half the steady state value of $\pi\rho U^2 c \sin\bar{\alpha}$, which is attained in the limit $t \to \infty$. The predicted non-zero lift at $t = 0$ is due to a noncirculatory flow acceleration.

Generally, $C_{x\dot{\alpha}} \simeq 0$, as thrust and drag are, by and large, noncirculatory in nature. On the other hand, the lag in lift and pitching moment can be large for a conventional airplane equipped with a horizontal tail. In order to appreciate the derivation of aerodynamic inertia derivatives due to flow circulation, consider an airplane whose angle-of-attack is changed abruptly at $t = 0$ from zero to a perturbed value, $\bar{\alpha}$, in a subsonic flow. Assuming a linear dependence of unsteady lift on the angle-of-attack, we write the following first-order differential equation for the lift coefficient:

$$\dot{C}_L(t) + a_2 C_L(t) = a_1 \alpha(t), \tag{4.83}$$

where a_1, a_2 are constants that depends on airplane geometry and Mach number. Taking a Laplace transform of (4.83) subject to initial condition, $C_L(0) = C_{Le}$, and a step change in angle-of-attack at $t = 0$, i.e., $\alpha(s) = \bar{\alpha}/s$, we have

$$(s + a_2) C_L(s) - C_{Le} = \frac{a_1}{s}\bar{\alpha}, \tag{4.84}$$

or,

$$C_L(t) = \bar{\alpha}\frac{a_1}{a_2}\left[1 + \left(\frac{a_2}{a_1\bar{\alpha}} C_{Le} - 1\right) e^{-a_2 t}\right] u_s(t), \tag{4.85}$$

where $u_s(t)$ is the unit step function applied at $t = 0$. The transient lift coefficient can also be approximated as follows:

$$C_L(t) = C_{L\alpha}\bar{\alpha} u_s(t) + C_{L\dot{\alpha}}\bar{\alpha}\delta(t), \tag{4.86}$$

where $\delta(t) = du_s(t)/dt$ is the unit impulse (or Dirac delta) function (Chap. 1). A plot of $C_L(t)$ for $C_{Le} = 0.5$, $a_1 = 3/\text{s per deg.}$, $a_2 = 5/\text{s}$, and $\bar{\alpha} = 1°$ is shown

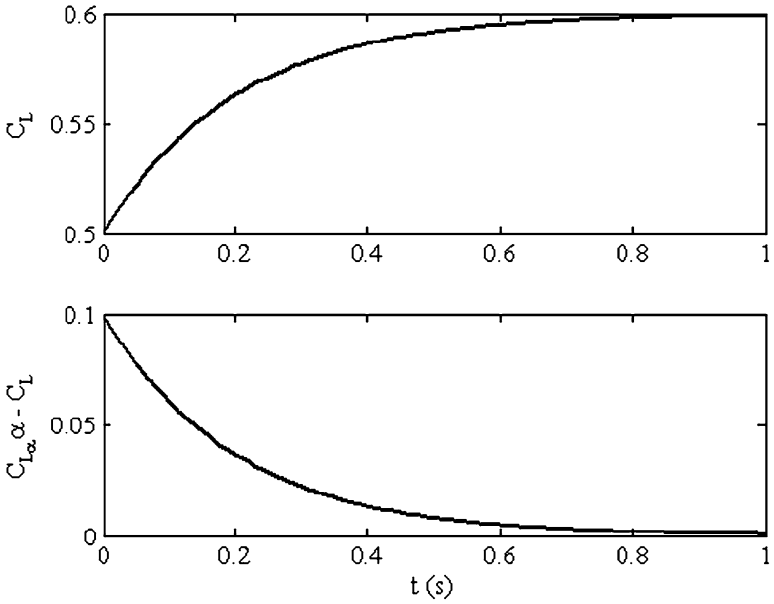


Fig. 4.6 The unsteady aerodynamic lift and lift deficiency due to a step change in the angle-of-attack at $t = 0$

in Fig. 4.6. Note that the step change in the angle-of-attack requires nearly 1 second for the lift to develop to its steady-state value of $C_{L\alpha}\bar{\alpha} = (a_1/a_2)(1) = 0.6$. The unsteady *lift deficiency*, $C_{L\alpha}\bar{\alpha} - C_L(t)$, persists until the steady-state is reached, and the total area under the lift deficiency curve can be used to estimate the derivative, $C_{z\dot{\alpha}}$, as follows:

$$C_{z\dot{\alpha}} = -\frac{2U}{\bar{c}}\frac{\partial C_L}{\partial \dot{\alpha}} = \frac{2U}{\bar{c}}\int_0^t \left[C_{L\alpha} - \frac{C_L(\tau)}{\bar{\alpha}}\right] d\tau. \tag{4.87}$$

The pitching moment derivative, $C_{m\dot{\alpha}}$, can be estimated using the quasi-steady correction of downwash at the tail caused by a rate of change of angle-of-attack, $\dot{\alpha}$,

$$C'_{Lt} = C_{L\alpha t}\dot{\alpha}\frac{\ell}{U}\frac{\partial \epsilon}{\partial \alpha}, \tag{4.88}$$

resulting in

$$C_{m\dot{\alpha}} = -\frac{2U}{\bar{c}}V_t C_{L\alpha t}\frac{\ell}{U}\frac{\partial \epsilon}{\partial \alpha} = -2V_t C_{L\alpha t}\frac{\ell}{\bar{c}}\frac{\partial \epsilon}{\partial \alpha}. \tag{4.89}$$

4.3.2 Longitudinal Modes

Substituting (4.19)–(4.23) into (4.18) and neglecting products of small perturbation quantities we have

$$\begin{aligned} m\dot{u} &= X_u u + X_\alpha \alpha + X_q q - mg\theta \cos \Theta_e \\ mU(\dot{\alpha} - q) &= Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}} \dot{\alpha} + Z_q q - mg\theta \sin \Theta_e \\ \dot{\theta} &= q \\ J_{yy}\dot{q} &= M_u u + M_\alpha \alpha + M_q q + M_{\dot{\alpha}} \dot{\alpha}, \end{aligned} \tag{4.90}$$

which can be expressed in a dimensional state space form as follows:

$$\begin{Bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{Bmatrix} = \mathsf{A}_{\text{Long}} \begin{Bmatrix} u \\ \alpha \\ \theta \\ q \end{Bmatrix}, \tag{4.91}$$

where

$$\mathsf{A}_{\text{Long}} = \begin{bmatrix} \frac{X_u}{m} & \frac{X_\alpha}{m} & -g\cos\Theta_e & 0 \\ \frac{Z_u}{mU - Z_{\dot{\alpha}}} & \frac{Z_\alpha}{mU - Z_{\dot{\alpha}}} & -\frac{mg\sin\Theta_e}{mU - Z_{\dot{\alpha}}} & \frac{mU + Z_q}{mU - Z_{\dot{\alpha}}} \\ 0 & 0 & 0 & 1 \\ \frac{M_u}{J_{yy}} + \frac{M_{\dot{\alpha}} Z_u}{J_{yy}(mU - Z_{\dot{\alpha}})} & \frac{M_\alpha}{J_{yy}} + \frac{M_{\dot{\alpha}} Z_\alpha}{J_{yy}(mU - Z_{\dot{\alpha}})} & -\frac{M_{\dot{\alpha}}(mg\sin\Theta_e)}{J_{yy}(mU - Z_{\dot{\alpha}})} & \frac{M_q}{J_{yy}} + \frac{M_{\dot{\alpha}}(mU + Z_q)}{J_{yy}(mU - Z_{\dot{\alpha}})} \end{bmatrix}$$

is the state dynamics matrix for longitudinal, small disturbance motion.

The longitudinal modes are determined from the quartic characteristic equation that can be factored into a pair of distinct quadratic modes:

$$|s\mathsf{I} - \mathsf{A}_{\text{Long}}| = \left(s^2 + 2\zeta_p \omega_p s + \omega_p^2\right)\left(s^2 + 2\zeta_s \omega_s s + \omega_s^2\right) = 0, \tag{4.92}$$

where (ω_p, ζ_p) are the natural frequency and damping ratio of the smaller frequency (or *long-period*[6]) mode, and (ω_s, ζ_s) are those of the higher frequency (or *short-period*) mode.

[6]The long-period mode is also called the *phugoid mode* in standard aeronautical terminology.

Example 4.4. Consider an airplane flying straight and level ($\Theta_e = 0$) with a constant speed, $U = 55$ m/s. The controls fixed longitudinal parameters of the airplane in this flight condition are as follows:

$$\frac{X_u}{m} = -0.045/\text{s}; \qquad \frac{Z_u}{m} = -0.36/\text{s}; \qquad \frac{X_\alpha}{m} = 1.96\ \text{m/s}^2$$

$$\frac{Z_\alpha}{m} = -108\ \text{m/s}^2; \qquad \frac{M_\alpha}{J_{yy}} = -8.6/\text{s}^2; \qquad \frac{M_{\dot{\alpha}}}{J_{yy}} = -0.9/\text{s}; \qquad \frac{M_q}{J_{yy}} = -2/\text{s}$$

while the stability derivatives, $Z_q, Z_{\dot{\alpha}}, M_u$, are negligible in comparison with the other terms.

The longitudinal state dynamics matrix of the airplane in the given equilibrium flight condition is calculated to be the following:

$$\mathsf{A}_{\text{Long}} = \begin{pmatrix} -0.045 & 1.96 & -9.81 & 0 \\ -0.0065 & -1.9636 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.0059 & -6.8327 & 0 & -2.9 \end{pmatrix}.$$

The longitudinal modes are computed using the MATLAB's *Control Systems Toolbox* (CST) command *damp* as follows:

```
>> A=[-0.045     1.96   -9.81      0
     -0.0065  -1.9636       0      1
           0        0       0      1
      0.0059  -6.8327       0   -2.9];

>> damp(A)

        Eigenvalue            Damping     Freq. (rad/s)
 -1.73e-002 + 2.09e-001i     8.26e-002      2.09e-001
 -1.73e-002 - 2.09e-001i     8.26e-002      2.09e-001
 -2.44e+000 + 2.57e+000i     6.88e-001      3.54e+000
 -2.44e+000 - 2.57e+000i     6.88e-001      3.54e+000
```

Thus, $\omega_p = 0.209$ rad/s, $\zeta_p = 0.0826$ for the phugoid (long period) mode and $\omega_s = 3.54$ rad/s, $\zeta_s = 0.688$ for the short-period mode of this particular airplane in the given straight and level flight condition. Note that the phugoid mode is lightly damped, while the short-period mode has a near critical damping ratio.

Since the phugoid and the short-period modes generally have widely different frequencies, there is little chance of interaction among the two, and thus each mode can be approximated by a distinct second order system. The approximate phugoid dynamics consists of a slow oscillation of airspeed, u, and pitch angle, θ, at a nearly constant angle-of-attack ($\alpha = 0$). The small damping of the phugoid mode is due to a small magnitude of the derivative, X_u, which is predominantly the change in

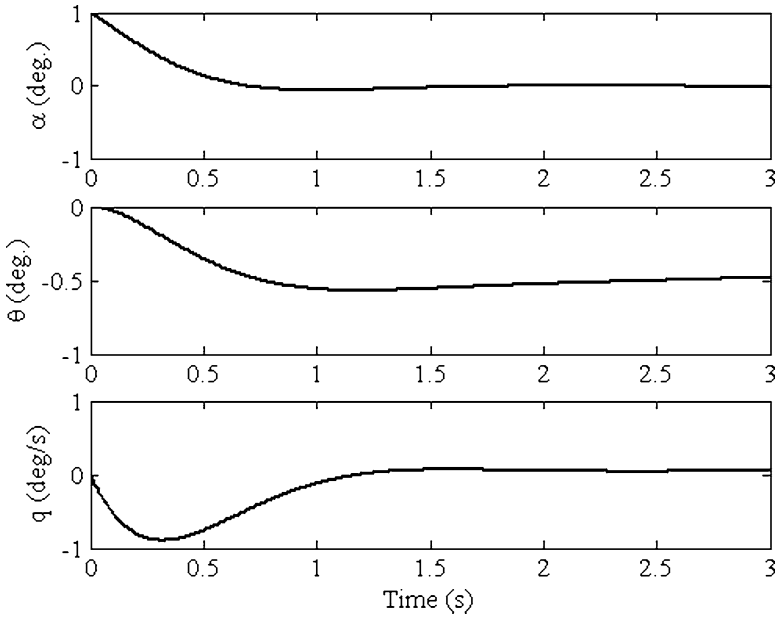


Fig. 4.7 The short-period longitudinal response of an airplane to an initial angle-of-attack disturbance of 1°

drag (itself a small force) due to variation in forward speed. Thus, the approximate phugoid dynamics can be obtained by neglecting the pitching moment equation and putting $\alpha = 0$ in the axial and normal force equations [21]. For a small equilibrium pitch angle, Θ_e, the phugoid natural frequency, ω_p, is directly influenced by the derivative, Z_u.

In contrast with the phugoid mode, the approximate short-period dynamics consists of a rapid, well damped oscillation in α and θ, but a negligible change in airspeed, u. Thus the short-period dynamics is approximated by neglecting the axial force equation and putting $u = 0$ in the remaining equations. The short-period damping is dominated by the derivatives Z_α and M_q.

Example 4.5. Consider a response of the airplane described in Example 4.4 to the initial condition, $u(0) = 1$ m/s, $\alpha(0) = 0$, $\theta(0) = 0$, $q(0) = 0$.

With the state dynamics matrix entered in Example 4.4, we use MATLAB's *Control Systems Toolbox* (CST) command *initial* as follows:

```
>> sys=ss(A,zeros(4,1),eye(4),zeros(4,1));
>> [y,t,x]=initial(sys,[0 pi/180 0 0]');
```

The initial response plotted for the first 3 s is shown in Fig. 4.7, and for 5–100 s is provided in Fig. 4.8. Note the short-period dynamics of well damped angle-of-attack and pitching response for the first 3 s, while the slightly damped, long-period dynamics of slow airspeed and pitching oscillation and a near constant

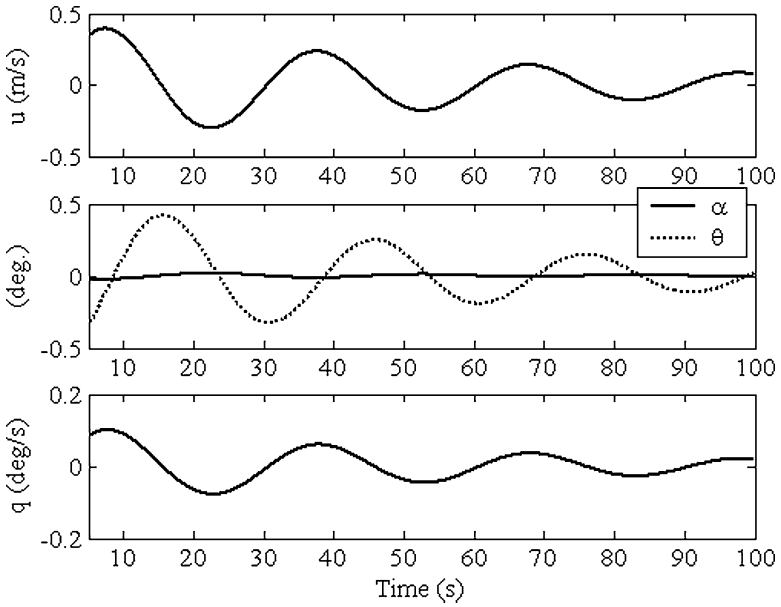


Fig. 4.8 The long-period (phugoid) longitudinal response of an airplane to an initial angle-of-attack disturbance of 1 °

angle-of-attack in Fig. 4.8. Clearly, the short-period transient decays in about 2 s, leaving the phugoid mode for a much longer time. This behavior exemplifies the lack of interaction between the two modes, thereby allowing a second-order approximation for each in their respective time scales.

4.3.2.1 Nondimensional Longitudinal Dynamics (Controls Fixed)

Since it is more useful to discuss longitudinal dynamics without bringing the size, mass, atmospheric density, and flight speed in question, we shall nondimensionalize the controls fixed longitudinal equations of motion, (4.91), according to the method presented in (4.25)–(4.43). Beginning with the forward speed equation, we write

$$\left(\frac{1}{U}\right)\left(\frac{\bar{c}}{2U}\right)\left[\dot{u} - \frac{X_u}{m}u - \frac{X_\alpha}{m}\alpha + g\theta\cos\Theta_e - \frac{X_q}{m}q\right] = 0, \tag{4.93}$$

or

$$\dot{\hat{u}} = \frac{(C_{xu} + 2C_{Le}\tan\Theta_e)}{\mu}\hat{u} + \frac{C_{x\alpha}}{\mu}\alpha - \frac{C_{Le}}{\mu}\theta + \frac{C_{xq}}{\mu}\hat{q}, \tag{4.94}$$

where the nondimensional derivatives are defined earlier, while the nondimensional mass is seen to be the following:

$$\mu = \frac{4m}{\rho S \bar{c}}. \tag{4.95}$$

Similarly, the nondimensional normal force and pitching moment equations are derived to be the following:

$$\dot{\hat{\alpha}} = \frac{(C_{zu} - 2C_{\mathrm{Le}})}{\mu - C_{z\dot{\alpha}}}\hat{u} + \frac{C_{z\alpha}}{\mu - C_{z\dot{\alpha}}}\alpha - \frac{C_{\mathrm{Le}} \tan \Theta_{\mathrm{e}}}{\mu - C_{z\dot{\alpha}}}\theta + \frac{\mu + C_{zq}}{\mu - C_{z\dot{\alpha}}}\hat{q}, \tag{4.96}$$

$$\dot{\hat{q}} = \frac{1}{j_y}\left[C_{mu} + \frac{C_{\mathrm{m}\dot{\alpha}}\,(C_{zu} - 2C_{\mathrm{Le}})}{\mu - C_{z\dot{\alpha}}}\right]\hat{u} + \frac{1}{j_y}\left[C_{\mathrm{m}\alpha} + \frac{C_{\mathrm{m}\dot{\alpha}} C_{z\alpha}}{\mu - C_{z\dot{\alpha}}}\right]\alpha$$

$$-\frac{C_{\mathrm{m}\dot{\alpha}} C_{\mathrm{Le}} \tan \Theta_{\mathrm{e}}}{j_y\,(\mu - C_{z\dot{\alpha}})}\theta + \frac{1}{j_y}\left[C_{mq} + \frac{C_{\mathrm{m}\dot{\alpha}}\,(\mu + C_{zq})}{\mu - C_{z\dot{\alpha}}}\right]\hat{q}, \tag{4.97}$$

where the nondimensional moment of inertia about the pitch axis is the following:

$$j_y = \frac{8J_{yy}}{\rho S \bar{c}^3}. \tag{4.98}$$

The longitudinal dynamics state equation can be written in the nondimensional form as:

$$\begin{Bmatrix} \dot{\hat{u}} \\ \dot{\hat{\alpha}} \\ \dot{\theta} \\ \dot{\hat{q}} \end{Bmatrix} = \hat{\mathsf{A}}_{\mathrm{Long}} \begin{Bmatrix} \hat{u} \\ \alpha \\ \theta \\ \hat{q} \end{Bmatrix}, \tag{4.99}$$

where

$$\hat{\mathsf{A}}_{\mathrm{Long}} =$$

$$\begin{bmatrix} \frac{(C_{xu}+2C_{\mathrm{Le}}\tan\Theta_{\mathrm{e}})}{\mu} & \frac{C_{x\alpha}}{\mu} & -\frac{C_{\mathrm{Le}}}{\mu} & 0 \\ \frac{C_{zu}-2C_{\mathrm{Le}}}{\mu-C_{z\dot{\alpha}}} & \frac{C_{z\alpha}}{\mu-C_{z\dot{\alpha}}} & -\frac{C_{\mathrm{Le}}\tan\Theta_{\mathrm{e}}}{\mu-C_{z\dot{\alpha}}} & \frac{\mu+C_{zq}}{\mu-C_{z\dot{\alpha}}} \\ 0 & 0 & 0 & 1 \\ \frac{C_{mu}}{j_y} + \frac{C_{\mathrm{m}\dot{\alpha}}(C_{zu}-2C_{\mathrm{Le}})}{j_y(\mu-C_{z\dot{\alpha}})} & \frac{C_{\mathrm{m}\alpha}}{j_y} + \frac{C_{\mathrm{m}\dot{\alpha}}C_{z\alpha}}{j_y(\mu-C_{z\dot{\alpha}})} & -\frac{C_{\mathrm{m}\dot{\alpha}}C_{\mathrm{Le}}\tan\Theta_{\mathrm{e}}}{j_y(\mu-C_{z\dot{\alpha}})} & \frac{C_{mq}}{j_y} + \frac{C_{\mathrm{m}\dot{\alpha}}(\mu+C_{zq})}{j_y(\mu-C_{z\dot{\alpha}})} \end{bmatrix}$$

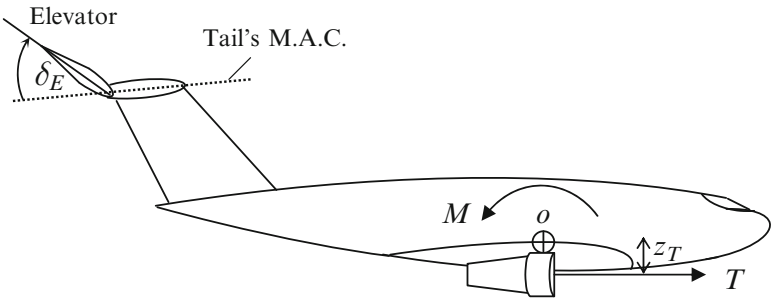


Fig. 4.9 The sign convention for positive elevator and thrust inputs for longitudinal control

Note that the nondimensional natural frequencies derived from the eigenvalues of $\hat{\mathsf{A}}_{\text{Long}}$ must be divided by the time factor, $\bar{c}/(2U)$, to get the dimensional natural frequencies.

4.3.3 Longitudinal Control

The control for longitudinal dynamics is exercised through the deflection of *elevator* – a control surface mounted on the horizontal tail – and the *engine throttle* that can change the thrust, T, (within the capacity of the engine) by regulating the fuel supply. While the elevator alone can effectively control the changes in pitch angle and the angle-of-attack, the control of the airspeed additionally requires the engine throttle input. The sign convention for the elevator deflection (the angle between the elevator chord and the tail's MAC), δ_{E}, measured is shown in Fig. 4.9 where a positive deflection causes a decrease in the tail lift, thereby applying a positive pitching moment about the center of mass, o. Generally, the engines are mounted very close to the center of mass and hence do not have an appreciable pitching moment contribution when a throttle input is applied, changing the thrust by a small amount. Furthermore, the thrust direction is generally along the flight direction in a well designed, conventional[7] airplane.

With a positive elevator deflection, we have the modified tail lift given by

$$\begin{aligned} C'_{\text{Lt}} &= -C_{\text{Lt}\delta}\delta_{\text{E}} \\ C'_{\text{m}} &= -\eta V_{\text{t}} C'_{\text{Lt}} = \eta V_{\text{t}} C_{\text{Lt}\delta}\delta_{\text{E}}, \end{aligned} \tag{4.100}$$

[7]Some special airplanes can have a thrust vectoring engine for better maneuverability, or for a shorter runway length requirement.

where the control derivative

$$C_{\mathrm{L}t\delta} = -\frac{\partial C_{\mathrm{L}t}}{\partial \delta_{\mathrm{E}}}, \tag{4.101}$$

is called *elevator effectiveness*, while

$$C_{\mathrm{m}\delta} = \frac{\partial C_{\mathrm{m}}}{\partial \delta_{\mathrm{E}}} = \eta V_t C_{\mathrm{L}t\delta}, \tag{4.102}$$

is termed as the *elevator power*. When no external moment is applied by either the pilot or an automatic control system on the elevator hinge line, the elevator is free to assume any deflection as dictated by the pressure distribution prevailing on it. Such a condition is referred to as *free elevator* (or *controls free*) case. Generally, the elevator is designed such that the free elevator deflection is quite close to the deflection required for maintaining the airplane in pitch equilibrium ($C_{\mathrm{m}} = 0$) at normal flying speeds, thereby requiring only a small elevator hinge moment to be applied by a controller. However, a free floating elevator reduces the static stability margin when compared to the controls fixed case [5], which implies that the controls free neutral point is forward with controls fixed. The actual case is, when a controller applies a nonzero, but finite hinge moment lies between the two extremes of infinite hinge moment (controls fixed) and zero hinge moment (controls free). Therefore, determination of both controls fixed and free neutral points is necessary for obtaining the bounds on the true stability margin. In our further treatment, we will assume that movement of elevator has been taken into account while computing the stability derivatives.

Both elevator and engine thrust are physically modeled by a separate actuating mechanism, which can be generally assumed to be a linear, second-order system. For elevator actuator, we have the governing differential equation

$$J_E \ddot{\delta}_E + C_E \dot{\delta}_{\mathrm{E}} + K_E \delta_{\mathrm{E}} = H_E, \tag{4.103}$$

where J_E is the constant rotary inertia, C_E the constant rotary viscous damping, K_E the constant rotary spring stiffness, and $H_E(t)$ the externally applied elevator hinge moment (either by a pilot, or an automatic controller). Equation (4.103) can be expressed in a nondimensional form by the transfer function

$$\frac{\delta_{\mathrm{E}}(s)}{C_{hE}(s)} = \frac{1/j_E}{s^2 + 2\zeta_E \omega_E s + \omega_E^2}, \tag{4.104}$$

where (ω_E, ζ_E) are the natural frequency and damping ratio, respectively, of the elevator actuator,

$$C_{hE} = \frac{H_E}{\frac{1}{2}\bar{q} S \bar{c}}, \tag{4.105}$$

is the elevator hinge moment coefficient, and

$$j_E = \frac{8J_E}{\bar{q}S\bar{c}^3} \tag{4.106}$$

the nondimensional elevator moment of inertia. The elevator actuator is either an electric motor, or an electro-hydraulic device.

The change in the thrust coefficient, C_{T}, can be modeled by the following second-order transfer function with respect to throttle valve deflection, β_{T}:

$$\frac{C_{\mathrm{T}}(s)}{\beta_{\mathrm{T}}(s)} = \frac{a_T/j_T}{s^2 + 2\zeta_T\omega_T s + \omega_T^2}, \tag{4.107}$$

where (ω_T, ζ_T) are the natural frequency and damping ratio, respectively, of the engine actuator, a_T is a nondimensional constant giving the thrust coefficient at equilibrium point per dimensionless engine shaft speed, n,

$$a_T = \frac{C_{\mathrm{T}}(s)}{n(s)}, \tag{4.108}$$

and j_T is the nondimensional moment of inertia of engine shaft. The values of the engine constants depend upon the type and maximum power of the engine. Generally, a propeller engine responds much more rapidly to changes in throttle setting than a jet engine. Also, a smaller engine has a faster response.

The actuating subsystem must faithfully and rapidly follow the commands of the control system. The settling-time of the actuator must be much smaller than that of the dominant dynamics being controlled. Generally, the elevator actuator must have a settling-time quite small compared to that of the short-period mode, while the engine actuator must respond in a fraction of the phugoid time period. In both cases, there should neither be a large overshoot nor a substantial steady-state error. Often, the actuator dynamics is unacceptable as it requires either a large settling-time, a large overshoot, or an appreciable steady-state error due to incorrect natural frequency and damping. In such cases, a feedback control system called *servo-actuator* (or in short, *servo*) must be designed and implemented for a fast and well damped actuation.

4.3.3.1 Elevator Servo

The elevator servo must have a zero steady-state error to a step commanded elevator deflection, δ_{Ec}, and a settling-time about one-tenth of that of the short-period mode. This is achieved by providing an integral action in the loop (since the actuator plant is Type-0) by a PID controller with transfer function,

$$H(s) = \frac{C_{hE}(s)}{E(s)} = k_1 + k_2 s + \frac{k_3}{s}, \tag{4.109}$$

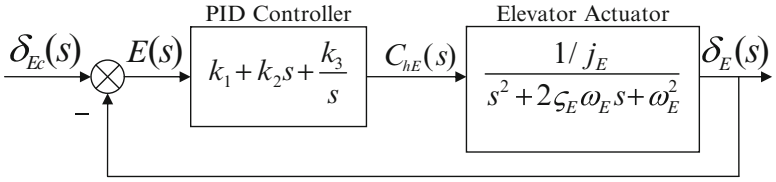


Fig. 4.10 Block diagram of an elevator servo with PID control

where $E(s) = \delta_{\mathrm{Ec}}(s) - \delta_{\mathrm{E}}(s)$, the deflection error and (k_1, k_2, k_3) are controller constants pre-selected for an acceptable servo dynamics. An optical angle encoder mounted on the elevator shaft can pick-up both angle, $\delta_{\mathrm{E}}(t)$, and rate, $\dot{\delta}_{\mathrm{E}}$, required by the proportional and derivative parts of the controller. Alternatively, a sensor combination of a potentiometer and a tachometer can be used for the same purpose. The integral action can be provided either by an operational amplifier, or approximately by a lag compensator circuit (Chap. 3) as part of controller architecture. The block diagram of an elevator servo is shown in Fig. 4.10, where the plant transfer function for the actuator is

$$G(s) = \frac{\delta_{\mathrm{E}}(s)}{C_{hE}(s)}. \tag{4.110}$$

The closed-loop transfer function of the elevator servo is thus denoted as:

$$\frac{\delta_{\mathrm{E}}(s)}{\delta_{\mathrm{Ec}}(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{\left(k_1 s + k_2 s^2 + k_3\right)/j_E}{s^3 + \left(\frac{k_2}{j_E} + 2\zeta_E \omega_E\right) s^2 + \left(\frac{k_1}{j_E} + \omega_E^2\right) s + \frac{k_3}{j_E}}. \tag{4.111}$$

The closed-loop characteristic polynomial can be factored into a real pole and a pair of complex conjugate poles,

$$s^3 + \left(\frac{k_2}{j_E} + 2\zeta_E \omega_E\right) s^2 + \left(\frac{k_1}{j_E} + \omega_E^2\right) s + \frac{k_3}{j_E} = (s + a)\left(s^2 + 2\zeta\omega s + \omega^2\right). \tag{4.112}$$

The optimal placement of poles is in a Butterworth pattern (Chap. 3) such that the poles are equidistant from the origin. For a third-order system like the elevator servo, we require for minimal controller input, $a = \omega$. However, in order to have the servo respond as a dominant second-order system with near critical damping, ζ, the real pole is often moved deeper into the left-half s-plane, e.g., $a = 2\zeta\omega$. Then the closed-loop natural frequency, ω, is chosen from the settling-time requirement, i.e.,

$$\omega \simeq \frac{4}{t_{\mathrm{s}}\zeta} = \frac{10\zeta_{\mathrm{s}}\omega_{\mathrm{s}}}{\zeta}, \tag{4.113}$$

where the servo is required to settle ten times faster than the short-period mode.

Example 4.6. Consider an elevator actuator with

$$\zeta_E = 0.01, \quad \omega_E = 15 \text{ rad/s}, \quad J_E = 20{,}000 \text{ kg m}^2,$$

mounted on an airplane with

$$m = 30{,}000 \text{ kg}, \quad \bar{c} = 4 \text{ m}, \quad S = 75 \text{ m}^2.$$

Design an elevator servo for this airplane for normal operation at $U = 150$ m/s and $\rho = 0.34$ kg/m^3 when the short-period mode is given by $\omega_s = 3$ rad/s and $\zeta_s = 0.56$. Plot the closed-loop servo response to a step commanded elevator deflection of 1°.

We begin by observing that the settling-time requirement for the servo is $t_s = 0.4/(3 \times 0.56) = 0.2381$ s. Choosing $\zeta = 0.7$ for the servo, we require $\omega = 4/(0.7 \times 0.2381) = 24$ rad/s. Furthermore, we select the real pole of the closed-loop PID servo to be $a = 2\zeta\omega = 33.6$ rad/s. Thus, we have

$$s^3 + \left(\frac{k_2}{j_E} + 2\zeta_E\omega_E\right)s^2 + \left(\frac{k_1}{j_E} + \omega_E^2\right)s + \frac{k_3}{j_E} = (s + 33.6)\left(s^2 + 33.6s + 576\right),$$

where

$$j_E = \frac{8J_E}{\bar{q}S\bar{c}^3} = \frac{16 \times 20{,}000}{.34 \times 150^2 \times 75 \times 4^3} = 0.0012047.$$

Comparing the coefficients in the closed-loop characteristic polynomial, we need

$$k_1 = 1.78277, \quad k_2 = 0.08059, \quad k_3 = 23.3135.$$

We construct the closed-loop elevator servo and obtain its unit step response with MATLAB's *Control Systems Toolbox* as follows:

```
>> sysG=tf(1/0.0012047,[1 .3 225]) % Actuator transfer function

Transfer function:
      830.1
-----------------
s^2 + 0.3 s + 225

>> sysH=tf([k2 k1 k3],[0 1 0])%PID controller transfer function

Transfer function:
0.08059 s^2 + 1.783 s + 23.31
-----------------------------
              s

>> sysOL=series(sysH,sysG) %open-loop servo transfer function

Transfer function: 66.9 s^2 + 1,480 s + 1.935e004
-----------------------------
    s^3 + 0.3 s^2 + 225 s

>> sys=feedback(sysOL,1) %closed-loop servo transfer function
```

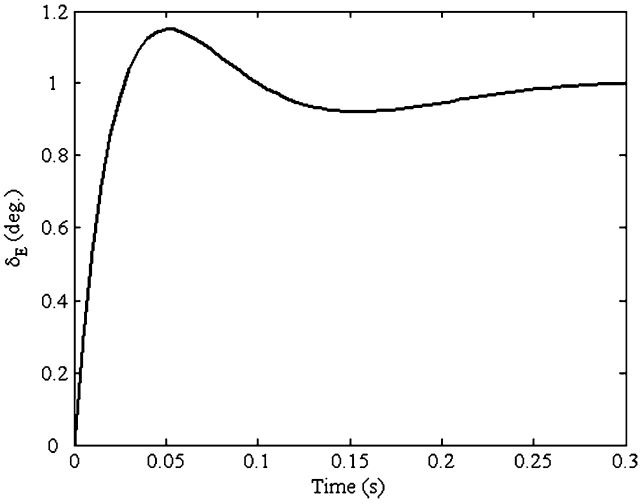


Fig. 4.11 Closed-loop unit step response of an elevator servo with PID control

```
Transfer function:
   66.9 s^2 + 1,480 s + 1.935e004
-----------------------------------
s^3 + 67.2 s^2 + 1,705 s + 1.935e004

>> damp(sys) % closed-loop servo poles

        Eigenvalue          Damping     Freq. (rad/s)

 -1.68e+001 + 1.71e+001i    7.00e-001   2.40e+001
 -1.68e+001 - 1.71e+001i    7.00e-001   2.40e+001
 -3.36e+001                 1.00e+000   3.36e+001

>> step(sys); %step response to 1 deg. commanded deflection
```

The resulting step response is plotted in Fig. 4.11. Note that a settling-time of $t_s = 0.245$ s is achieved with a 15% maximum overshoot. In order to reduce the overshoot, the real pole could be moved towards the origin, which will, however, increase the settling-time.

4.3.3.2 Engine Servo

The engine thrust must be accurately controlled by a servo in order to achieve a small steady-state error to a step commanded thrust, $C_{Tc}(s)$, and a settling-time a fraction of that of the phugoid mode. However, in contrast with the elevator servo, the engine dynamics is much more complicated. The relationship between engine shaft speed, n, and thrust coefficient, C_T, is generally nonlinear and depends not only upon the throttle input, but also the airspeed and altitude. Furthermore, it is difficult to obtain a linear transfer function between the throttle input, β_T, and the shaft

speed. Fortunately, when the engine operation is near an equilibrium flight speed and altitude, linear approximations can be employed with the transfer functions, $C_{\mathrm{T}}(s)/n(s)$, and $n(s)/\beta_{\mathrm{T}}(s)$, examples of which are given by (4.107) and (4.108).

While a sophisticated engine control system – such as full authority, digital engine control (FADEC) that carefully regulates fuel supply depending upon freestream variables – is commonly used in a modern, high-speed airplane, we shall restrict ourselves to a second-order engine dynamics and a much simpler PID servo controller for flight control applications. Therefore, our model for engine servo is similar to that of the elevator servo, with the essential difference that the settling-time is often an order of magnitude larger than that of the elevator servo. Also, since the engine dynamics is normally well damped, there is no need to provide additional damping by derivative action (which is good because derivative action is usually problematic from noise amplification point of view [20]). This might suggest that we require a PI controller that provides integral action for zero steady-state error to a step command and modifies the natural frequency to achieve a reduction in the settling-time. However, the presence of proportional feedback in a third-order closed-loop system would either cause an *increase* of settling-time, or a large overshoot, unless accompanied by derivative action. Since there is normally no need to change the engine dynamics dramatically, we will remove the proportional gain which would otherwise degrade the closed-loop step response. Therefore, an integral controller is employed to reduce the steady state error to zero, while accepting a slight reduction in closed-loop damping.

For the integral engine servo controller, we have

$$H(s) = \frac{\beta_{\mathrm{T}}(s)}{E(s)} = \frac{k}{s}, \tag{4.114}$$

where

$$E(s) = n_c(s) - n(s) = \frac{1}{a_T}\left[C_{\mathrm{Tc}}(s) - C_{\mathrm{T}}(s)\right],$$

the engine shaft speed error and k, is the controller constant pre-selected for an acceptable servo dynamics. An optical angle encoder mounted on the engine shaft can pick-up both angular speed, $n(t)$, and angular position, $\int n(t)\mathrm{d}t$, thereby providing integral action for the controller. Alternatively, an operational amplifier can be used with angular speed feedback from a tachometer for the same purpose. The block diagram of an engine servo is shown in Fig. 4.12, where the plant transfer function between nondimensional engine shaft speed and throttle valve deflection is

$$G(s) = \frac{n(s)}{\beta_{\mathrm{T}}(s)}. \tag{4.115}$$

The closed-loop transfer function of the engine servo is calculated as follows:

$$\frac{C_{\mathrm{T}}(s)}{C_{\mathrm{Tc}}(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{k/j_T}{s^3 + 2\zeta_T\omega_T s^2 + \omega_T^2 s + \frac{k}{j_T}}. \tag{4.116}$$

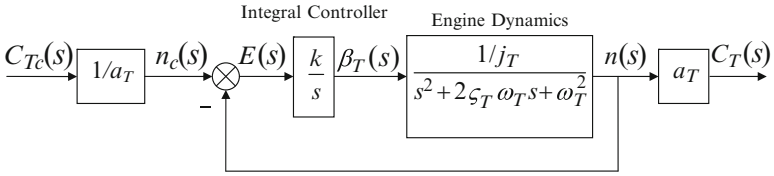


Fig. 4.12 Block diagram of an engine servo with integral control

The closed-loop characteristic polynomial can be factored into a real pole and a pair of complex conjugate poles,

$$s^3 + 2\zeta_T\omega_T s^2 + \omega_T^2 s + \frac{k}{j_T} = (s+a)\left(s^2 + 2\zeta\omega s + \omega^2\right). \tag{4.117}$$

which implies

$$\begin{aligned} 2\zeta\omega + a &= 2\zeta_T\omega_T \\ \omega^2 + 2\zeta\omega a &= \omega_T^2 \\ a\omega^2 &= \frac{k}{j_T}, \end{aligned} \tag{4.118}$$

or

$$k = 2j_T\omega^2(\zeta_T\omega_T - \zeta\omega), \tag{4.119}$$

where

$$\omega = \omega_T \frac{-2\zeta\zeta_T \pm \sqrt{1-4\zeta^2(1-4\zeta_T^2)}}{1-4\zeta^2}. \tag{4.120}$$

Clearly, the largest possible value of closed-loop damping ratio is

$$\zeta = \frac{1}{2\sqrt{1-\zeta_T^2}}, \tag{4.121}$$

for which

$$\omega = \omega_T\zeta_T\sqrt{1-\zeta_T^2}. \tag{4.122}$$

With any choice of ζ, we will invariably have $\zeta\omega < \zeta_T\omega_T$, which implies an increase in settling-time compared to that of the plant. A possible design choice that produces a well damped response without drastically increasing the settling-time is $\zeta\omega = \zeta_T\omega_T - 1/2$, which gives the integral gain as:

$$k = j_T\omega^2 = j_T\left(1 + \omega_T^2 - 2\zeta_T\omega_T\right). \tag{4.123}$$

Example 4.7. Consider a large jet engine with

$$\zeta_T = 0.3, \quad \omega_T = 3 \text{ rad/s}, \quad j_T = 4,$$

mounted on an airplane with

$$\bar{c} = 6.4 \text{ m}, \quad S = 321 \text{ m}^2,$$

for design operation at $\rho = 0.35 \text{ kg/m}^3$. Design a suitable engine servo and plot the closed-loop servo response to a step commanded nondimensional angular speed change, $n_c = 1$.

The plant has a reasonable settling-time, $t_s = 4/0.9 = 4.44$ s, and a steady-state error,

$$e_\infty = \lim_{s \to 0} sE(s) = 1 - G(0) = 1 - \frac{1/4}{9} = 0.9722,$$

which is unacceptable. We construct the integral feedback servo with three likely values of the integral gain, $k = 20°, \ 25°, \ 30°$, and obtain its unit step response with MATLAB's *Control Systems Toolbox* as follows:

```
>> sysG=tf(1/4,[1 2*.3*3 3^2]) % Engine plant

Transfer function:
     0.25
---------------
s^2 + 1.8 s + 9

>> sysH=tf(25,[1 0]) % Controller, H(s)=25/s

Transfer function:
25
--
s

>> sysOL=series(sysH,sysG) % Open-loop servo

Transfer function:
       6.25
-------------------
s^3 + 1.8 s^2 + 9 s

>> sys=feedback(sysOL,1) % Closed-loop servo

Transfer function:
           6.25
--------------------------
s^3 + 1.8 s^2 + 9 s + 6.25

>> damp(sys) % Closed-loop poles

Eigenvalue                  Damping     Freq. (rad/s)

 -7.61e-001                 1.00e+000       7.61e-001
 -5.19e-001 + 2.82e+000i    1.81e-001       2.87e+000
 -5.19e-001 - 2.82e+000i    1.81e-001       2.87e+000
```

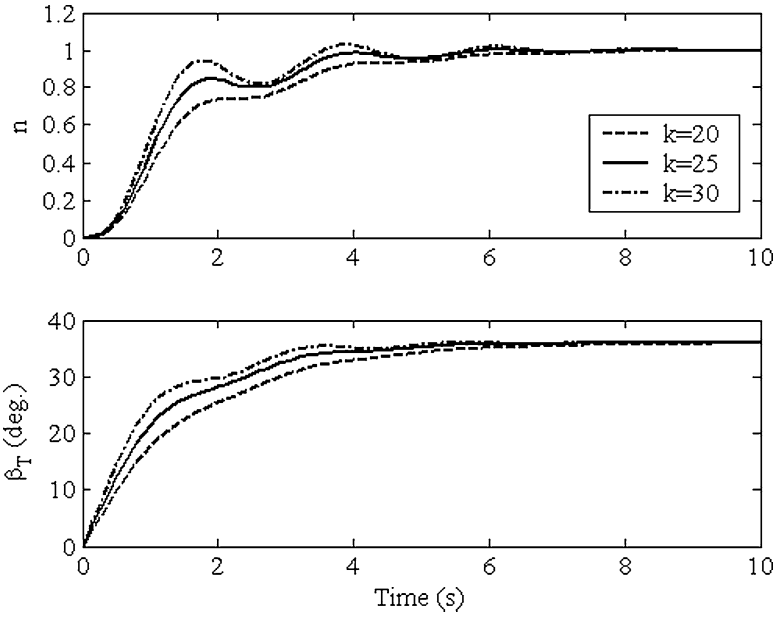


Fig. 4.13 Closed-loop unit step response of shaft speed and throttle valve position for integral engine servo with $k = 20°,\ 25°,\ 30°$

```
>> [y,t]=step(sys,0:0.1:10); % Servo step response

>> sysu=feedback(sysH,sysG)
% TF from commanded shaft speed to throttle angle

Transfer function:
   25 s^2 + 45 s + 225
--------------------------
s^3 + 1.8 s^2 + 9 s + 6.25

>> [u,t]=step(sysu,0:0.1:10); % Throttle response
```

The shaft speed and throttle response to a step commanded shaft speed for servo designs with $k = 20°,\ 25°,\ 30°$ are plotted in Fig. 4.13, while the shaft speed error of the three servo designs as well as the plant are shown in Fig. 4.14. While the smallest closed-loop settling-time of $t_s = 5.4$ s is obtained for $k = 30°$ with a maximum overshoot of 3%, the response without an overshoot (but $t_s = 6.2$ s) is observed for $k = 20°$. Thus, $k = 25°$ is the best design that offers a compromise between a small overshoot (0.2%) and an acceptable settling-time (5.6 s). It requires a maximum (steady-state) throttle valve angle, $\beta_T = 30°$ for the unit step shaft speed command.

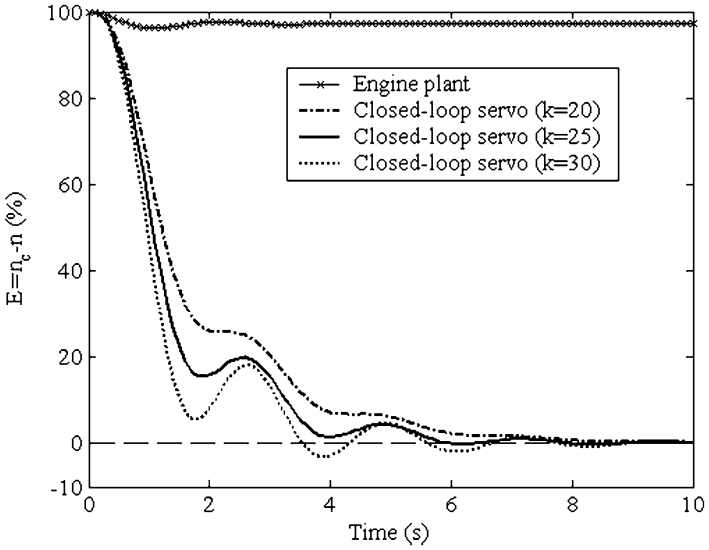


Fig. 4.14 Unit step shaft speed error response of engine plant and integral engine servos with $k = 20°,\ 25°,\ 30°$

4.3.3.3 Aircraft Plant for Longitudinal Control

The aircraft plant for longitudinal control is derived by adding the control contributions to axial force, normal force, and pitching moment equations through *control derivatives*,

$$X_T = \left(\frac{\partial X}{\partial \beta_{\mathrm{T}}}\right)_{\mathrm{e}},$$

$$Z_\delta = \left(\frac{\partial Z}{\partial \delta_{\mathrm{E}}}\right)_{\mathrm{e}},$$

etc., or their respective nondimensional forms,

$$C_{xT} = \left(\frac{\partial C_x}{\partial \beta_{\mathrm{T}}}\right)_{\mathrm{e}},$$

$$C_{z\delta} = \left(\frac{\partial C_z}{\partial \delta_{\mathrm{E}}}\right)_{\mathrm{e}},$$

etc. The rate of change of the control surface and throttle is generally small enough for the rate derivatives,

$$\left(\frac{\partial X}{\partial \dot{\delta}_{\mathrm{T}}}\right)_{\mathrm{e}}, \qquad \left(\frac{\partial Z}{\partial \dot{\delta}_{\mathrm{E}}}\right)_{\mathrm{e}},$$

etc., to be negligible.

The nondimensional static control derivatives due to throttle input, β_{T}, are derived from the steady-state engine actuator transfer function (DC gain) by using the final value theorem of Laplace transform and (4.107) as follows:

$$C_{xT} = \left(\frac{\partial C_{\mathrm{T}}}{\partial \beta_{\mathrm{T}}}\right)_{\mathrm{e}} = \lim_{s\to 0} s\frac{G(s)}{s} = G(0) = \frac{a_T}{j_T \omega_T^2}. \tag{4.124}$$

The pitching moment contribution of thrust can be seen from Fig. 4.9 to be the following:

$$C_{mT} = \left(\frac{\partial C_{\mathrm{m}}}{\partial \beta_{\mathrm{T}}}\right)_{\mathrm{e}} = \frac{2z_T}{\bar{c}} C_{xT}. \tag{4.125}$$

A well designed airplane must have negligible normal force and pitching moment thrust contributions, $C_{zT} = C_{mT} = 0$, otherwise there would be a degraded closed-loop transient response to throttle input.

The nondimensional elevator control derivatives are obtained by following (4.101), (4.102):

$$\begin{aligned} C_{z\delta} &= -\left(\frac{\partial C_{\mathrm{L}}}{\partial \delta_{\mathrm{E}}}\right)_{\mathrm{e}} = \frac{S_{\mathrm{t}}}{S} C_{\mathrm{L}t\delta} \\ C_{\mathrm{m}\delta} &= \eta V_{\mathrm{t}} C_{\mathrm{L}t\delta}. \end{aligned} \tag{4.126}$$

Therefore, the controlled longitudinal dynamics in the state space form can be expressed as

$$\dot{\mathbf{x}} = \hat{\mathsf{A}}_{\mathrm{Long}}\mathbf{x} + \hat{\mathsf{B}}_{\mathrm{Long}}\mathbf{u}, \tag{4.127}$$

where the state vector is given by

$$\mathbf{x} = \begin{Bmatrix} \hat{u} \\ \alpha \\ \theta \\ \hat{q} \end{Bmatrix}, \tag{4.128}$$

the control input vector by

$$\mathbf{u} = \begin{Bmatrix} \delta_{\mathrm{E}} \\ \beta_{\mathrm{T}} \end{Bmatrix}, \tag{4.129}$$

the state dynamics matrix, $\hat{\mathsf{A}}_{\mathrm{Long}}$, is the same as that given in (4.99), and the *control coefficient matrix* is as follows:

$$\hat{\mathsf{B}}_{\mathrm{Long}} = \begin{pmatrix} \frac{C_{x\delta}}{\mu} & \frac{C_{xT}}{\mu} \\ \frac{C_{z\delta}}{\mu - C_{z\dot{\alpha}}} & \frac{C_{zT}}{\mu - C_{z\dot{\alpha}}} \\ 0 & 0 \\ \frac{C_{\mathrm{m}\delta}}{j_y} + \frac{C_{\mathrm{m}\dot{\alpha}} C_{z\delta}}{j_y(\mu - C_{z\dot{\alpha}})} & \frac{C_{mT}}{j_y} + \frac{C_{\mathrm{m}\dot{\alpha}} C_{zT}}{j_y(\mu - C_{z\dot{\alpha}})} \end{pmatrix}. \tag{4.130}$$

Note that when engine throttle input is the control input, the engine servo[8] is replaced by a throttle servo with an appropriate transfer function, $\beta_{\mathrm{T}}(s)/\beta_{\mathrm{Tc}}(s)$. In a dimensional form, the control coefficient matrix is written as the following:

$$\mathsf{B}_{\mathrm{Long}} = \begin{pmatrix} \frac{X_\delta}{m} & \frac{X_T}{m} \\ \frac{Z_\delta}{mU - Z_{\dot\alpha}} & \frac{Z_T}{mU - Z_{\dot\alpha}} \\ 0 & 0 \\ \frac{M_\delta}{J_{yy}} + \frac{M_{\dot\alpha} Z_\delta}{J_{yy}(mU - Z_{\dot\alpha})} & \frac{M_T}{J_{yy}} + \frac{M_{\dot\alpha} Z_T}{J_{yy}(mU - Z_{\dot\alpha})} \end{pmatrix}. \tag{4.131}$$

The aircraft longitudinal plant augmented by second-order elevator and throttle servos is given by the following state equation:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathsf{A}}\bar{\mathbf{x}} + \bar{\mathsf{B}}\bar{\mathbf{u}}, \tag{4.132}$$

where the augmented state vector is given by

$$\bar{\mathbf{x}} = \begin{Bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{Bmatrix}, \tag{4.133}$$

the commanded control input vector by

$$\bar{\mathbf{u}} = \begin{Bmatrix} \delta_{\mathrm{Ec}} \\ \beta_{\mathrm{Tc}} \end{Bmatrix}, \tag{4.134}$$

$$\bar{\mathsf{A}} = \begin{pmatrix} \hat{\mathsf{A}}_{\mathrm{Long}} & \hat{\mathsf{B}}_{\mathrm{Long}} & 0 \\ 0 & 0 & \mathsf{I} \\ 0 & -\mathsf{K}_s & -\mathsf{C}_s \end{pmatrix}, \tag{4.135}$$

$$\bar{\mathsf{B}} = \begin{pmatrix} 0 \\ 0 \\ \mathsf{K}_s \end{pmatrix}, \tag{4.136}$$

with

$$\mathsf{K}_s = \begin{pmatrix} \omega_E^2 & 0 \\ 0 & \omega_T^2 \end{pmatrix},$$

$$\mathsf{C}_s = \begin{pmatrix} 2\zeta_E \omega_E & 0 \\ 0 & 2\zeta_T \omega_T \end{pmatrix}. \tag{4.137}$$

[8] In such a case, the engine servo is part of the aircraft plant.

4.4 Automatic Longitudinal Control

Automatic attitude control systems for aircraft are required for improving the stability characteristics, as well as for tracking a command signal input by the pilot or an automatic navigation system in order to meet a kinematic constraint. For the longitudinal case, the two aspects are often combined into a single controller called an *autopilot*. The autopilot tasks can thus be divided into two categories: (1) maintaining the vehicle near a pre-set flight condition despite atmospheric disturbances and fuel consumption (thereby relieving pilot fatigue in a long duration flight), and (2) performing a high-precision task that may be too demanding for a pilot, such as landing in poor visibility conditions. In either case, the autopilot must faithfully track the changing commands from either the pilot or the automatic navigation system, while compensating for any undesirable behavior of the open-loop, longitudinal plant.

An autopilot requires feedback of certain output variables that are supplied by (a) *motion sensors* such as rate gyro for measuring q, vertical gyro for θ, and accelerometer (located at distance ℓ aft of the center of mass) for normal acceleration, $a_z = U(\dot{\alpha} - q) + \ell\dot{q}$, (b) *navigational sensors* such as radar altimeter, global positioning system (GPS), and inertial navigation system (INS), and (c) *air data sensors*, which measure static and stagnation pressures and temperatures as well as the relative flow direction. The air data is processed by a *central air data computer* (CADC) in order to generate flow parameters such as the altitude, h, airspeed, U, dynamic pressure, $\bar{q}$, Mach number, $\mathcal{M}$, and angle-of-attack, α.

Example 4.8. Consider a longitudinal autopilot designed to follow a reference altitude profile in km,

$$h_d(t) = 5 - 0.001\left[\{x(t) - 10\}^3 + 15x(t)^2 - 300x(t)\right] \text{ km};$$
$$(0 \le x(t) \le 10 \text{ km}),$$

while maintaining a constant airspeed, U, and a contant heading (azimuth), beginning from a straight and level equilibrium condition. Here, $x(t)$ is the downrange measured from $t = 0$ in km. Such a maneuver is necessary in order to increase the cruising altitude by 0.1 km, either for clearing an obstacle in the flight path or as commanded by air traffic control.

The longitudinal dynamic plant is the same as in (4.127). However, the commanded maneuver requires the consideration of translational kinematics equations,

$$\dot{h} = U\sin\phi\ ; \qquad \dot{x} = U\cos\phi,$$

with ϕ being the flight path angle (Chap. 2). This translates into the following command for the flight path angle:

$$\tan\phi_d = \frac{\mathrm{d}h_d}{\mathrm{d}x} = -0.003(x-10)^2 - 0.03x + 0.3; \qquad (0 \le x \le 10 \text{ km}).$$

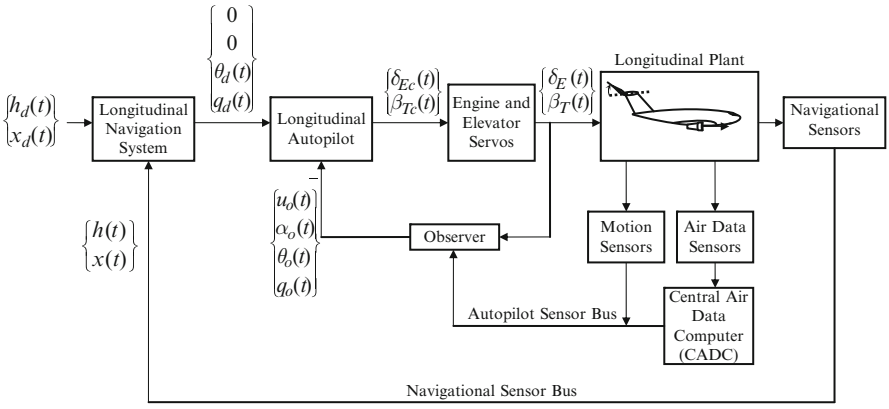


Fig. 4.15 Longitudinal tracking system for a smooth altitude profile with constant airspeed and angle-of-attack

Moreover, since $\phi = \theta - \alpha$, we can achieve the desired flight path by merely changing the pitch angle, $\theta(t)$, while maintaining a nearly constant angle-of-attack, i.e., $\alpha \simeq 0$, and airspeed. Thus, we have $u_d = 0, \alpha_d = 0$ and $\theta_d = \phi_d$. Such a flight profile will also be more comfortable for the passengers, whose drinks could be spilled by a change of angle-of-attack (normal acceleration) or of airspeed (forward acceleration). Furthermore, due to the small angle required, we can used the approximation

$$\theta_d \simeq -0.003(x - 10)^2 - 0.03x + 0.3; \qquad (0 \leq x \leq 10 \text{ km}).$$

The pitch rate command can be derived by differentiating $\theta_d(t)$ with time and assuming $\dot{x} \simeq U$:

$$q_d = \dot{\theta}_d \simeq -0.006(x - 5)U; \qquad (0 \leq x \leq 10 \text{ km}).$$

The autopilot must track the desired trajectory by actuating both elevator and throttle inputs in concert. On the other hand, the automatic navigation system supplies the desired pitch angle at every instant, taking feedback of downrange from either the GPS or the INS. A block-diagram of the resulting tracking system is depicted in Fig. 4.15.

A high-performance aircraft can have varying plant parameters throughout its flight envelope, thus requiring an adjustment of the controller parameters with the flow parameters. A control system that can adapt itself to a changing set of flight parameters is called an *adaptive control system*, and is required by all high-speed aircraft. A commonly used adaptation law is *gain scheduling*, in which pre-determined functions of controller gains with important flight parameters – such as the dynamic

pressure and Mach number – are stored onboard in a flight control computer and continuously supplied in a closed-loop from the CADC to the autopilot.

We shall now consider how the longitudinal flight variables can be individually controlled as basic, single-variable systems. Thereafter, multivariable longitudinal control via state space methods will be addressed.

4.5 Single-Input Longitudinal Control Systems

The traditional method of controlling aircraft longitudinal dynamics is by using either the elevator, or the throttle input at a time (but not both) and feedback of one or more output signals. Such systems can be generally designed by either the transfer function (frequency domain) approach, or state space (time domain) method of pole-placement. While being simple to design and implement, single variable control can suffer from lack of suitable performance and robustness. We shall consider such control systems in the present section.

Since the longitudinal modes of a conventional aircraft generally have widely spaced natural frequencies, it is possible to design single-input control systems for either pitch, or airspeed control using the short-period or phugoid modes, respectively, as the basic plant model. However, when the modes coalesce (as in the case of some supersonic aircraft), it is not feasible to design separate single-input control systems for them, thereby requiring a multivariable approach.

A longitudinal autopilot for maintaining level flight at constant airspeed is the most common application of longitudinal control, which relieves the pilot from fatigue in a long duration flight. Traditional single-input, single-output longitudinal autopilots can be categorized as either pitch (altitude) hold, or airspeed hold mechanisms. However, with the advent of modern, state space design techniques, it is possible to have a simultaneous regulation of both airspeed and altitude solely by the elevator input.

4.5.1 Pitch Control by Elevator Input

Pitch control systems primarily addresses the short-period mode, where the pitch angle, pitch rate, and angle-of-attack are regulated solely by elevator input. The bandwidth of a pitch control system is roughly ten times larger than that of airspeed control, which mainly addresses the phugoid mode.

While pitch controllers are classically designed for using only the approximate, second-order short-period dynamics [2], we will consider the complete longitudinal plant that does not make such an approximation. This allows an active control of both phugoid and short-period modes, thereby enabling a regulation of airspeed as well as the angle-of-attack, and also leads to a better robustness when the control system is finally implemented.

The dimensional state equation of the longitudinal plant, with elevator control is the following:

$$\dot{\mathbf{x}} = \mathsf{A}_{\text{Long}}\mathbf{x} + \mathsf{B}_E\delta_{\text{E}}, \tag{4.138}$$

where B_E is the first column of B_{Long}. Since the pair $(\mathsf{A}_{\text{Long}}, \mathsf{B}_E)$ is controllable, one can design a state feedback regulator by pole placement,

$$\delta_{\text{Ec}} = -\mathsf{K}\mathbf{x}. \tag{4.139}$$

Note that with the elevator input alone, it is not feasible to design a feedforward/feedback tracking system (Chap. 3) for a general, non-zero desired state, $\mathbf{x}_d$, because a feedforward gain matrix, K_d would not satisfy $(\mathsf{A} + \mathsf{B}\mathsf{K}_\mathsf{d})\mathbf{x}_d = \mathbf{0}$ which is necessary for an asymptotically stable error dynamics (Chap. 3). Thus we are dealing strictly with a regulator problem. Instead of state feedback, it is more practical to design an output feedback system with a linear observer using the output, $\mathbf{y}(t)$, to generate an estimated state, $\mathbf{x}_o(t)$, to form a closed-loop compensator with zero desired state. In that case, the control law for commanded elevator deflection becomes

$$\delta_{\text{Ec}} = -\mathsf{K}\mathbf{x}_o. \tag{4.140}$$

4.5.1.1 Airspeed and Altitude Hold Autopilot

The most common longitudinal autopilots are those used for maintaining a level flight attitude despite disturbances. The first such autopilots – that appeared in the 1930s and are still employed in small, general aviation airplanes – were of *displacement* type that worked simply on pitch angle feedback from vertical gyro via an amplifier, to the elevator servo. Thus commanded elevator deflection is made directly proportional to pitch displacement. Since there is no way of regulating either the airspeed or the angle-of-attack in this manner, the airplane can either climb or descend, accelerate or decelerate, while maintaining a level pitch orientation. Such a behavior is compounded by the presence of vertical atmospheric gusts wherein the vehicle might even stall while appearing to fly straight and level. Hence, it is neither possible to maintain a constant airspeed, nor a constant altitude by a pitch displacement autopilot. Furthermore, if the amplifier gain is adjusted too high, there is the likelihood of a sharp reduction in the short-period damping, which could be destabilizing for a modern, high-speed aircraft that does not have a large short-period damping to begin with.

A more practical autopilot is the one that employs an observer based controller for simultaneously regulating the speed, angle-of-attack, pitch angle and pitch rate by elevator input. Such a controller is called a flight stabilization system (FSS) which can maintain both a constant airspeed and a constant altitude by improving closed-loop damping of all state transients. The FSS is indispensable for all modern passenger aircraft as it allows good ride by alleviating the pitch rate and angle-of-attack variations (thus normal acceleration) in the presence of gusts. Usually,

airspeed and altitude as outputs provide the best observer based design due to the small condition number of the associated observability test matrix, N (Chap. 3), of all available outputs.

The altitude, $h = h_e + \Delta h$, is not one of the state variables of the longitudinal plant, but rather has its own state equation given by

$$\dot{h} = U \sin\phi = U \sin(\Theta - \alpha) \simeq U \left[\sin\Theta_{\mathrm{e}} + (\theta - \alpha)\cos\Theta_{\mathrm{e}}\right], \tag{4.141}$$

which for a level flight becomes

$$\dot{h} \simeq U(\theta - \alpha). \tag{4.142}$$

Hence, in order to be successful, the plant for an altitude control autopilot must be augmented by the altitude state equation, (4.141), as follows:

$$\dot{\mathbf{x}} = \mathsf{A}\mathbf{x} + \mathsf{B}\delta_{\mathrm{Ec}}, \tag{4.143}$$

where $\mathbf{x} = (u, \Delta h, \alpha, \theta, q)^{\mathrm{T}}$,

$$\mathsf{A} = \begin{pmatrix} a_{11} & 0 & a_{12} & a_{13} & 0 \\ 0 & 0 & -U & U & 0 \\ a_{21} & 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 & 1 \\ a_{41} & 0 & a_{42} & a_{43} & a_{44} \end{pmatrix}; \quad \mathsf{B} = \begin{pmatrix} b_{11} \\ 0 \\ b_{21} \\ 0 \\ b_{41} \end{pmatrix}, \tag{4.144}$$

and a_{ij}, b_{ij} are the elements of the longitudinal state space coefficient matrices, $\mathsf{A}_{\mathrm{Long}}, \mathsf{B}_{\mathrm{Long}}$, respectively.

When airspeed and altitude measurements are taken as outputs, the coefficient matrices of the output equation are given by

$$\mathsf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}; \quad \mathsf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.145}$$

In order to design a reduced-order observer with the two outputs, we partition the state vector into $\mathbf{x} = (\mathbf{x_1}^{\mathrm{T}}, \mathbf{x_2}^{\mathrm{T}})^{\mathrm{T}}$, where $\mathbf{x_1} = \mathbf{y} = (u, \Delta h)^{\mathrm{T}}$ and $\mathbf{x_2} = (\alpha, \theta, q)^{\mathrm{T}}$. The observer with gain matrix, L, produces a state estimate for $\mathbf{x_2}$ according to (Chap. 3)

$$\mathbf{x_{o2}} = \mathsf{L}\mathbf{y} + \mathbf{z},$$

where $\mathbf{z}(t)$ is the solution of the observer's state equation,

$$\dot{\mathbf{z}} = \mathsf{F}\mathbf{z} + \mathsf{H}\delta_{\mathrm{Ec}} + \mathsf{G}\mathbf{y}.$$

The partitioning of the plant produces the following coefficient matrices,

$$\mathsf{A}_{11} = \begin{pmatrix} a_{11} & 0 \\ 0 & 0 \end{pmatrix} \; ; \; \mathsf{A}_{12} = \begin{pmatrix} a_{12} & a_{13} & 0 \\ -U & U & 0 \end{pmatrix} \; ; \; \mathsf{B}_1 = \begin{pmatrix} b_{11} \\ 0 \end{pmatrix} \tag{4.146}$$

$$\mathsf{A}_{21} = \begin{pmatrix} a_{21} & 0 \\ 0 & 0 \\ a_{41} & 0 \end{pmatrix} \; ; \; \mathsf{A}_{22} = \begin{pmatrix} a_{22} & a_{23} & a_{24} \\ 0 & 0 & 1 \\ a_{42} & a_{43} & a_{44} \end{pmatrix} \; \mathsf{B}_2 = \begin{pmatrix} b_{21} \\ 0 \\ b_{41} \end{pmatrix}, \tag{4.147}$$

and $\mathsf{C}_1 = \mathsf{I}$. The observer's coefficient matrices are thus given by

$$\begin{aligned} \mathsf{F} &= \mathsf{A}_{22} - \mathsf{L}\mathsf{C}_1\mathsf{A}_{12} = \mathsf{A}_{22} - \mathsf{L}\mathsf{A}_{12} \\ \mathsf{G} &= \mathsf{F}\mathsf{L} + (\mathsf{A}_{21} - \mathsf{L}\mathsf{C}_1\mathsf{A}_{11})\,{\mathsf{C}_1}^{-1} \\ &= \mathsf{F}\mathsf{L} + \mathsf{A}_{21} - \mathsf{L}\mathsf{A}_{11} \\ \mathsf{H} &= \mathsf{B}_2 - \mathsf{L}\mathsf{C}_1\mathsf{B}_1 = \mathsf{B}_2 - \mathsf{L}\mathsf{B}_1. \end{aligned} \tag{4.148}$$

The observer gain matrix, L, can be derived by a multi-variable design method, such as the linear, optimal control, or the eigenstructure assignment (Chap. 3), in order that the matrix F has all of its poles in the left-half s-plane.

With the observer based feedback, the control law for the regulator is written as follows:

$$\delta_{\mathrm{Ec}} = -\mathsf{K}_1\mathbf{x_1} - \mathsf{K}_2\mathbf{x_{o2}} = -\left(\mathsf{K}_1 + \mathsf{K}_2\mathsf{L}\right)\mathbf{x_1} - \mathsf{K}_2\mathbf{z}, \tag{4.149}$$

where $\mathsf{K} = (\mathsf{K}_1, \mathsf{K}_2)$ is the partitioned feedback gain matrix. Thus, the regulated closed-loop system has the following state equation:

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{z}} \end{Bmatrix} = \begin{bmatrix} \mathsf{A} - \mathsf{B}(\mathsf{K}_1 + \mathsf{K}_2\mathsf{L}) & -\mathsf{B}\mathsf{K}_2 \\ \{\mathsf{G} - \mathsf{H}(\mathsf{K}_1 + \mathsf{K}_2\mathsf{L})\}\,\mathsf{C} & \mathsf{F} - \mathsf{H}\mathsf{K}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{z} \end{Bmatrix}. \tag{4.150}$$

A schematic block-diagram of the airspeed and altitude hold autopilot is shown in Fig. 4.16.

Example 4.9. Consider the airplane of Example 4.4. The elevator control derivatives at standard sea level with a constant speed, $U = 55$ m/s are as follows:

$$\frac{X_\delta}{m} \simeq 0; \qquad \frac{Z_\delta}{m} = 0.3 \text{ m/s}^2/\text{rad}; \qquad \frac{M_\delta}{J_{yy}} = 0.1243 \text{ s}^{-2}.$$

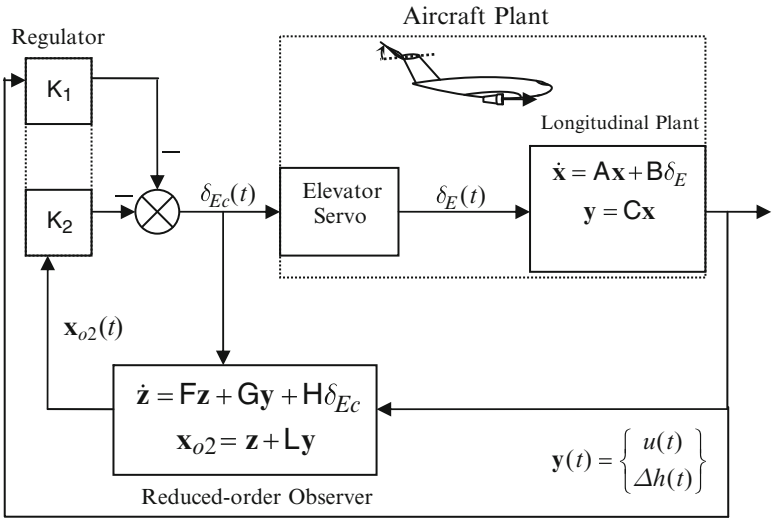


Fig. 4.16 Altitude and airspeed hold autopilot based upon reduced-order observer and elevator control input

The elevator servo is assumed to have the following approximate transfer function:

$$\frac{\delta_{\mathrm{E}}(s)}{\delta_{\mathrm{Ec}}(s)} = \frac{15}{s+15}.$$

Using the airspeed and altitude outputs from a CADC, design an observer based, pitch autopilot for maintaining a constant speed and a level altitude despite an initial perturbation of $\alpha(0) = 5°$ caused by a vertical atmospheric gust. The elevator servo is physically constrained by $|\delta_{\mathrm{E}}| \leq 25°$.

The longitudinal dynamics plant is first augmented by adding the altitude as an additional state variable and the altitude state equation in the state coefficient matrices, A, B. This adds an eigenvalue of $s = 0$ to the plant. We then adopt a pole-placement approach for both the regulator and the full-order observer. The regulator poles are chosen to be $s_1 = -2$, $s_{2,3} = -0.8 \pm 0.8\mathrm{i}$ and $s_{4,5} = -2.5 \pm 2.5\mathrm{i}$ by moving the pole at zero and increasing the damping of both the longitudinal modes to a critical value, $\zeta = 1/\sqrt{2}$, without much affecting their natural frequencies. The regulator gains are calculated by the Ackermann pole-placement formula as follows:

```
>> A=  [-0.0450         0    1.9600   -9.4757         0
              0         0  -55.0000   55.0000         0
         -0.0065         0   -1.9636   -0.0462    1.0000
              0         0         0         0    1.0000
          0.0059         0   -6.8327    0.0415   -2.9000];

>> B = [0    0   0.3    0     0.1243]';

>> K=acker(A,B,[-2,-0.8-0.8i,-0.8+0.8i,-2.5+2.5i,-2.5-2.5i])

K =    -29.8289   -5.8290   -2.9657  104.2397   36.8552
```

```
>> damp(A-B*K) % regulator poles

        Eigenvalue            Damping     Freq. (rad/s)
 -2.50e+000 + 2.50e+000i     7.07e-001      3.54e+000
 -2.50e+000 - 2.50e+000i     7.07e-001      3.54e+000
 -2.00e+000                  1.00e+000      2.00e+000
 -8.00e-001 + 8.00e-001i     7.07e-001      1.13e+000
 -8.00e-001 - 8.00e-001i     7.07e-001      1.13e+000
```

Note that, while the short-period frequency is unchanged from its open-loop value (Example 4.4), there is a five-fold increase in the phugoid frequency. For designing the bi-output observer, we select the eigenstructure assignment approach. The observer poles are chosen to be $s_1 = -0.5$ and $s_{2,3} = -0.5 \pm 0.5\text{i}$ so that they are outside the regulator bandwidth, thereby not affecting the natural frequencies and damping ratios of the closed-loop longitudinal modes by a large amount. Furthermore, increasing the observer gains magnitude and beyond a certain point would destabilize the system.

```
>> A11=A(1:2,1:2)
A11 =
   -0.0450         0
         0         0

>> A12=A(1:2,3:5)
A12 =
    1.9600   -9.4757         0
  -55.0000   55.0000         0

>> A21=A(3:5,1:2)
A21 =
   -0.0065         0
         0         0
    0.0059         0

>> A22=A(3:5,3:5)
A22 =
   -1.9636   -0.0462    1.0000
         0         0    1.0000
   -6.8327    0.0415   -2.9000

>> B1=B(1:2,:)
B1 =
     0
     0

>> B2=B(3:5,:)
B2 =
    0.3000
         0
    0.1243

% Observer gain matrix, L, by eigenstructure assignment:
>> Lp=place(A22',A12',[-0.5,-0.5-0.5i,-0.5+0.5i]);L=Lp'
L =
    0.5203    0.0669
    0.2527    0.0308
    0.1039    0.0734

>> F=A22-L*A12
F =
    0.6984    1.2020    1.0000
    1.1980    0.7016    1.0000
   -2.9990   -3.0110   -2.9000
```

```
>> damp(F)

        Eigenvalue            Damping     Freq. (rad/s)
 -5.00e-001 + 5.00e-001i     7.07e-001      7.07e-001
 -5.00e-001 - 5.00e-001i     7.07e-001      7.07e-001
 -5.00e-001                  1.00e+000      5.00e-001

>> G=F*L+A21-L*A11
G =
    0.7880    0.1572
    0.9159    0.1752
   -2.6122   -0.5063

>> H=B2-L*B1
H =
    0.3000
         0
    0.1243

>> K1=K(1:2)
K1 =
  -29.8289   -5.8290

>> K2=K(3:5)
K2 =
   -2.9657  104.2397   36.8552
```

Finally, we assemble the closed-loop system with the elevator servo, regulator, and observer as follows:

```
>> a=15; % elevator servo pole
>> Abar=[A B;zeros(1,5) -a] % Plant dynamics augmented with elevator servo
Abar =
   -0.0450         0    1.9600   -9.4757         0         0
         0         0  -55.0000   55.0000         0         0
   -0.0065         0   -1.9636   -0.0462    1.0000    0.3000
         0         0         0         0    1.0000         0
    0.0059         0   -6.8327    0.0415   -2.9000    0.1243
         0         0         0         0         0  -15.0000

>> Bbar=[zeros(size(B)); a] % Augmented plant's B matrix
Bbar =
     0
     0
     0
     0
     0
    15

>> Cbar=[1     0     0     0     0     0
         0     1     0     0     0     0]; %Augmented plant's C matrix

>> Ac=[Abar-Bbar*(K1+K2*L)*Cbar  -Bbar*K2;
       (G-H*(K1+K2*L))*Cbar       F-H*K2] % Closed-loop dynamics matrix

Ac =

      -0.045        0    1.96 -9.4757    0      0       0       0       0
           0        0     -55      55    0      0       0       0       0
     -0.0065        0 -1.9636 -0.0462    1    0.3       0       0       0
           0        0       0       0    1      0       0       0       0
      0.0059        0 -6.8327  0.0415 -2.9 0.1243       0       0       0
      17.926   1.6909       0       0    0    -15  44.485 -1563.6 -552.83
      1.1466  0.19099       0       0    0      0  1.5881  -30.07 -10.057
     0.91592   0.1752       0       0    0      0   1.198 0.70158       1
     -2.4636 -0.49233       0       0    0      0 -2.6304 -15.968 -7.4811
```

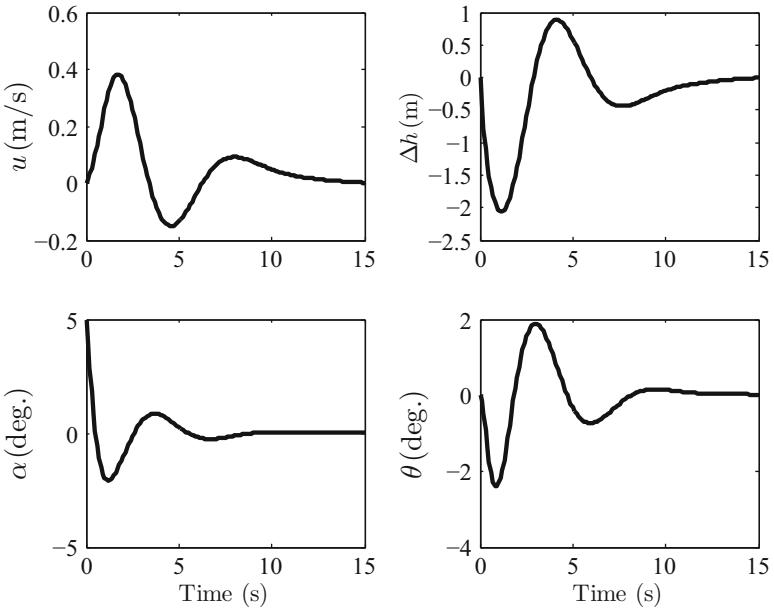


Fig. 4.17 Closed-loop initial response of an observer based altitude and airspeed hold autopilot for an initial angle-of-attack perturbation of 5°

The closed-loop response to the initial condition, $\alpha(0) = 5°$ is obtained and plotted in Figs. 4.17 and 4.18:

```
>> sys=ss(Ac,zeros(9,1),zeros(1, 9),0); % closed-loop homogeneous system
>> [y,t,x]=initial(sys,[0  0 5*pi/180 zeros(1,6)]',15); % closed-loop
   initial response
>> u=-x(:,1:2)*(K1+K2*L)'-x(:,7:9)*K2'; % Commanded elevator deflection (rad)
```

The initial angle-of-attack perturbation corresponds to an initial altitude error of -4.8 m. The closed-loop transients settle in 14 s, with the maximum overshoots of about 0.4 m/s in airspeed, −2 m in altitude, −2.1° in the angle-of-attack, and −2.4° in the pitch angle. The maximum elevator deflection required is −21.44°, which is well within the ±25° limits (Fig. 4.18). This is an excellent performance compared with a classical pitch displacement autopilot.

In order to verify the robustness of the design, we obtain the frequency response of the transfer function between elevator command and airspeed output as follows:

```
% closed-loop system with commanded elevator input and airspeed output:
>> sys=ss(Ac,[Bbar;H],[1 zeros(1,8)],0);
>> tf(sys)

Transfer function:

8.82 s^6 + 24.79 s^5 + 282.3 s^4 + 398 s^3 + 259 s^2 + 64.03 s

                                                    - 1.782e-013
```

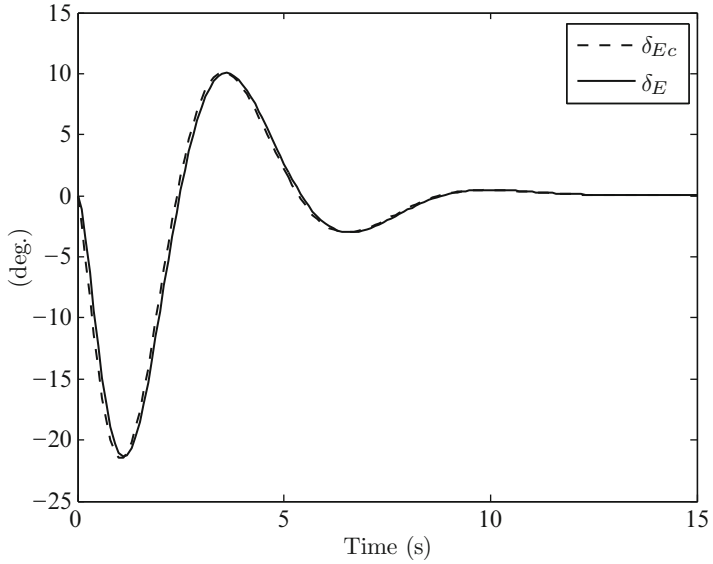


Fig. 4.18 Closed-loop elevator command and response of an observer based altitude and airspeed hold autopilot for an initial angle-of-attack perturbation of 5°

```
-----------------------------------------------------------------------
s^9 + 25.1 s^8 + 199.2 s^7 + 845.7 s^6 + 2090 s^5 + 3163 s^4

                                + 3213 s^3 + 2014 s^2 + 738 s + 120
>> bode(sys),grid
```

The resulting Bode plot (Fig. 4.19) shows a gain margin of about 20 dB and an infinite phase margin, which indicates an excellent robustness. Thus, the observer based airspeed and altitude hold system is fairly robust in addition to its good performance.

4.5.1.2 Glideslope Tracking and Automatic Landing System

For aircraft operation in poor visibility conditions, it is necessary to have an autopilot that can follow a glideslope signal from a ground radio beacon in order to carry out what is called an *instrument landing system* (ILS) approach. After reaching a set height (decision height) above the runway, the pilot can either perform the landing manually, or engage an *automatic landing system* to make a smooth touchdown. The longitudinal part of the glideslope tracking and automatic landing is carried out with elevator input, as the engine is set to a near constant thrust level. In any case, power adjustments with the modern jet engines have a large time lag, and thus cannot be relied upon in correcting small glideslope errors. Thus elevator is the only longitudinal control input. We shall assume the following that a lateral autopilot keeps the vehicle aligned with the runway centerline throughout the maneuver.

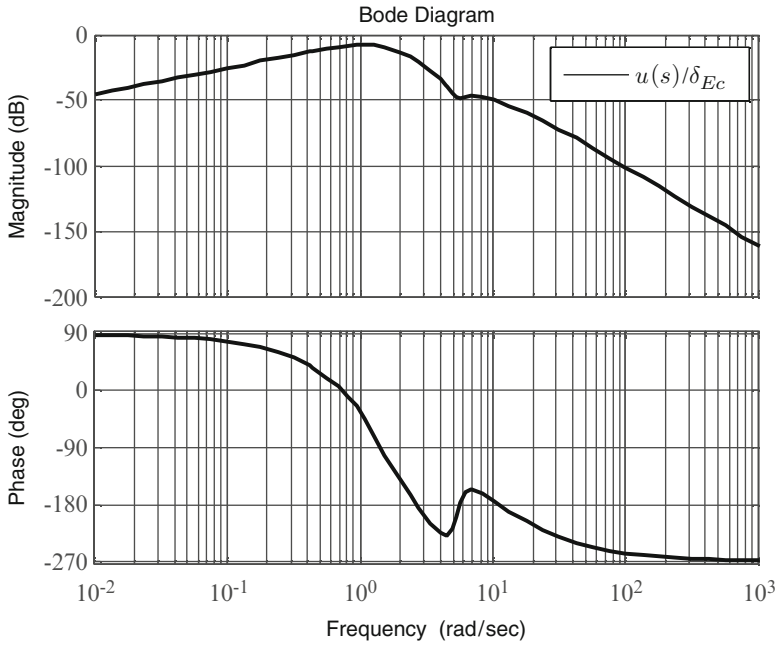


Fig. 4.19 Frequency response of the closed-loop transfer function, $u(s)/\delta_{\mathrm{Ec}}(s)$, of an observer based altitude and airspeed hold autopilot

For following a small, negative, glideslope angle that is known apriori (generally $\Phi_{\mathrm{e}} = -3°$), we require an onboard radio receiver to intercept the glideslope radio signal and follow it to the decision height. Thus the radio sensor provides a feedback of the pitch angle error, θ, to the autopilot. The plant for ILS approach must have $\Theta_{\mathrm{e}} = \Phi_{\mathrm{e}}$ in the coefficient matrix, A, of (4.144) through the elements a_{13} and a_{43}, which the pilot can set by a knob. Hence the autopilot is a set-point regulator for maintaining the state vector, $\mathbf{x} = (u, \Delta h, \alpha, \theta, q)^{\mathrm{T}} = \mathbf{0}$, with the given equilibrium state, and its design can be carried out in much the same way as in the case of altitude and airspeed hold autopilot. The observer can be based on pitch angle feedback from the glideslope sensor, along with airspeed measurement, for a good control of the glideslope.

The automatic landing system requires a precise measurement of the geometric altitude, which can only be possible with a radar altimeter that works by reflecting radio waves off the ground. The airspeed is continuously decreasing during the flight maneuver, which consists of changing the flight path from a glide to one nearly parallel with the ground, such that the touchdown is made at a speed about 10% above the *stalling speed* [21], U_{stall}. In effect, one is primarily interested in simultaneously achieving $h = 0$, $U = 1.1\, U_{\mathrm{stall}}$ while making the sink rate, $\dot{h} = U \sin \Phi$, a small, negative number (about 100 ft/min) that allows a firm contact with the ground without bouncing. Of course, $\Theta_{\mathrm{e}} = 0$ and $U = 1.1\, U_{\mathrm{stall}}$ are set as references in the coefficient matrices, A and B (4.144) for the regulator of the automatic landing plant.

Example 4.10. For the airplane of Example 4.9, design an automatic ILS approach system with a glideslope signal of $-3°$ and approach speed, $U = 55$ m/s. Assume the engines are already throttled back to a constant thrust for a steady glide of the required angle. Simulate the closed-loop response to an initial glideslope error of 3° at level interception of the glideslope signal.

The longitudinal plant for automatic ILS approach is modified by removing the altitude state equation, putting $\Theta_e = -3°$, and interchanging the second and third rows (and columns) in the matrices A and B:

```
>> th=-3*pi/180 % glideslope angle
>> A=[-0.045  -9.81*cos(th)           1.96    0;
          0              0              0    1;
     -0.0065 -9.81*sin(th)/55     -1.9636    1;
      0.0059  0.9*9.81*sin(th)/55 -6.8327 -2.9]

>> B = [0    0  0.3     0.1243]';

A =
   -0.0450   -9.7966    1.9600         0
         0         0         0    1.0000
   -0.0065    0.0093   -1.9636    1.0000
    0.0059   -0.0084   -6.8327   -2.9000

>> damp(A)

        Eigenvalue            Damping     Freq. (rad/s)
 -2.43e+000 + 2.57e+000i     6.88e-001      3.54e+000
 -2.43e+000 - 2.57e+000i     6.88e-001      3.54e+000
 -2.05e-002 + 2.09e-001i     9.75e-002      2.10e-001
 -2.05e-002 - 2.09e-001i     9.75e-002      2.10e-001
```

Note that there is little change in the natural frequencies, although the phugoid damping is increased by about 20% when compared with the level flight plant. We now repeat the design steps of Example 4.9, except for the altitude output, Δh, is replaced by glideslope angle error, θ, for the observer:

```
>> K=acker(A,B,[-0.8-0.8i,-0.8+0.8i,-2.5+2.5i,-2.5-2.5i]) %regulator gains

K =    0.8047  -13.6730    6.4894   -2.0549

>> A11=A(1:2,1:2)
A11 =
   -0.0450   -9.7966
         0         0

>> A12=A(1:2,3:4)
A12 =
    1.9600         0
         0    1.0000

>> A21=A(3:4,1:2)
A21 =
      -0.0065    0.0093348
       0.0059   -0.0084013

>> A22=A(3:4,3:4)
A22 =
      -1.9636            1
      -6.8327         -2.9

>> B1=B(1:2,:)
B1 =
     0
     0
```

```
>> B2=B(3:4,:)
B2 =
    0.3000
    0.1243

% Observer gain matrix, L, by eigenstructure assignment:
>> Lp=place(A22',A12',[-0.5-0.5i,-0.5+0.5i]);L=Lp'
L =
     -0.74673          0.5
       -3.231         -2.4

>> F=A22-L*A12
F =
   -0.5000    0.5000
   -0.5000   -0.5000

>> G=F*L+A21-L*A11
G =
      -1.2822      -8.7561
       1.8494      -30.711

>> H=B2-L*B1
H =
    0.3000
    0.1243

>> K1=K(1:2)
K1 =    0.8047  -13.6730

>> K2=K(3:4)
K2 =    6.4894   -2.0549

>> a=15; % servo pole
>> Abar=[A B;zeros(1,4) -a]; % Plant augmented with servo
>> Bbar=[zeros(size(B)); a];
>> Cbar=[1 0 0 0 0;0 1 0 0 0];

% Closed-loop dynamics matrix:
>> Ac=[Abar-Bbar*(K1+K2*L)*Cbar  -Bbar*K2;(G-H*(K1+K2*L))*Cbar F-H*K2]

Ac =
       -0.045      -9.7966      1.96      0        0         0          0
            0            0         0      1        0         0          0
      -0.0065    0.0093348   -1.9636      1      0.3         0          0
       0.0059   -0.0084013   -6.8327   -2.9   0.1243         0          0
      -38.974       82.446         0      0      -15   -97.341     30.824
      -2.0617      -7.1072         0      0        0   -2.4468     1.1165
       1.5264      -30.028         0      0        0   -1.3066   -0.24457

>> damp(Ac)

        Eigenvalue            Damping     Freq. (rad/s)
 -1.51e+001                  1.00e+000      1.51e+001
 -2.52e+000 + 2.51e+000i     7.09e-001      3.56e+000
 -2.52e+000 - 2.51e+000i     7.09e-001      3.56e+000
 -6.93e-001 + 7.85e-001i     6.62e-001      1.05e+000
 -6.93e-001 - 7.85e-001i     6.62e-001      1.05e+000
 -5.54e-001 + 5.17e-001i     7.31e-001      7.57e-001
 -5.54e-001 - 5.17e-001i     7.31e-001      7.57e-001

% closed-loop initial response:
>> sys=ss(Ac,zeros(7,1),zeros(1, 7),0);
>> [y,t,x]=initial(sys,[0  3*pi/180 zeros(1,5)]',15);
>> u=-x(:,1:2)*(K1+K2*L)'-x(:,6:7)*K2';
```

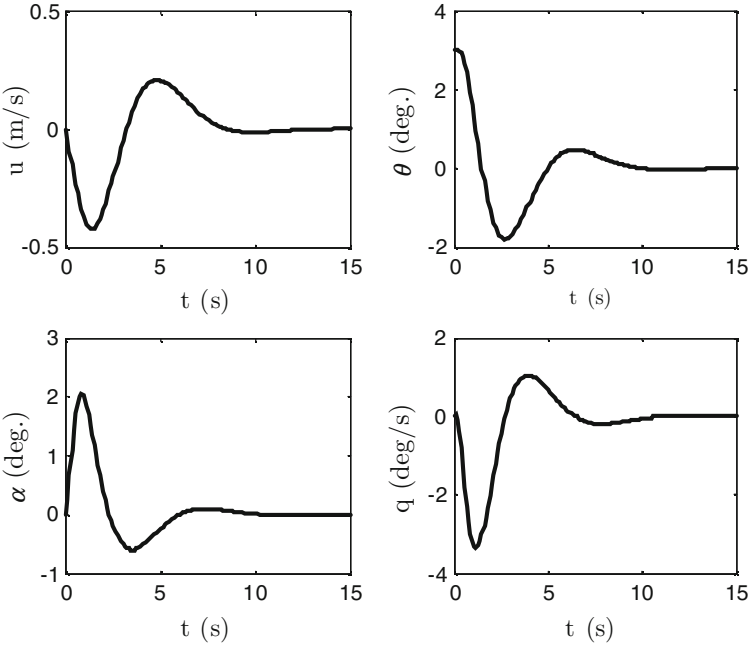


Fig. 4.20 Closed-loop initial response of an observer based, automatic instrument landing system (ILS) for an initial glideslope error of 3°

The initial response is plotted in Figs. 4.20 and 4.21. Note that the closed-loop transients settle in about 8 s, with the maximum airspeed deviation less than ±0.5 m/s, maximum glideslope undershoot not exceeding −2°, angle-of-attack overshoot within ±2°, and maximum transient pitch rate below ±4°/s. The elevator deflection required is well within ±25° limits (Fig. 4.21).

4.5.1.3 Gain Scheduling for Pitch Control

The dimensional stability derivatives vary with the flight dynamic pressure. Many high-speed aircraft also experience a reduction in short-period damping at transonic and supersonic speeds due to the aft movement of the wing–fuselage aerodynamic center as the speed of sound is crossed. In order to provide adequate damping in pitch throughout the flight envelope, the autopilot must schedule its feedback gains with the Mach number and dynamic pressure. By taking the nondimensional plant into account, one can make nondimensional gains independent of the dynamic pressure, which are merely multiplied by $\bar{q}$ in order to produce the dimensional gains at any flight condition. However, Mach number scheduling is still necessary, as we shall illustrate in the following example.

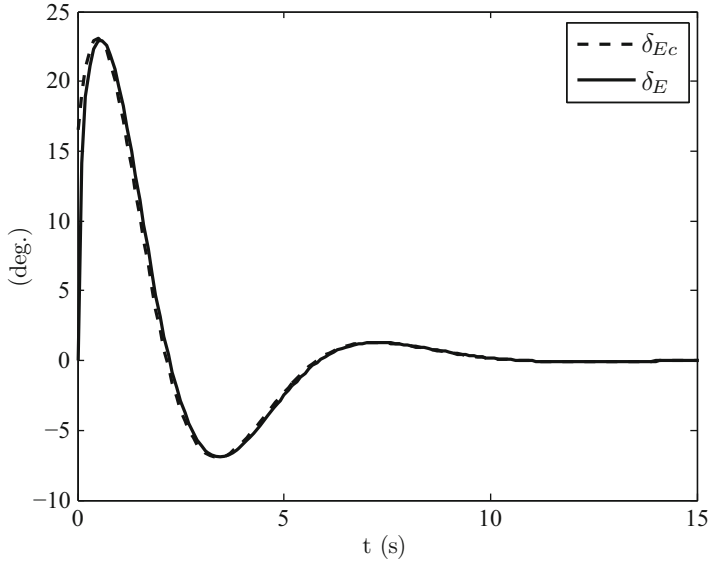


Fig. 4.21 Closed-loop elevator command and response of an observer based, automatic instrument landing system (ILS) for an initial glideslope error of 3°

Table 4.1 Variation of longitudinal stability derivatives with the mach number

$\mathcal{M}$	$C_{z\alpha}$	$C_{\mathrm{m}\alpha}$	$C_{\mathrm{m}\dot{\alpha}}$	C_{mq}	$C_{z\delta}$	$C_{\mathrm{m}\delta}$
0.3	−3.8	−0.45	−0.031	−0.06	0.58	0.8
0.6	−4.2	−0.42	−0.017	−0.031	0.59	0.82
0.9	−5.1	−0.55	−0.011	−0.022	0.54	0.79
1.1	−4.5	−0.98	−0.008	−0.018	0.39	0.58
1.5	−3.5	−0.41	−0.006	−0.015	0.35	0.53

Example 4.11. Consider a fighter aircraft with $m = 8{,}000$ kg, $S = 30$ m^2, $\bar{c} = 3.5$m, $J_{yy} = 4.8 \times 10^4$ kg m^2, $C_{z\dot{\alpha}} = C_{zq} = 0$. The variation of the aircraft's nondimensional stability derivatives with Mach number is described in Table 4.1. Design a stability augmentation system (SAS) with center of mass normal acceleration and pitch rate outputs so as to provide a short-period damping ratio of 0.707 in a level flight within the given range of Mach numbers at $h = 10$ km, where the density is $\rho = 0.41362$ kg/m^3, and the speed of sound is 299.97 m/s. Plot the required feedback gains as functions of the Mach number.

In order to design the regulator, we consider the following equation for the normal acceleration at the center of mass, which is obtained with the approximation $Z_{\dot{\alpha}} = Z_q = 0$:

$$a_z = U\,(\dot{\alpha} - q) = \frac{Z_\alpha}{m}\alpha + \frac{Z_\delta}{m}\delta_{\mathrm{E}}$$

Thus, with the state variables, $x_1 = \alpha$, $x_2 = q$, input, δ_E, and outputs $y_1 = a_z$, $y_2 = q$, we have the following output coefficient matrices:

$$\mathsf{C} = \begin{pmatrix} \frac{Z_\alpha}{m} & 0 \\ 0 & 1 \end{pmatrix}; \quad \mathsf{D} = \begin{pmatrix} \frac{Z_\delta}{m} \\ 0 \end{pmatrix};$$

With the state feedback control law

$$\delta_E = -\mathsf{K}\mathbf{x},$$

we write

$$\mathbf{y} = (\mathsf{C} - \mathsf{D}\mathsf{K})\mathbf{x},$$

or

$$\delta_E = -\mathsf{K}(\mathsf{C} - \mathsf{D}\mathsf{K})^{-1}\mathbf{y} = -k_z a_z - k_q q,$$

which is the output feedback control law. Hence, we can directly measure and feedback the outputs, rather than the state variables.

We begin by writing a MATLAB code called *stabilitydata.m* given in Table 4.2 for calculating the approximate short-period dynamics plant (dimensional) matrices, A, B from interpolation of the data provided in Table 4.1. The plot of the plant's natural frequency and damping ratio (Fig. 4.22) shows that the short-period damping ratio is never higher than 0.13 and drops sharply in the transonic range to about 0.08.

The output feedback gains, k_z, k_q, are computed and plotted in Fig. 4.23. Such a variation of gains with Mach number can be easily stored onboard and used at any given speed by interpolation. Of course, in order to take altitude variations into account, a two-dimensional table would need to be pre-computed and stored with a standard atmospheric model [21].

```
>> Mach=0.3:0.05:1.5;
>> n=size(Mach,2);
>> for i=1:n
       [A,B,C,D]=stabilitydata(Mach(i));
       k=acker(A,B,[-2.5-2.5i,-2.5+2.5i]);
       K(i,:)=k*inv(C-D*k);
   end
```

4.5.2 Airspeed Control by Throttle Input

While we have previously considered airspeed control in level flight by elevator input, we shall now focus on the engine throttle as the sole control input. Airspeed control is generally based on airspeed feedback because of the low frequency of the concerned phugoid mode, which is unaffected by the higher frequencies of a pitch (or normal acceleration) feedback system intended for controlling the short-period

Table 4.2 Program *stabilitydata.m* for generating the approximate short-period dynamics of a fighter aircraft in level flight as a function of Mach number

```
% (c) 2010 Ashish Tewari
function [A,B,C,D]=stabilitydata(Mach)
m=8000; % mass (kg)
S=30; % wing planform area (m^2)
c=3.5; % wing mean aerodynamic chord (m)
rho=0.41362; % density at 10 km altitude (kg/m^3)
a=299.97; % speed of sound at 10 km altitude (m/s)
v=Mach*a; % true airspeed (m/s)
Jyy=4.8e4; % pitch moment of inertia (kg\,m^2)
mu=4*m/(rho*S*c); %nondimensional mass
jy=8*Jyy/(rho*S*c^3);%nondimensional moment of inertia
M=[0.3 0.6 0.9 1.1 1.5]; % Mach numbers for interpolation

% Stability & control derivatives data at given Mach Nos:
Cza=-[3.8 4.2 5.1 4.5 3.5];
Cma=-[.45 .42 .55 .98 .41];
Cmad=-[.031 .017 .011 .008 .006];
Cmq=-[.06 .031 .022 .018 .015];
Czdelta=[.58 .59 .54 .39 .35];
Cmdelta=[.8 .82 .79 .58 .53];

% Stability & control derivatives at given flight point:
cza=interp1(M,Cza,Mach);
cma=interp1(M,Cma,Mach);
cmad=interp1(M,Cmad,Mach);
cmq=interp1(M,Cmq,Mach);
czdelta=interp1(M,Czdelta,Mach);
cmdelta=interp1(M,Cmdelta,Mach);

% Dimensional state space coefficient matrices for
% approximate short-period mode:
A=[cza/mu 1; cma/jy+cmad*cza/(mu*jy) (cmq+cmad)/jy]*2*v/c;
B=[czdelta/mu; cmdelta/jy+cmad*czdelta/(mu*jy)]*2*v/c;
C=[cza*2*v^2/(mu*c) 0;0 1];
D=[czdelta*2*v^2/(mu*c); 0];
```

mode. Since damping of the phugoid mode is largely governed by the axial force derivative, C_{xu}, the engine throttle is the primary control input for affecting the axial force, thereby actively damping out the phugoid mode.

4.5.2.1 Approximate Phugoid Plant

The airspeed control system by throttle can be designed by ignoring the short-period mode in the following phugoid approximation ($\alpha \simeq 0, M \simeq 0$) of the longitudinal dynamics:

$$\begin{aligned} m\dot{u} &\simeq X_u u + X_q q - mg\theta\cos\Theta_\mathrm{e} + X_T\beta_\mathrm{T} \\ 0 &\simeq Z_u u + (mU + Z_q)q - mg\theta\sin\Theta_\mathrm{e} + Z_T\beta_\mathrm{T} \\ \dot{\theta} &= q, \end{aligned} \tag{4.151}$$

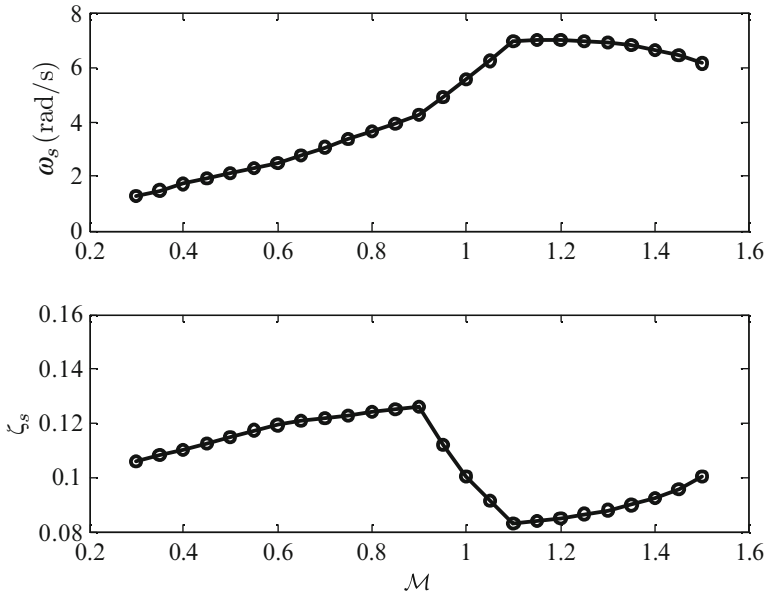


Fig. 4.22 Approximate short-period dynamics of a fighter aircraft without closed-loop stability augmentation

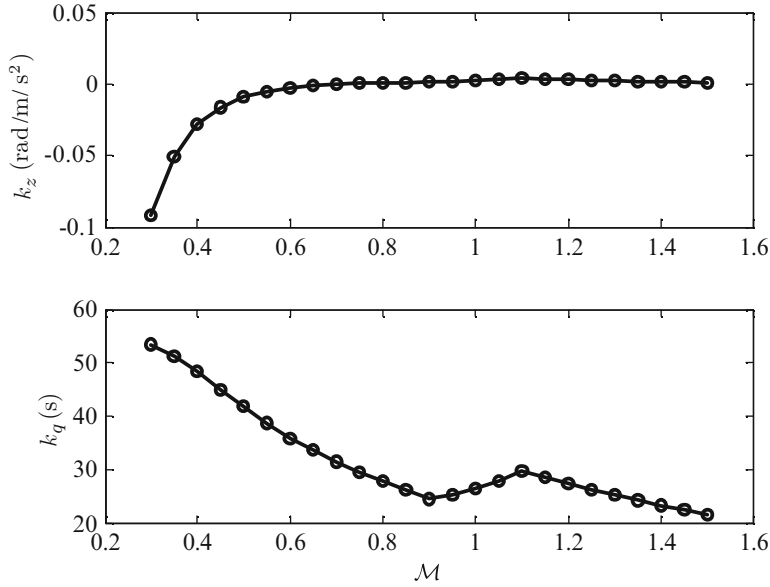


Fig. 4.23 Output feedback gains of a fighter aircraft with closed-loop stability augmentation

or

$$\begin{Bmatrix} \dot{u} \\ \dot{\theta} \end{Bmatrix} \simeq \begin{pmatrix} \frac{X_u}{m} & -g\cos\Theta_\mathrm{e} \\ -\frac{Z_u}{mU+Z_q} & \frac{mg\sin\Theta_\mathrm{e}}{mU+Z_q} \end{pmatrix} \begin{Bmatrix} u \\ \theta \end{Bmatrix} + \begin{pmatrix} \frac{X_T}{m} \\ -\frac{Z_T}{mU+Z_q} \end{pmatrix} \beta_\mathrm{T}. \tag{4.152}$$

Thus, the transfer function, $u(s)/\beta_\mathrm{T}(s)$, of the approximate phugoid mode is as follows:

$$G(s) = \frac{u(s)}{\beta_\mathrm{T}(s)} = \frac{as+b}{s^2 + 2\zeta_\mathrm{p}\omega_\mathrm{p}s + \omega_\mathrm{p}^2}, \tag{4.153}$$

where

$$a = \frac{X_T}{m}, \qquad b = \frac{Z_T g\cos\Theta_\mathrm{e} - X_T g\sin\Theta_\mathrm{e}}{mU + Z_q}, \tag{4.154}$$

with the approximate phugoid natural frequency and damping ratio given by[9]

$$\omega_\mathrm{p} = \sqrt{\frac{g(X_u\sin\Theta_\mathrm{e} - Z_u\cos\Theta_\mathrm{e})}{mU + Z_q}}, \tag{4.155}$$

$$\zeta_\mathrm{p} = -\frac{\left[\frac{X_u}{m}\left(mU + Z_q\right) + mg\sin\Theta_\mathrm{e}\right]}{2\sqrt{g(X_u\sin\Theta_\mathrm{e} - Z_u\cos\Theta_\mathrm{e})(mU + Z_q)}}. \tag{4.156}$$

Unfortunately, the approximate natural frequency and damping ratio given by (4.155) and (4.156) are seldom accurate enough to be used in control system design. In order to have a better frequency domain approximation of the phugoid mode, we

[9] A further approximation, called *Lanchester's model*, can be obtained in the incompressible flow limit:

$$Z_u \simeq -\rho USC_\mathrm{Le} = -\frac{2mg}{U},$$

and $Z_q \ll mU$, resulting in

$$\omega_\mathrm{p} \simeq \frac{g\sqrt{2}}{U}; \qquad \zeta_\mathrm{p} \simeq -\frac{X_u U}{2\sqrt{2}mg}.$$

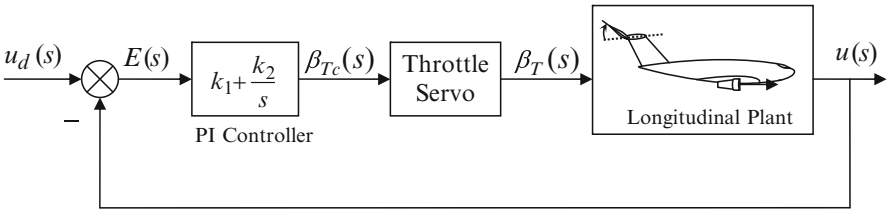


Fig. 4.24 A typical airspeed control system for an aircraft

shall factor the longitudinal transfer function as follows; noting that the numerator polynomial effectively cancels out the short-period denominator quadratic in the low frequency range:

$$G(s) = \frac{u(s)}{\beta_{\mathrm{T}}(s)} = \frac{as\left(s^2 + 2\zeta_{\mathrm{s}}\omega_{\mathrm{s}}s + \omega_{\mathrm{s}}^2\right) + c}{\left(s^2 + 2\zeta_{\mathrm{s}}\omega_{\mathrm{s}}s + \omega_{\mathrm{s}}^2\right)\left(s^2 + 2\zeta_{\mathrm{p}}\omega_{\mathrm{p}}s + \omega_{\mathrm{p}}^2\right)}, \tag{4.157}$$

where a, c are constants and the natural frequencies, $\omega_{\mathrm{p}}, \omega_{\mathrm{s}}$, and damping ratios, $\zeta_{\mathrm{p}}, \zeta_{\mathrm{s}}$, are the actual values of the longitudinal plant. At low frequencies, the short-period dynamics can be approximated by the DC gain,

$$b \simeq \lim_{s\to 0} \frac{c}{s^2 + 2\zeta_{\mathrm{s}}\omega_{\mathrm{s}}s + \omega_{\mathrm{s}}^2} = \frac{c}{\omega_{\mathrm{s}}^2}, \tag{4.158}$$

which, substituted into (4.157) produces the following phugoid approximation:

$$G(s) = \frac{u(s)}{\beta_{\mathrm{T}}(s)} \simeq \frac{as + b}{s^2 + 2\zeta_{\mathrm{p}}\omega_{\mathrm{p}}s + \omega_{\mathrm{p}}^2}. \tag{4.159}$$

4.5.2.2 PI Control of Airspeed by Throttle Input

The presence of the zero, $s = -b/a$, in the approximate phugoid plant indicates that proportional–integral (PI) control can provide an increased closed-loop damping and natural frequency without the need of derivative action, while having a zero steady-state error to a step airspeed command. However, since $b < 0$ for a conventional aircraft, the right-half plane zero leads to a non-minimum phase plant [20] that is problematic from design point of view due to transient response being of opposite sign compared to the input, as well as a reduced closed-loop gain margin. Hence, special attention is to be paid for airspeed controller design. Block diagram of a typical airspeed control system using airspeed feedback and throttle input with PI control is shown in Fig. 4.24. The PI controller transfer function is given by

$$H(s) = \frac{\beta_{\mathrm{Tc}}(s)}{E(s)} = k_1 + \frac{k_2}{s}, \tag{4.160}$$

where $E(s) = u_d(s) - u(s)$, is the airspeed error and (k_1, k_2) are controller gains that are selected by equating the actual and desired closed-loop, third degree characteristic polynomials,

$$1 + G(s)H(s) = s^3 + (2\zeta_p\omega_p + ak_1)s^2 + \left(\omega_p^2 + ak_2 + bk_1\right)s = (s+\lambda)(s^2 + 2\zeta\omega s + \omega^2),$$

resulting in

$$\begin{aligned} \lambda\omega^2 &= bk_2 \\ \lambda + 2\zeta\omega &= ak_1 + 2\zeta_p\omega_p \\ \omega^2 + 2\zeta\omega\lambda &= \omega_p^2 + bk_1 + ak_2. \end{aligned} \tag{4.161}$$

Clearly, only two of the three closed-loop parameters, (λ, ζ, ω), are uniquely determined by the gains (k_1, k_2). This imposes a limit on the desired closed-loop response.

Example 4.12. The airplane of Example 4.4 has a single turboprop engine with the following additional parameters at standard sea level with a constant speed, $U = 55$ m/s:

$$\frac{X_T}{m} = 0.2452 \text{ m/s}^2/\text{deg.}; \qquad \frac{Z_T}{m} \simeq 0; \qquad \frac{M_T}{J_{yy}} \simeq 0.$$

The engine is driven by a throttle servo of approximate transfer function,

$$\frac{\beta_T(s)}{\beta_{Tc}(s)} = \frac{5}{s+5}.$$

Design a closed-loop airspeed control system for the design flight condition of steady climb with $\Theta_e = 15°$ and $U = 55$ m/s at standard sea level such that the settling-time is reduced as much as possible from that of the plant. The maximum throttle deflection required for a step airspeed change of 1 m/s should be about $\pm 10°$ and must not result in speed overshoot/undershoot greater than ± 1 m/s. Simulate the closed-loop step response.

We begin by computing the longitudinal state space coefficient matrices with throttle input and airspeed output to be the following:

```
>> sysG=ss(A,B,[1 0 0 0],0)% (A,B) pre-computed and stored

A =
             x1         x2         x3         x4
   x1    -0.045       1.96     -9.476          0
   x2  -0.006545     -1.964   -0.04616          1
   x3         0          0          0          1
   x4   0.005891     -6.833    0.04155      -2.9

B =
          u1
   x1  0.2452
   x2       0
   x3       0
   x4       0
```

```
C =
       x1  x2  x3  x4
   y1   1   0   0   0

D =
       u1
   y1   0

Continuous-time model.
```

The longitudinal natural frequencies and damping ratios in the climb condition are obtained as follows:

```
>> damp(A)

        Eigenvalue            Damping     Freq. (rad/s)

 -1.84e-003 + 2.02e-001i     9.11e-003      2.02e-001
 -1.84e-003 - 2.02e-001i     9.11e-003      2.02e-001
 -2.45e+000 + 2.58e+000i     6.89e-001      3.56e+000
 -2.45e+000 - 2.58e+000i     6.89e-001      3.56e+000
```

Thus, $\omega_p = 0.2018$ rad/s, $\zeta_p = 0.00911$, $\omega_s = 3.5579$ rad/s, and $\zeta_s = 0.6893$. The phugoid mode has a settling-time of $t_s \simeq 4/(0.2018 \times 0.00911) = 2,175.8$ s.

Next, the transfer function of the complete longitudinal dynamics from the throttle input to airspeed output of the aircraft is obtained as follows:

```
>> tf(sysG) % TF  u(s)/beta_T(s)

Transfer function:
  0.2452 s^3 + 1.193 s^2 + 3.062 s - 0.09736
-----------------------------------------------
s^4 + 4.909 s^3 + 12.72 s^2 + 0.2463 s + 0.5155
```

The approximate phugoid plant transfer function is derived as follows:

```
>> a= 0.2452; b=-0.09736/3.56^2

b =   -0.007682

>> wp=0.2018; zp=0.00911;
>> sysG1=tf([a b],[1 2*zp*wp wp^2]) % Approx. phugoid plant TF

Transfer function:
   0.2452 s - 0.007682
--------------------------
s^2 + 0.003679 s + 0.04073
```

The frequency response (Bode diagram) and step response to 1° throttle input of the full longitudinal and approximate transfer functions are compared in Fig. 4.25 and 4.26, respectively. Note the excellent agreement between the two in the given frequency range.

```
>> bode(sysG,logspace(-2,1)) % Bode diagram of longitudinal plant
>> step(sysG) % Step response to 1 deg. throttle movement
```

The Bode diagram (Fig. 4.25) shows the single natural frequency 0.202 rad/s corresponding to the phugoid mode and confirms that the throttle input does not excite the short-period mode (the peak corresponding to which is absent). The unit step response of the airspeed plant to a 1° throttle input is plotted in Fig. 4.26 for the first 120 s showing the phugoid behavior and an absent short-period mode. Thus,

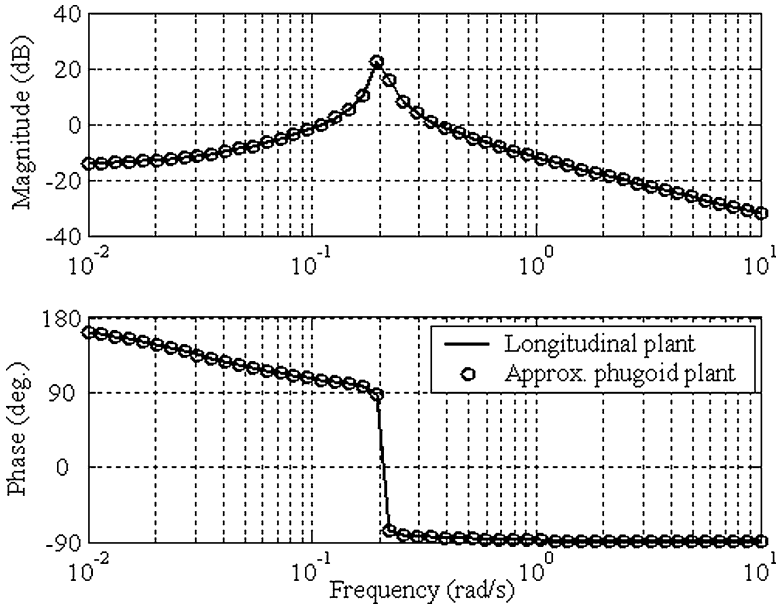


Fig. 4.25 Bode diagram for the transfer function, $u(s)/\beta_T(s)$, for a turboprop engined aircraft in a steady climb with exact and approximate (phugoid) dynamics

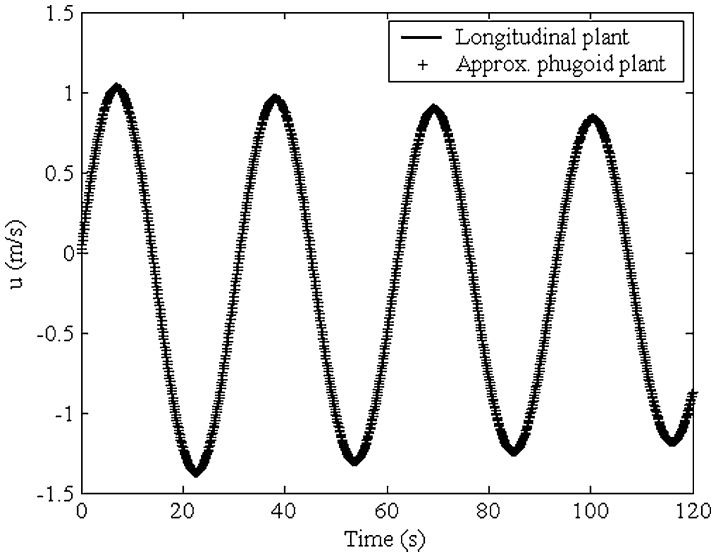


Fig. 4.26 Step response of the airspeed plant, $u(s)/\beta_T(s)$, to 1° displacement of throttle with exact and approximate (phugoid) dynamics

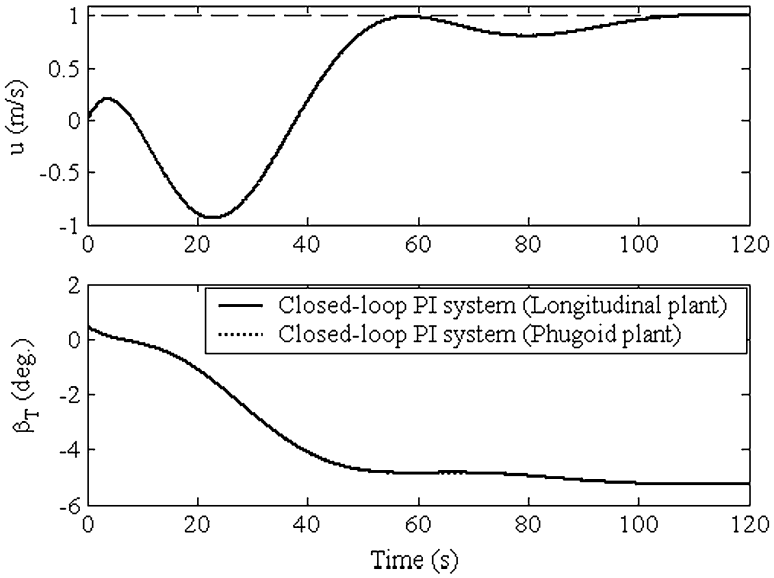


Fig. 4.27 Airspeed and throttle step responses of the closed-loop PI airspeed control system, to a 1 m/s commanded airspeed change with exact and approximate (phugoid) dynamics

the fourth-order longitudinal transfer function essentially behaves as a second-order one, thereby validating the phugoid approximation used for designing the airspeed control system.

A PI controller is constructed with $k_1 = 0.475, k_2 = -0.08$, after some trial and error in order to meet the design objectives. The servo dynamics – being of a much higher frequency than controller bandwidth – is safely neglected during controller design. The closed-loop systems with full longitudinal plant and approximate phugoid plant are constructed with PI controller and throttle servo and their step (Fig. 4.27) and frequency (Fig. 4.28) responses are compared using the following MATLAB's Control Systems Toolbox (CST) commands:

```
>> sysH=tf([0.475 -0.08],[1 0]) % PI controller

Transfer function:
0.475 s - 0.08
--------------
      s

>> sysG1=series(tf(5,[1 5]),sysG1)%Approx. phugoid plant & throttle servo

Transfer function:
         1.226 s - 0.03841
-------------------------------------
s^3 + 5.004 s^2 + 0.05912 s + 0.2036

>> sysOL=series(sysH,sysG1)%open-loop system with approx. phugoid plant

Transfer function:
    0.5824 s^2 - 0.1163 s + 0.003073
```

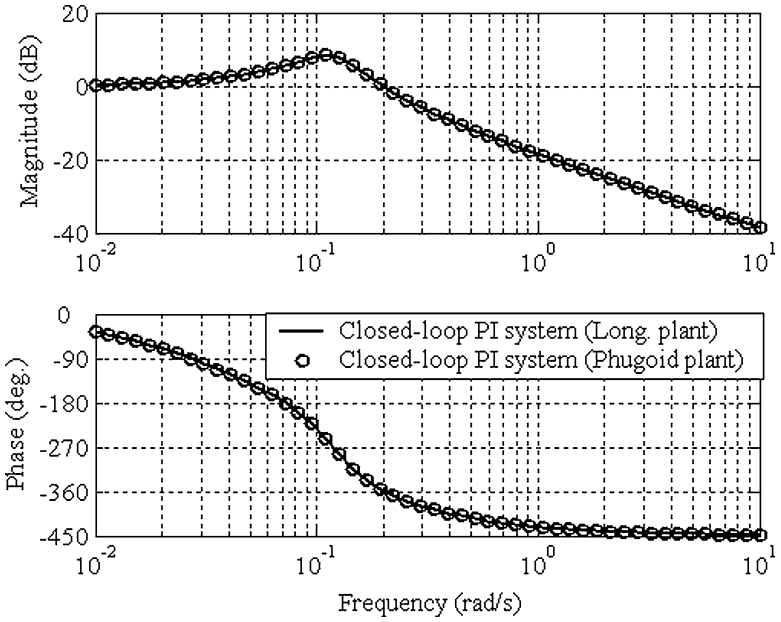


Fig. 4.28 Bode diagrams of the closed-loop PI airspeed control system, $u(s)/u_d(s)$, with exact and approximate (phugoid) dynamics

```
-----------------------------------------
s^4 + 5.004 s^3 + 0.05912 s^2 + 0.2036 s

>> sys1=feedback(sysOL,1)%closed-loop system with approx. phugoid plant

Transfer function:
         0.5824 s^2 - 0.1163 s + 0.003073
-----------------------------------------------------
s^4 + 5.004 s^3 + 0.6415 s^2 + 0.08729 s + 0.003073

>> sysG=series(tf(5,[1 5]),sysG)%Full longitudinal plant & throttle servo

Transfer function:
        1.226 s^3 + 5.963 s^2 + 15.31 s - 0.4868
-----------------------------------------------------
s^5 + 9.909 s^4 + 37.26 s^3 + 63.83 s^2 + 1.747 s + 2.578

>> sysOL=series(sysH,sysG)%open-loop system with full longitudinal plant

Transfer function:
   0.5824 s^4 + 2.735 s^3 + 6.795 s^2 - 1.456 s + 0.03894
-------------------------------------------------------------
s^6 + 9.909 s^5 + 37.26 s^4 + 63.84 s^3 + 1.747 s^2 + 2.578 s

>> sys=feedback(sysOL,1)%closed-loop system with full longitudinal plant

Transfer function:
        0.5824 s^4 + 2.735 s^3 + 6.795 s^2 - 1.456 s + 0.03894
-----------------------------------------------------------------------
s^6 + 9.909 s^5 + 37.85 s^4 + 66.57 s^3 + 8.542 s^2 + 1.122 s + 0.03894
```

```
>> [y,t]=step(sys,0:0.1:120); % step response to unit airspeed command
>> [y1,t1]=step(sys1,0:0.1:120);
>> sysu1=feedback(sysH,sysG1)% closed-loop beta_T(s)/u(s) for phugoid

Transfer function:
0.475 s^4 + 2.297 s^3 - 0.3722 s^2 + 0.09199 s - 0.01629
---------------------------------------------------------
  s^4 + 5.004 s^3 + 0.6415 s^2 + 0.08729 s + 0.003073

>> sysu=feedback(sysH,sysG)% closed-loop beta_T(s)/u(s) for long. dynamics

Transfer function:
0.475 s^6 + 4.627 s^5 + 16.91 s^4 + 27.34 s^3 - 4.277 s^2 + 1.085 s - 0.2062
-----------------------------------------------------------------------------
  s^6 + 9.909 s^5 + 37.85 s^4 + 66.57 s^3 + 8.542 s^2 + 1.122 s + 0.03894

>> [u,t]=step(sysu,0:0.1:180);%CL throttle response to unit airspeed command
>> [u1,t1]=step(sysu1,0:0.1:180);
```

Note the undershoot (Fig. 4.27) in the airspeed response due to non-minimum phase character of the plant, which is unavoidable. While the transient and throttle input magnitude limits are met (there is a steady-state value of $\beta_T = -5.05°$), and the settling-time is reduced to about 105 s, the transient response is still quite slow. Figure 4.28 indicates a closed-loop gain margin of only about 5 dB and a phase margin of 180°. Thus, a moderate variation of the equilibrium flight speed and climb angle from the design values may destabilize the PI airspeed control system due to the inadequate gain margin.

Example 4.12 illustrated the limitations of the classical, single input, transfer function approach in modern control design. Since then, the initial airspeed undershoot is caused by an increase in the pitch angle, θ, as the throttle input is applied. It is likely that the transient behavior can be improved when throttle input is accompanied by an appropriate elevator control that prevents the pitch angle from building up. However, such a multi-input controller design is not easily carried out with the transfer function approach and instead requires the time domain, state space design methodology, which we consider in the next section.

4.6 Longitudinal Control by Elevator and Throttle

Multi-input control can significantly improve the performance and robustness of a longitudinal control system. As we saw in Chap. 3, such a control system can be designed by either eigenstructure assignment, or linear optimal control. We express the linear longitudinal plant as:

$$\dot{\mathbf{x}} = \mathsf{A}\mathbf{x} + \mathsf{B}\mathbf{u}, \tag{4.162}$$

with the state vector, $\mathbf{x} = (u, \alpha, \theta, q)^T$, the control input, $\mathbf{u} = (\delta_E, \beta_T)^T$, and the state space coefficient matrices, $\mathsf{A} = \mathsf{A}_{Long}$ and $\mathsf{B} = \mathsf{B}_{Long}$.

For a constant desired state, $\mathbf{x}_d(t)$, an output feedback tracking system is to be designed with the following linear control law,

$$\mathbf{u} = \mathsf{K}_d \mathbf{x}_d + \mathsf{K}(\mathbf{x}_d - \mathbf{x}_o), \tag{4.163}$$

where regulator gain matrix, $\mathbf{K}$, is selected by a state-space design technique (eigenstructure assignment or optimal control). A linear observer is required for supplying the estimated state, $\mathbf{x_o}(t)$, to the tracking system and can be designed in a manner similar to the regulator by the separation principle. Recall the state equation of a full-order observer (Chap. 3):

$$\dot{\mathbf{x}}_o = (\mathsf{A} - \mathsf{LC})\mathbf{x}_o + (\mathsf{B} - \mathsf{LD})\mathbf{u} + \mathsf{L}\mathbf{y}, \tag{4.164}$$

where L is the observer gain matrix. The overall closed-loop control system dynamics is thus given by the state equation

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_o \end{Bmatrix} = \begin{pmatrix} \mathsf{A} & -\mathsf{BK} \\ \mathsf{LC} & \mathsf{A} - \mathsf{BK} - \mathsf{LC} \end{pmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{x}_o \end{Bmatrix} + \begin{pmatrix} \mathsf{B}(\mathsf{K} + \mathsf{K_d}) \\ \mathsf{B}(\mathsf{K} + \mathsf{K_d}) \end{pmatrix} \mathbf{x}_d. \tag{4.165}$$

Example 4.13. Consider the design of an airspeed control system with both elevator and throttle inputs and the airspeed perturbation, $u(t)$, as the sole output. We have choice to design either a full-order, or a reduced-order observer (Chap. 3).

It is necessary to have an appropriate feedforward gain matrix, $\mathsf{K_d}$, for tracking a step change in airspeed, i.e., $u_d = \text{const}$. It can be easily shown that for satisfying (3.122), we require

$$(\mathsf{A} + \mathsf{BK_d})(u_d, 0, 0, 0)^{\mathrm{T}} = \mathbf{0}, \tag{4.166}$$

or, $Z_T/m = 0$, $M_T/m = 0$. Also, if we assume $X_\delta/m = 0$, we have

$$\mathsf{K_d} = \begin{pmatrix} -Z_u/Z_\delta & 0 & 0 & 0 \\ -X_u/X_T & 0 & 0 & 0 \end{pmatrix}. \tag{4.167}$$

In order to illustrate the design process, we select the design requirements of Example 4.12 with the elevator parameters specified in Example 4.9. Additional design specifications of elevator deflection restricted to $\pm 20^\circ$ and a steady-state error smaller than $\pm 2\%$ are added for a unit step change in airspeed from the given equilibrium condition.

Substituting the data from Examples 4.9 and 4.12, we have

$$\mathsf{K_d} = \begin{pmatrix} 0.0217 & 0 & 0 & 0 \\ 0.1835 & 0 & 0 & 0 \end{pmatrix}.$$

We choose eigenstructure assignment for the following regulator poles:

$$s_{1,2} = -0.55 \pm 0.55\text{i}; \qquad s_{3,4} = -0.9 \pm 0.9\text{i}.$$

for which the MATLAB-CST command *place.m* is used to solve the algebraic Riccati equation as follows:

```
>> A = [-0.0450    1.9600   -9.4757         0
        -0.0065   -1.9636   -0.0462    1.0000
               0         0         0    1.0000
         0.0059   -6.8327    0.0415   -2.9000];

>> B = [0              0.2452
        0.3000            0
           0              0
        0.1243            0];

>> K=place(A,B,[-0.55+0.55i,-0.55-0.55i,-0.9+0.9i,-0.9-0.9i])

% Regulator gain matrix:

K =
   -0.1692   -9.0247   -0.5597    0.6508
    2.5201   43.5365  -45.8079   10.4790
```

Similarly, we design a full-order observer by eigenstructure assignments for placing the observer poles at

$$s_{1,2} = -2 \pm 2\text{i}; \qquad s_{3,4} = -2.5 \pm 2.5\text{i},$$

as follows:

```
>> C = [1     0     0     0];

>> Lp=place(A',C',[-2+2i,-2-2i,-2.5+2.5i,-2.5-2.5i]);

% Observer gain matrix (L=Lp'):

L =
    4.0914
    0.0316
   -0.8255
   -0.0497
```

Note that the observer poles are much deeper into the left-half s-plane than the poles of the regulator, which is required for a much faster observer dynamics.

The regulator, observer, and feedforward controller are assembled along with the longitudinal plant, as well as the elevator and throttle servos into the *Simulink* model shown in Fig. 4.29. The closed-loop step response is simulated with a fourth-order Runge–Kutta algorithm, and the results are plotted in Figs. 4.30 and 4.31. The angles, $\alpha, \theta, \delta_\text{E}$, and pitch-rate, q, are first converted from radian to degree before plotting, while throttle displacement, β_T is already in degree. Note the settling-time of 5 s with a maximum overshoot in the airspeed less than 0.05 m/s. This is possible due to aggressive elevator input (Fig. 4.31) that quickly brings the initial pitch-up caused by throttle input to zero without a large variation in the angle-of-attack

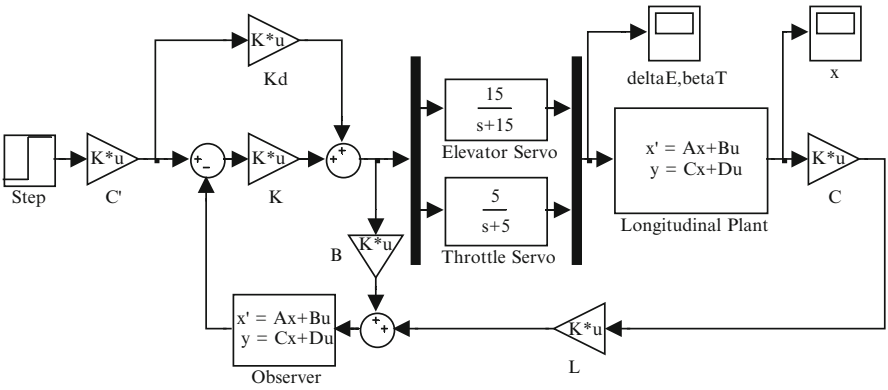


Fig. 4.29 Simulink model of the multi-input, output feedback, airspeed tracking system with a full-order observer and servo actuators for elevator and throttle

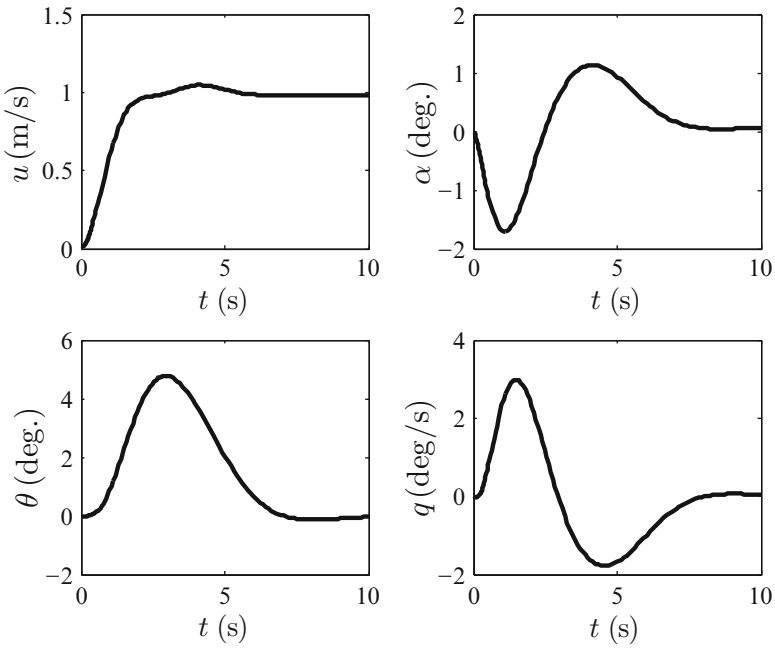


Fig. 4.30 Closed-loop simulation results (airspeed, angle-of-attack, pitch angle, and pitch rate) of an output feedback multi-input airspeed control system

(Fig. 4.30). Although there is a steady-state airspeed error of less than 0.02 m/s due to a lack of integral action in the forward path (Chap. 3), it meets the design requirement. The initially opposite trends for θ and q transients are typical of a non-minimum phase system (i.e., zeros in the right-half s-plane).

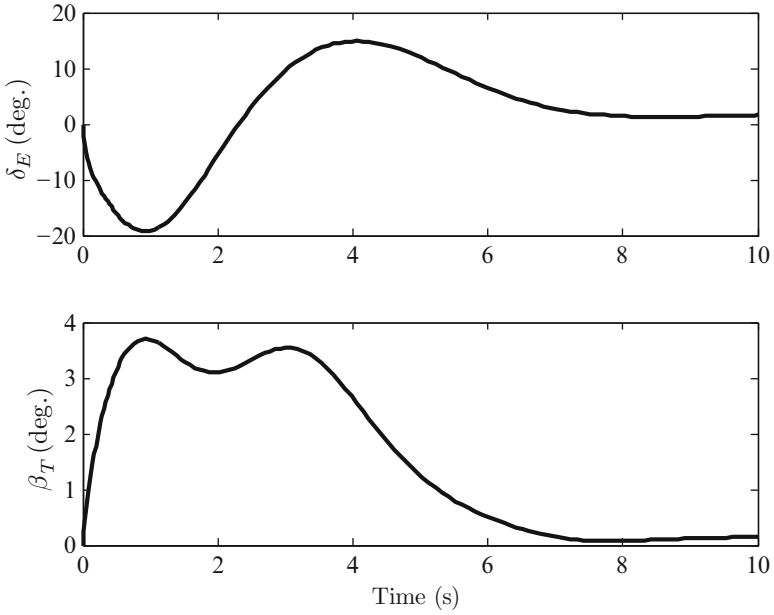


Fig. 4.31 Closed-loop simulation results (elevator and throttle displacement) of an output feedback multi-input airspeed control system

We will now construct the closed-loop system with the first-order servos as follows:

```
>> a=15; b=5; % servo poles
>> Abar=[A B;zeros(1,4) -a 0; zeros(1,4) 0 -b] % Plant dynamics matrix with
  servos

Abar =
   -0.0450    1.9600   -9.4757         0         0    0.2452
   -0.0065   -1.9636   -0.0462    1.0000    0.3000         0
         0         0         0    1.0000         0         0
    0.0059   -6.8327    0.0415   -2.9000    0.1243         0
         0         0         0         0  -15.0000         0
         0         0         0         0         0   -5.0000

>> Bbar=[zeros(size(B)); a 0; 0 b] % Augmented plant's B matrix

Bbar =
     0     0
     0     0
     0     0
     0     0
    15     0
     0     5

>> Cbar=[C 0 0] % Augmented plant's C matrix

Cbar =
     1     0     0     0     0     0

>> Kd=[0.0217 0  0  0; 0.1835 0 0 0] % Feedforward gain matrix
>> Ac=[Abar -Bbar*K; L*Cbar A-L*C-B*K] % Closed-loop dynamics matrix
```

```
Ac =

-0.0450    1.9600   -9.4757         0         0    0.2452
-0.0065   -1.9636   -0.0462    1.0000    0.3000         0
      0         0         0    1.0000         0         0
 0.0059   -6.8327    0.0415   -2.9000    0.1243         0
      0         0         0         0  -15.0000         0
      0         0         0         0         0   -5.0000
 4.0914         0         0         0         0         0
 0.0316         0         0         0         0         0
-0.8255         0         0         0         0         0
-0.0497         0         0         0         0         0

           0         0         0         0
           0         0         0         0
           0         0         0         0
           0         0         0         0
      2.5379  135.3712    8.3953   -9.7624
    -12.6004 -217.6825  229.0394  -52.3948
     -4.7543   -8.7152    1.7564   -2.5694
      0.0126    0.7438    0.1217    0.8048
      0.8255         0         0    1.0000
      0.0766   -5.7109    0.1111   -2.9809

>> damp(Ac) % Closed-loop system's eigenvalues

        Eigenvalue             Damping     Freq. (rad/s)
 -1.50e+001                  1.00e+000      1.50e+001
 -7.66e+000                  1.00e+000      7.66e+000
 -2.47e+000 + 2.57e+000i     6.93e-001      3.57e+000
 -2.47e+000 - 2.57e+000i     6.93e-001      3.57e+000
 -6.97e-001 + 2.25e+000i     2.96e-001      2.36e+000
 -6.97e-001 - 2.25e+000i     2.96e-001      2.36e+000
 -6.06e-001 + 8.92e-001i     5.62e-001      1.08e+000
 -6.06e-001 - 8.92e-001i     5.62e-001      1.08e+000
 -8.44e-001 + 2.57e-001i     9.57e-001      8.82e-001
 -8.44e-001 - 2.57e-001i     9.57e-001      8.82e-001

>> Bc=[Bbar*(K+Kd); B*(K+Kd)] % Closed-loop input coefficients matrix

Bc =
           0         0         0         0
           0         0         0         0
           0         0         0         0
           0         0         0         0
   -2.2124 -135.3712   -8.3953    9.7624
   13.5179  217.6825 -229.0394   52.3948
    0.6629   10.6752  -11.2321    2.5694
   -0.0442   -2.7074   -0.1679    0.1952
         0         0         0         0
   -0.0183   -1.1218   -0.0696    0.0809
```

The eigenvalues of the closed-loop system indicate minor changes in the regulator poles due to the addition of servo dynamics. However, the higher frequency observer poles are moved closer to the dominant poles due to servo lag, which is fortunately a more optimal position from control effort viewpoint. A shift in the observer poles can be avoided by designing the observer based upon the augmented plant (rather than the plant without servos, as we have done here). Finally, the two first-order servos now appear as a single second-order system due to feedback.

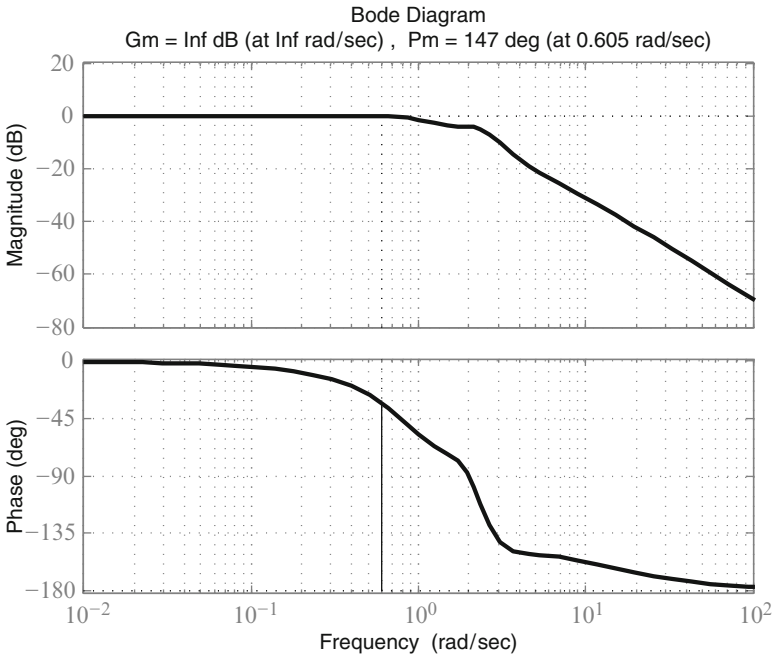


Fig. 4.32 Closed-loop frequency response, $u(s)/u_d(s)$, of an output feedback, multi-input airspeed control system with gain and phase margins

The closed-loop frequency response of $u(s)/u_d(s)$ with gain and phase margins is computed as follows and plotted in Fig. 4.32:

```
>> sys=ss(Ac,Bc(:,1),[1 zeros(1,9)],0); margin(sys),grid
```

The gain and phase margins and the respective crossover frequencies of the closed-loop airspeed frequency response can also be computed as follows:

```
>> [Gm,Pm,Wg,Wp] = margin(sys)

Gm =
   Inf

Pm =
  146.9268

Wg =
   Inf

Wp =
    0.6048
```

Thus, we have an infinite gain margin, and a phase margin of 147°, which indicates that the airspeed control system with both elevator and throttle inputs not only meets the closed-loop performance requirements, but also is quite robust

with respect to parametric variations. Moreover. Figure 4.32 shows a gain roll-off of -40 dB per decade at high frequencies ($\omega > 10$ rad/s) thereby exhibiting excellent noise rejection properties.

4.7 Lateral-Directional Control Systems

Rotation and translation is not confined to the plane of symmetry and is called *lateral-directional* dynamics. It consists of roll (Φ, P), yaw (Ψ, R), and sideslip (V) as its three degrees of freedom. As in the case of longitudinal dynamics (pitch rotation, forward, and downward translation) the roll, yaw, and sideslip are inextricably intertwined. Since the flight direction in the horizontal plane is governed by the lateral-directional dynamics, it is crucial to have a dedicated lateral-directional control system for aircraft navigation (maneuvering in the horizontal plane).

The lateral-directional dynamics is governed by the following equations derived from (4.5), (4.10), and (4.15):

$$\begin{aligned}
\dot{\Phi} &= P + Q \sin\Phi \tan\Theta + R \cos\Phi \tan\Theta \\
\dot{\Psi} &= Q \sin\Phi \sec\Theta + R \cos\Phi \sec\Theta \\
Y &+ mg \sin\Phi \cos\Theta = m(\dot{V} + RU - PW) \\
L &= J_{xx}\dot{P} + J_{xz}(\dot{R} + PQ) + \left(J_{zz} - J_{yy}\right) QR \\
N &= J_{zz}\dot{R} + J_{xz}(\dot{P} - QR) + \left(J_{yy} - J_{xx}\right) PQ.
\end{aligned} \tag{4.168}$$

We note that the lateral-directional dynamics plant is of order five and is inherently nonlinear. It is also influenced by the longitudinal variables (Θ, Q, U, W). However, by assuming a small perturbation from an equilibrium flight condition, it is possible not only to uncouple the lateral-directional and longitudinal dynamics, but also to linearize the resulting plant about the equilibrium. Thus, we have the following linearized lateral-directional dynamics:

$$\begin{aligned}
\dot{\phi} &= p + r \tan\Theta_{\mathrm{e}} \\
\dot{\psi} &= r \sec\Theta_{\mathrm{e}} \\
Y &+ mg\phi \cos\Theta_{\mathrm{e}} = mU(\dot{\beta} + r) \\
L &= J_{xx}\dot{p} + J_{xz}\dot{r} \\
N &= J_{zz}\dot{r} + J_{xz}\dot{p},
\end{aligned} \tag{4.169}$$

where ϕ, ψ, p, r are small perturbations in roll angle, yaw angle, roll rate, and yaw rate, respectively, from a straight line, equilibrium flight condition ($\Phi_e = V_e = P_e = R_e = Q_e = 0$),[10] and β is a small *sideslip angle* given by

$$\beta = \tan^{-1} \frac{v}{U \cos\alpha} \simeq \frac{v}{U}.$$

The lateral-directional aerodynamic force and torque components can be expanded in a first-order Taylor series about the equilibrium condition:

$$\begin{aligned} Y &= Y^e + Y_\beta \beta + Y_\phi \phi + Y_{\dot\beta} \dot\beta + Y_p p + Y_r r + Y_R \delta_R \\ L &= L^e + L_\beta \beta + L_\phi \phi + L_{\dot\beta} \dot\beta + L_p p + L_r r + L_A \delta_A + L_R \delta_R \\ N &= N^e + N_\beta \beta + N_\phi \phi + N_{\dot\beta} \dot\beta + N_p p + N_r r + N_A \delta_A + N_R \delta_R. \end{aligned} \tag{4.170}$$

Note that these are independent of the longitudinal perturbation quantities, (θ, q, u, α) due to the plane of symmetry. The sideforce, the rolling moment, and the yawing moment must vanish at equilibrium, $Y^e = L^e = N^e$. Furthermore, due to the plane of symmetry, the derivatives, L_ϕ, N_ϕ, Y_ϕ, are necessarily zeros. The β and $\dot\beta$ derivatives are primarily due to the vertical tail (or *fin*) (Fig. 4.33), and are caused by the same unsteady aerodynamic effects as the α and $\dot\alpha$ derivatives. The rate derivatives L_p, N_r affect the damping of the lateral dynamic modes and are called damping in roll and yaw, respectively. The other rate derivatives, Y_p, L_r, N_p, are mainly due to the wing and the vertical tail. These signify the inherent aerodynamic cross-coupling among roll, yaw, and sideslip degrees of freedom [5].

The control surface deflections, *aileron angle*, δ_A, and *rudder*, δ_R (Fig. 4.33) are the lateral-directional control inputs. A pair of control-surfaces in the (xy) plane, located symmetrically about the axis (ox) and deflected in mutually opposite directions by angles δ_1 and δ_2 (Fig. 4.33) are called *ailerons*, which are used as roll control devices. The aileron angle, δ_A, is the average of the two separate deflections, $\delta_A = (\delta_1 + \delta_2)/2$, and is designed such that a control rolling moment, $L_A \delta_A$, is produced along with a much smaller, undesirable yawing moment, $N_A \delta_A$, called *adverse aileron yaw*. A control-surface mounted on the vertical tail (fin) in the (xz) plane behind the (oy) axis is called the *rudder*. A rudder deflection, δ_R, creates a sideforce, $Y_R \delta_R$, control yawing moment, $N_R \delta_R$, and a much smaller rolling moment, $L_R \delta_R$. Rudder is primarily used to correct a lateral flight asymmetry. A positive aileron angle creates a positive rolling moment, which is an upward deflection of the control surface on the right wing, and a downward deflection on the left, as shown in Fig. 4.33. A positive rudder deflection generates a positive yawing moment, as shown in Fig. 4.33.

10
1. Note that the equilibrium flight direction, Ψ_e, need not be zero.
2. We are using ϕ and ψ for roll and yaw angle perturbations in this section. This should not cause confusion with the usage of ϕ for the flight path angle and ψ for the heading (azimuth) angle elsewhere in the book (such as in Chap. 2).

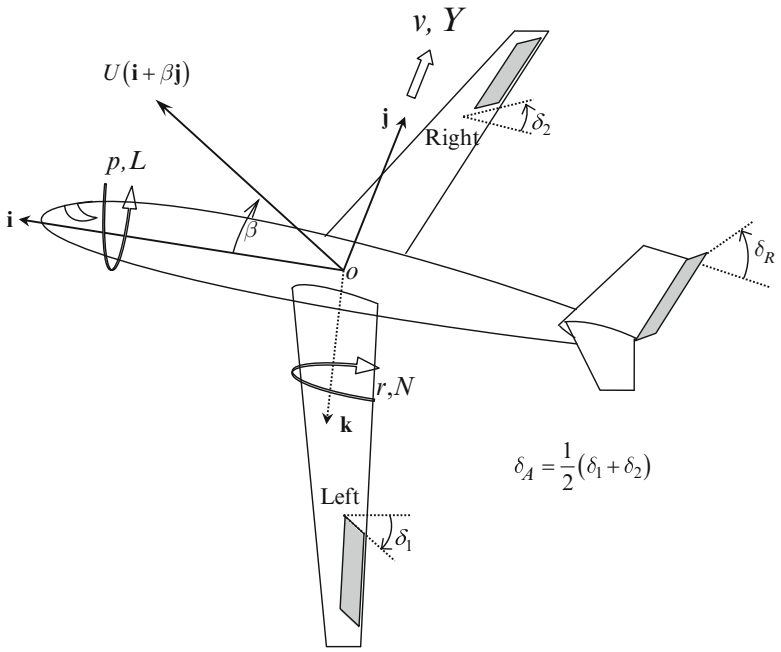


Fig. 4.33 Geometry for lateral-directional dynamics, showing a positive aileron angle, δ_{A}, and a positive rudder angle, δ_{R}

Substitution of (4.170) into (4.169) results in the following state-space representation for lateral-directional dynamics:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{Bmatrix} = \mathsf{A}_{\mathrm{LD}} \begin{Bmatrix} \phi \\ \psi \\ \beta \\ p \\ r \end{Bmatrix} + \mathsf{B}_{\mathrm{LD}} \begin{Bmatrix} \delta_{\mathrm{A}} \\ \delta_{\mathrm{R}} \end{Bmatrix}, \tag{4.171}$$

where

$$\mathsf{A}_{\mathrm{LD}} = \begin{bmatrix} 0 & 0 & 0 & 1 & \tan\Theta_{\mathrm{e}} \\ 0 & 0 & 0 & 0 & \sec\Theta_{\mathrm{e}} \\ \frac{g\cos\Theta_{\mathrm{e}}}{U} & 0 & \frac{Y_\beta}{mU} & \frac{Y_p}{mU} & \left(\frac{Y_r}{mU} - 1\right) \\ 0 & 0 & (AL_\beta + BN_\beta) & (AL_p + BN_p) & (AL_r + BN_r) \\ 0 & 0 & (CN_\beta + BL_\beta) & (CN_p + BL_p) & (CN_r + BL_r) \end{bmatrix}, \tag{4.172}$$

$$\mathsf{B}_{\mathrm{LD}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{Y_A}{mU} & \frac{Y_{\mathrm{R}}}{mU} \\ (AL_{\mathrm{A}} + BN_{\mathrm{A}}) & (AL_{\mathrm{R}} + BN_{\mathrm{R}}) \\ (CN_{\mathrm{A}} + BL_{\mathrm{A}}) & (CN_{\mathrm{R}} + BL_{\mathrm{R}}) \end{bmatrix}, \tag{4.173}$$

with the inertial terms given by

$$\begin{aligned} A &= \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2} \\ B &= \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2} \\ C &= \frac{J_{xx}}{J_{xx}J_{zz} - J_{xz}^2}. \end{aligned} \tag{4.174}$$

4.7.1 Pure Rolling Mode

Before we carry out an eigenvalue analysis of the lateral-directional system, it is evident from (4.171) that a particular form of lateral-directional motion involves $\beta = r = 0$, which produces

$$\begin{aligned} \dot{\phi} &= p \\ \dot{\psi} &= 0 \\ Y_{\mathrm{R}}\delta_{\mathrm{R}} &= -mg\phi\cos\Theta_{\mathrm{e}} - Y_p p - Y_A\delta_{\mathrm{A}} \\ L_{\mathrm{A}}\delta_{\mathrm{A}} + L_{\mathrm{R}}\delta_{\mathrm{R}} &= J_{xx}\dot{p} - L_p p \\ N_{\mathrm{A}}\delta_{\mathrm{A}} + N_{\mathrm{R}}\delta_{\mathrm{R}} &= -J_{xz}\dot{p} - N_p p. \end{aligned} \tag{4.175}$$

Upon neglecting the much smaller terms, $Y_A\delta_{\mathrm{A}}$, $Y_p p$, $L_{\mathrm{R}}\delta_{\mathrm{R}}$, $J_{xz}\dot{p}$, and $N_p p$ in comparison with the other terms, we have the following approximation:

$$\ddot{\phi} - \frac{L_p}{J_{xx}}\dot{\phi} = \frac{L_{\mathrm{A}}}{J_{xx}}\delta_{\mathrm{A}}, \tag{4.176}$$

and

$$\delta_{\mathrm{R}} = -\frac{mg\cos\Theta_{\mathrm{e}}}{Y_{\mathrm{R}}}\phi. \tag{4.177}$$

Equation (4.176) represents a single degree-of-freedom rolling motion called the *pure rolling mode*, which emerges from the lateral-directional dynamics for straight flight ($\Theta_e = \text{const.}$), if the rudder input is coordinated with the roll angle according to (4.177). Generally, for a well designed airplane only a small rudder deflection is required for simultaneously nullifying both sideslip and yaw rate. Equation (4.177) also implies that it is possible to roll the aircraft by the application of rudder input alone, which is sometimes tried by aerobatic pilots.

Equation (4.176) can be expressed as follows:

$$\ddot{\phi} + a_0\dot{\phi} = b_0\delta_A, \tag{4.178}$$

where

$$\begin{aligned} a_0 &= -\frac{L_p}{J_{xx}} \\ b_0 &= \frac{L_A}{J_{xx}} \end{aligned} \tag{4.179}$$

are constant coefficients. The transfer function for the pure rolling mode is thus given by

$$G(s) = \frac{\phi(s)}{\delta_A(s)} = \frac{b_0}{s^2 + a_0 s}. \tag{4.180}$$

While the transfer function between the rudder and aileron angles is given in the following:

$$\frac{\delta_R(s)}{\delta_A(s)} = \frac{\delta_R(s)}{\phi(s)} G(s) = -\frac{mgb_0 \cos\Theta_e / Y_R}{s^2 + a_0 s}. \tag{4.181}$$

Example 4.14. Consider a fighter airplane climbing steadily with a constant flight path angle of 30° and airspeed U = 272 m/s at an altitude of 5 km. The airplane has the following parameters at the given flight condition:

$$m = 6,181 \text{ kg};\ J_{xx} = 1.5 \times 10^4 \text{ kg m}^2;\ J_{zz} = 5 \times 10^4 \text{ kg m}^2;\ J_{xz} = 456 \text{ kg m}^2$$

$$Y_\beta = -5.46 \times 10^5 \text{ N/rad};\ Y_p = 0;\ Y_r = 2.87 \times 10^5 \text{ N s/rad}$$

$$Y_A = 0;\ Y_R = -1.5 \times 10^5 \text{ N/rad}$$

$$L_\beta = -7 \times 10^5 \text{ N m/rad};\ L_p = -9.3 \times 10^4 \text{ N m s/rad}$$

$$L_r = 1.02 \times 10^4 \text{ N m s/rad};$$

$$L_A = 1.5 \times 10^6 \text{ N m/rad};\ L_R = -805 \text{ N m/rad}$$

$$N_\beta = 1.1 \times 10^6 \text{ N m/rad};\ N_p = -960 \text{ N m s/rad};$$

$$N_r = -3.19 \times 10^4 \text{ N m s/rad};$$

$$N_A = -1.06 \times 10^3 \text{ N m/rad};\ N_R = 3.185 \times 10^4 \text{ N m/rad}$$

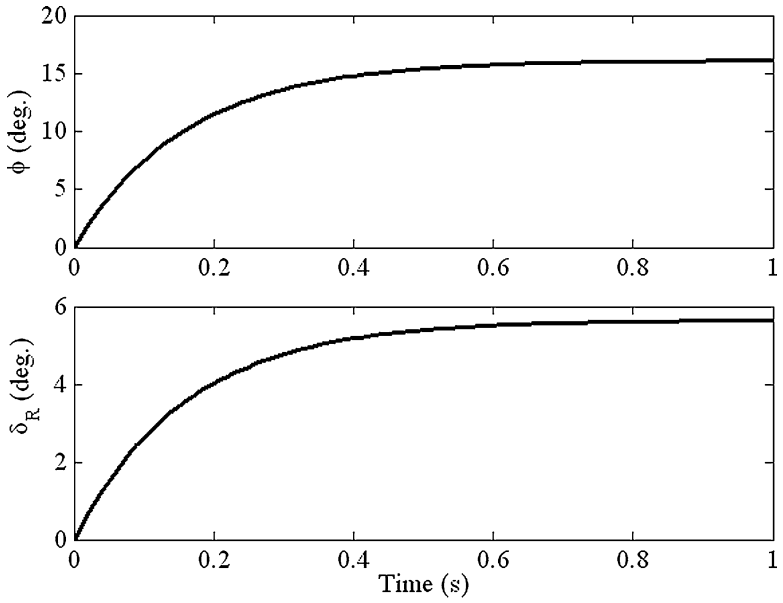


Fig. 4.34 Response of the pure rolling mode to a unit step aileron deflection

Analyze the stability of the pure rolling mode in the given flight condition, and plot the roll angle and rudder angle response to a step aileron deflection of 1°.

We begin by computing the coefficients of the rolling mode transfer function by (4.179):

$$a_0 = -\frac{-9.3}{1.5} = 6.2 \text{ s}^{-1}$$
$$b_0 = \frac{150}{1.5} = 100 \text{ s}^{-2}.$$

The roots of the plant's characteristic polynomial are thus given by

$$s_1 = 0; \qquad s_2 = -a_0 = -6.2,$$

which implies stability with a settling-time of $t_s \simeq 2\pi/6.2 = 1.01$ s. The roll angle response and the required rudder deflection of the pure rolling mode to a 1° step aileron deflection are computed using MATLAB-CST as follows, and plotted in Fig. 4.34:

```
>> sys=tf(100,[1 6.2]) %roll angle/aileron angle transfer function

Transfer function:
  100
-------
s + 6.2

>> [y,t]=step(sys); %Unit step response for roll angle
```

```
% rudder angle:
>> sys1=tf(100*6181*9.81*cos(pi/6)/1.5e5,[1 6.2])

Transfer function:
 35.01
-------
s + 6.2

>> [y1,t1]=step(sys1); %Unit step response for rudder deflection
>> subplot(211),plot(t,y),hold on,subplot(212),plot(t1,y1)
```

Note the steady-state roll angle of 16.13° and a corresponding steady-state rudder deflection of 5.65°, which are the DC gains of the respective transfer functions.

4.7.2 Roll Control System

Roll control is one of the most important handling characteristics of an aircraft, where horizontal maneuverability depends upon how quickly and accurately a desired roll angle can be achieved without requiring a large aileron input (much like the steering control in a car). Good roll control is especially important for fighter type aircraft, so much so that it is often included in primary mission requirements.

A roll control system requires a step change in the roll angle should be achieved quickly and without a steady-state error. Since the rolling mode has a pure integrator in the open-loop plant, integral action is not required for producing a zero steady-state error to a step change roll angle. However, it may be necessary for better maneuverability to reduce the settling-time by increasing the damping and/or natural frequency of the second-order, closed-loop system. Thus, a proportional-derivative (PD) controller, $H(s) = k_1 + k_2 s$ seems well suited for the task, with which the closed-loop characteristic polynomial becomes the following:

$$1 + G(s)H(s) = s^2 + (a_0 + k_2 b_0)s + k_1 b_0. \tag{4.182}$$

The gain k_2 directly affects the settling-time,

$$t_s = \frac{8}{a_0 + k_2 b_0}, \tag{4.183}$$

while k_1 can be selected for a desired damping ratio,

$$\zeta = \frac{a_0 + k_2 b_0}{2\sqrt{k_1 b_0}}. \tag{4.184}$$

Considering the practical implementation of PD control, roll angle is not the sole output variable because taking a derivative of the output is fraught with inaccuracies, and is also susceptible to measurement noise. Instead, roll angle output

of a rate-integrating vertical gyro is combined with the rate gyro output for roll rate, $p = \dot{\phi}$. Consequently, the control-law for a desired roll angle, ϕ_d, is as follows:

$$\delta_{\text{Ac}} = k_1 (\phi_d - \phi) - k_2 p, \tag{4.185}$$

with the commanded aileron deflection, δ_{Ac}, fed to the aileron servo. The aileron servo can be cable–pulley–spring mechanical type for a small aircraft, electro-hydraulic type for a large transport aircraft, or electric motor driven (*fly-by-wire*) type for a modern fighter aircraft. Due to the higher natural frequency of the pure rolling mode when compared to the longitudinal short-period mode, the aileron servo is typically faster than the elevator servo. Time constants of the order of 0.05 s are common in the modern electric fly-by-wire aileron and rudder servos for fighter aircraft.

The linear range of rolling moment with aileron deflection is usually limited to $\delta_{\text{A}} = \pm 25°$, thus aileron servo can be regarded as being saturated outside these limits. Similar limits are valid for the rudder deflection, δ_{R}.

Example 4.15. For the aircraft of Example 4.14 in the given flight condition, design a roll control system to achieve a roll angle of 20° in about 0.2 s, without exceeding the maximum aileron and rudder deflections of ±25°. Consider first-order aileron and rudder servos of time constant 0.05 s for the closed-loop simulation and neglect the dynamics of the rate and rate-integrating gyros.

For a second-order system with settling-time of 0.2 s and a damping ratio of $\zeta = 1/\sqrt{2}$, we could achieve the desired roll performance with the gains given by (4.183) and (4.184). However, with the presence of the aileron servo the system is of third-order and we can hardly expect these feedback gains to perform as expected. Instead, we increase the gains by increasing the target natural frequency and damping ratio and is provided as follows:

$$k_2 = \frac{160 - 6.2}{100} = 1.538,$$

and

$$k_1 = \frac{(6.2 + 100k_2)^2}{(400)(1.49^2)} = 28.8275.$$

Even with the increased feedback gains, there will be a lag in the closed-loop response due to aileron servo time constant of 0.05 s, thereby increasing the settling-time. A way of speeding-up the closed-loop response without appreciably increasing the control effort is by putting a *lead-compensator* (Chap. 3) of transfer function,

$$F(s) = \frac{0.05s + 1}{0.02s + 1},$$

in series with the aileron servo. The lead-compensator has the effect of moving the servo pole from $s = 20$ to $s = 50$ and does not affect the steady-state error because

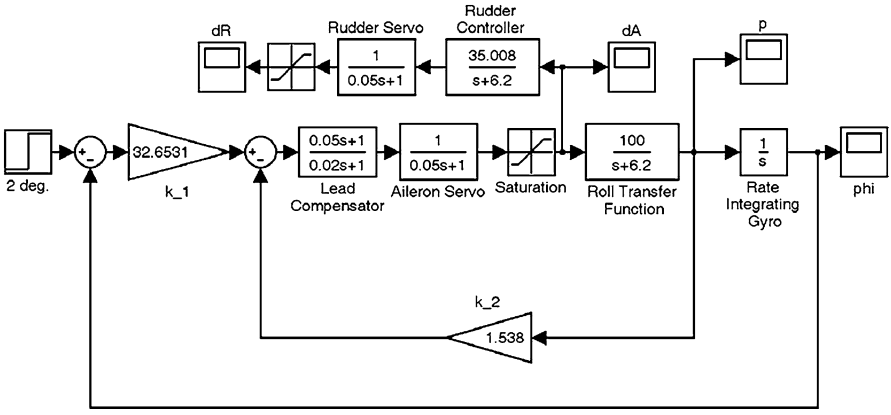


Fig. 4.35 *Simulink* block diagram of roll control system for a fighter aircraft with 20° step desired roll angle

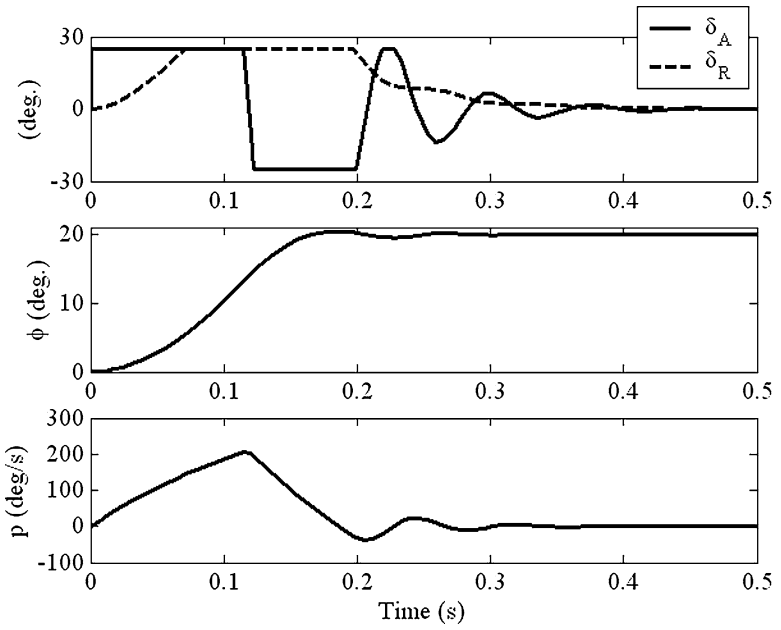


Fig. 4.36 Closed-loop response of roll control system for a fighter aircraft to a 20° step desired roll angle

it has a DC gain of unity. A *Simulink* block diagram of the closed-loop system with PD controller, lead-compensator, rudder controller, aileron and rudder servos and saturation blocks is shown in Fig. 4.35, while its response is plotted in Fig. 4.36. Note that the desired roll angle of 20° is achieved with a settling-time of 0.19 s

and with maximum transient roll rate exceeding 200°/s. Both aileron and rudder see saturation in the first 0.2 s, and thereafter settle to zero in about 0.45 s. This example is typical of a modern fighter aircraft, which can also have aileron deflection in the nonlinear range ($|\delta_A| > 25°$) for increasing the peak roll rate.

4.7.3 Lateral-Directional Modes

One can determine the lateral-directional stability in the controls-fixed case from the eigenvalues of the dynamics matrix, A_{LD}. Furthermore, we may also be able to approximate the lateral-directional dynamics by first- and second-order modes as we did in the case of longitudinal dynamics.

The eigenvalues of the lateral-directional plant dynamics matrix, A_{LD}, are the roots of the following fifth-order characteristic polynomial that can be typically factored into a pole at the origin ($s = 0$), two real poles, ($s = -a_0, s = -a_1$), and a pair of complex conjugate poles:

$$|s\mathsf{I} - \mathsf{A}_{LD}| = s(s + a_0)(s + a_1)\left(s^2 + 2\zeta_d\omega_d s + \omega_d^2\right) = 0, \tag{4.186}$$

where (ω_d, ζ_d) are the natural frequency and damping ratio, respectively, of the quadratic subsystem called the *Dutch-roll mode* in standard aeronautical terminology. We have already identified $s = -a_0 \simeq L_p/J_{xx}$ as an approximate, rolling mode; thus the concerned subsystem is called the *pure rolling mode* (or *roll subsidence mode*). Both pure rolling and Dutch-roll mode are of high natural frequency, while the remaining pole, $s = -a_1$ is generally located close to the imaginary axis and is called the *spiral mode*. If we consider the eigenvectors associated with the respective modes, we can say which degrees of freedom dominate in a particular mode.

The pure-rolling mode is directly related to an aircraft's maneuverability, as discussed in the previous subsection. The alacrity with which the bank angle can be changed has a proportional effect on the ability to initiate a turn quickly. However, once the vehicle starts to turn, the other two dynamic modes usually enter the scene. Of these, the spiral mode, being either a slow decay or growth of bank angle with time – depending upon whether the mode is stable – can be easily controlled through aileron input. However, it is the Dutch-roll mode (DRM) that offers the greatest challenge in lateral-directional control because it consists of a simultaneous, high-frequency oscillation in bank and yaw. A general lateral-directional autopilot would involve an active control of all the three modes by the application of both aileron and rudder inputs, and thus requires a multi-variable approach.

Example 4.16. For the aircraft of Example 4.14, determine the eigenvalues of the system in the given flight condition. Using the eigenvector analysis, identify the predominant degrees of freedom for each mode.

With the numbers given in Example 4.14, the dynamics matrix is provided in the following:

$$\mathsf{A}_{\mathrm{LD}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0.5774 \\ 0 & 0 & 0 & 0 & 1.1547 \\ 0.0312 & 0 & -0.3248 & 0 & -0.8293 \\ 0 & 0 & -46.0115 & -6.2023 & 0.6608 \\ 0 & 0 & 21.5518 & -0.0757 & -0.6311 \end{bmatrix}.$$

The eigenvalues are computed using MATLAB-CST as follows:

```
>> A =[  0         0         0    1.0000    0.5774
         0         0         0         0    1.1547
    0.0312         0   -0.3248         0   -0.8293
         0         0  -46.0115   -6.2023    0.6608
         0         0   21.5518   -0.0757   -0.6311];

>> damp(A)

        Eigenvalue            Damping      Freq. (rad/s)
  1.72e-002                -1.00e+000       1.72e-002
  0.00e+000                -1.00e+000       0.00e+000
 -4.48e-001 + 4.28e+000i    1.04e-001       4.31e+000
 -4.48e-001 - 4.28e+000i    1.04e-001       4.31e+000
 -6.28e+000                 1.00e+000       6.28e+000
```

Clearly, the pole associated with the pure rolling mode is $s = -6.28$, which is quite close to the approximate pole estimated in Example 4.14. The Dutch-roll mode is given by $\omega_{\mathrm{d}} = 4.31$ rad/s, $\zeta_{\mathrm{d}} = 0.104$, while the spiral mode is unstable with pole $s = 0.0172$. Next, we conduct the eigenvector analysis as follows:

```
>> [V,D]=eig(A)

V =

     0   -0.3728       0.0531-0.1365i   0.0531+0.1365i  -0.1579
1.0000   -0.9278       0.1427+0.0767i   0.1427-0.0767i  -0.0012
     0   -0.0004      -0.0996-0.0633i  -0.0996+0.0633i   0.0017
     0    0.0016          0.7571           0.7571        0.9874
     0   -0.0139      -0.3398+0.4997i  -0.3398-0.4997i   0.0066

D =
     0         0             0                0               0
     0    0.0172             0                0               0
     0         0      -0.4481+4.2839i         0               0
     0         0             0          -0.4481-4.2839i       0
     0         0             0                0         -6.2791
```

The spiral mode has the eigenvector

$$\phi = -0.3728,\ \psi = -0.9278,\ \beta = -0.0004,\ p = 0.0016,\ r = -0.0139$$

which implies a negligible sideslip and roll rate, but a significant yawing motion with non-zero roll angle. Thus the spiral mode is approximately a coordinated turn ($\beta \simeq 0$) with a slowly varying roll angle, which translates into a spiral path. For a stable spiral mode, one expects the turn to eventually decay into a straight line with roll angle slowly decreasing (and turn radius increasing) with time. An unstable spiral mode (as we have here) results in a tightening spiral in which the roll angle keeps on increasing with time.

The complex conjugate eigenvectors of the Dutch-roll mode indicate oscillations in roll, yaw and sideslip. A closer inspection reveals that the sideslip angle response is nearly equal in magnitude to that of the yaw angle,

$$|\psi| = |0.1427 + 0.0767i| = 0.162\ ;\ \ |\beta| = |-0.0996 - 0.0633i| = 0.118,$$

but nearly opposite in phase,

$$\text{phase}(\psi) = 28.2^\circ;\ \ \text{phase}(\beta) = 212.4^\circ,$$

thereby implying $\beta \simeq -\psi$. Furthermore, the combined roll and yaw rate magnitudes are much larger than that of the sideslip, which indicates an insignificant sideslip motion in comparison with the coupled roll and yaw oscillation. Hence, we visualize the Dutch-roll mode to be a snaking roll-yaw oscillation, with a small sideslip component that is opposite in phase to the yaw angle.

4.7.3.1 Open-Loop Response to Aileron and Rudder Inputs

Unit step aileron and rudder inputs applied individually, can reveal the lateral-directional modes. Such a method is often employed to determine the unknown modes by a flight test. However, rather than looking directly at the time domain data which might reveal nothing but an unstable spiral mode, it is more practical to employ frequency domain analysis (fast Fourier transform [20]) to the raw signal in order to estimate the natural frequencies of some modes.

Example 4.17. For the aircraft of Example 4.14, determine the roll angle frequency response of the system to both aileron and rudder inputs at the given flight condition.

For obtaining the frequency response, we will consider the transfer functions, $\phi(s)/\delta_A(s)$ and $\phi(s)/\delta_R(s)$, which are obtained by the following MATLAB-CST statements, beginning with a pre-computed matrix, B_{LD}, by (4.173) and the matrix A_{LD} computed in Example 4.16:

```
>> B=[0   0; 0    0; 0   -0.0892; 100.0271   -0.0343; 0.8910   0.6367];
>> sys=ss(A,B(:,1),[1 0 0 0 0],0); tf(sys) %phi(s)/delta_A(s)
>> tf(sys)
```

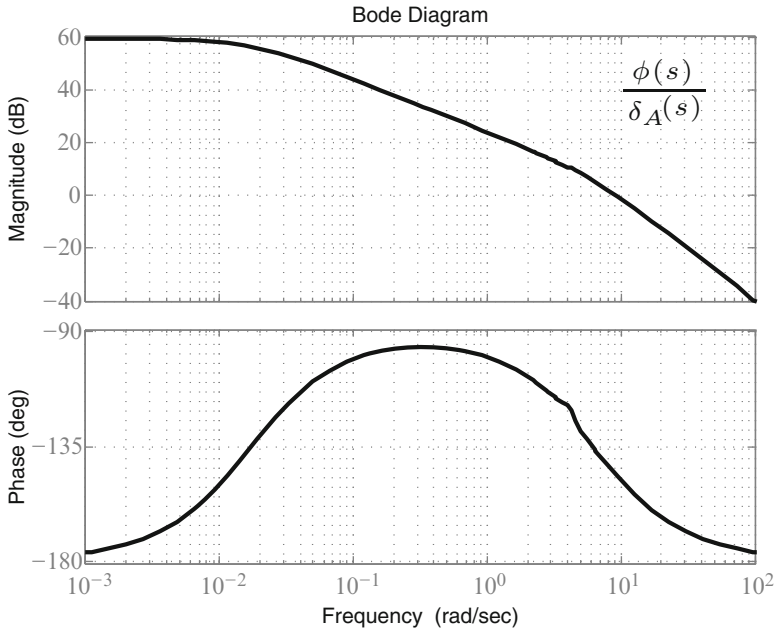


Fig. 4.37 Open-loop frequency response of the transfer function, $\phi(s)/\delta_{\mathrm{A}}(s)$, for a fighter aircraft

```
Transfer function:
         100.5 s^2 + 95.19 s + 1842
-----------------------------------------------
s^4 + 7.158 s^3 + 24.06 s^2 + 116.1 s - 2.009

>> bode(sys)
>> sys=ss(A,B(:,2),[1 0 0 0 0],0); tf(sys) %phi(s)/delta_R(s)

Transfer function:
        0.3333 s^2 + 5.784 s + 18.81
-----------------------------------------------
s^4 + 7.158 s^3 + 24.06 s^2 + 116.1 s - 2.009

>> bode(sys)
```

The resulting Bode plots are shown in Figs. 4.37 and 4.38. The first plot (Fig. 4.37) indicates two distinct modes as variations in the gain slope, accompanied by phase changes of 90°. The spiral mode at natural frequency of nearly $10^{(-1.76)} = 0.0174$ rad/s is roughly identified at phase crossing the midpoint of −180° and −90° (i.e., −135°), which is unstable due to an increasing phase at that point. Similarly, the second mode is identified as Dutch-roll mode in the slope of the gain plot and a small notch in the phase plot at frequency, 4.3 rad/s. The mode is stable due to a decreasing phase at the natural frequency. The Dutch-roll mode is more clearly visible in Fig. 4.38 for the transfer function, $\phi(s)/\delta_{\mathrm{R}}(s)$, as a sharp peak in the gain plot accompanied by a phase drop by 90°.

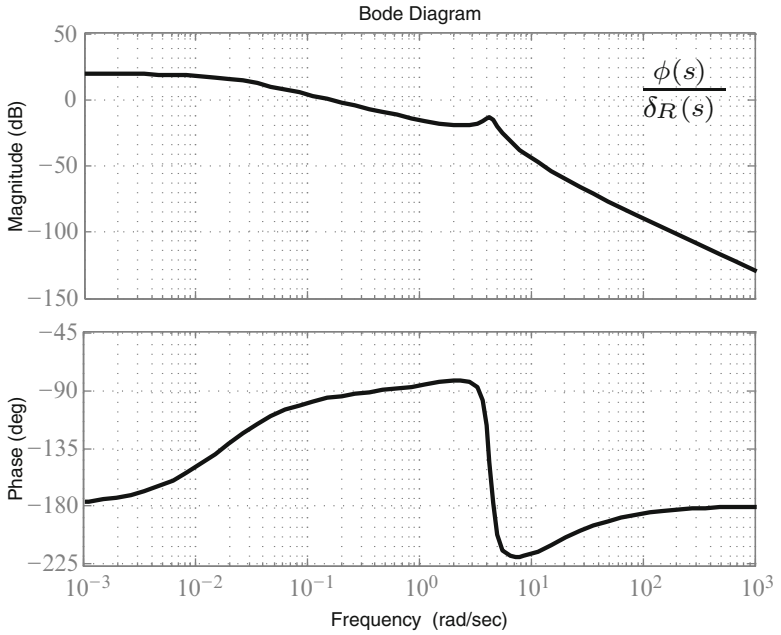


Fig. 4.38 Open-loop frequency response of the transfer function, $\phi(s)/\delta_{\mathrm{R}}(s)$, for a fighter aircraft

4.7.4 Dutch-Roll (or Yaw) Damper

The damping of the Dutch-roll mode (DRM) is crucial to the mission of any aircraft. If too lightly damped, the DRM leads to an uncomfortable ride in a passenger aircraft, or a loss of weapons pointing, targeting, and delivery accuracy in a military aircraft. Dutch-roll damping depends largely on the derivatives N_r and L_β, while its frequency is also affected by N_β. In most unmanned aircraft, DRM damping is sacrificed in favor of having a stable spiral mode, which requires a larger magnitude of L_β that is achieved by having a pronounced wing *dihedral* angle (angle made by the wings with the horizontal, in a level attitude). However, the reverse is true for most manned aircraft, where the dihedral angle must be reduced by a designer for a better DRM damping. High-speed aircraft experience a marked decline in the yaw damping derivative, N_r, due to an aft shift of the fuselage aerodynamic center as the speed of sound is crossed (in a manner quite similar to pitch damping, M_q, as seen in Example 4.11). Therefore, in order to maintain an adequate DRM damping, a closed-loop action is necessary for most high-speed aircraft, which is commonly called a *yaw damper*, or *directional stability augmentation system*.

Since the aileron input produces an adverse aileron yaw ($N_{\mathrm{A}} < 0$) thereby aggravating the DRM, they are generally avoided in designing a yaw damper, which consists simply of a proportional feedback from a yaw rate gyro to the rudder servo

$$\delta_{Rc} = -k_r r. \tag{4.187}$$

The value of the feedback gain, k_r, is normally scheduled with Mach number (Example 4.11) such that an adequate level of DRM damping is obtained throughout the flight envelope.

Example 4.18. For the aircraft of Example 4.14, design a closed-loop yaw damper based on yaw rate feedback such that the DRM damping ratio is doubled, while leaving its natural frequency unchanged. Simulate the closed-loop response due to a lateral gust that causes an initial sideslip perturbation of $\beta(0) = 5°$ at the given flight condition. The rudder servo has a time constant 0.05 s and the maximum deflection limits of $\pm 25°$.

Example 4.16 revealed an open-loop DRM damping ratio of 0.104. We shall attempt to double, while keeping the closed-loop DRM natural frequency unchanged near 4 rad/s. Since the yaw damper would not greatly affect the pure-rolling and spiral modes, their respective eigenvalues are not expected to change. Keeping in mind the specifications of the maximum sideslip disturbance as well as the maximum allowable rudder deflection, we select $k_r = 1.25$, and compute the closed-loop initial response that is provided as follows:

```
>> A = [ 0          0          0    1.0000    0.5774
         0          0          0         0    1.1547
    0.0312          0    -0.3248         0   -0.8293
         0          0   -46.0115   -6.2023    0.6608
         0          0    21.5518   -0.0757   -0.6311];

>> B = [ 0          0    -0.0892   -0.0343    0.6367]';

>> K=[0 0 0 0 1.25]; % feedback gain matrix

>> a=20; % rudder servo pole (a=1/0.05)

% Plant augmented with rudder servo:
>> Abar=[A B;0 0 0 0 0 -a]; Bbar=[zeros(size(B)); a];

>> Ac=Abar-Bbar*[K 0]; % closed-loop dynamics matrix

>> damp(Ac) % closed-loop eigenvalues

Eigenvalue             Damping     Freq. (rad/s)
  0.00e+000                -1.00e+000      0.00e+000
 -1.90e+001                 1.00e+000      1.90e+001
  8.73e-003                -1.00e+000      8.73e-003
 -9.21e-001 + 4.04e+000i     2.22e-001      4.14e+000
 -9.21e-001 - 4.04e+000i     2.22e-001      4.14e+000
 -6.28e+000                 1.00e+000      6.28e+000

% Closed-loop system:
>> sys=ss(Ac,zeros(6,1),zeros(2, 6),zeros(2,1));

a =
            x1       x2       x3       x4       x5       x6
   x1        0        0        0        1   0.5774        0
   x2        0        0        0        0    1.155        0
   x3   0.0312        0  -0.3248        0  -0.8293  -0.0892
   x4        0        0   -46.01   -6.202   0.6608  -0.0343
   x5        0        0    21.55  -0.0757  -0.6311   0.6367
   x6        0        0        0        0      -25      -20
```

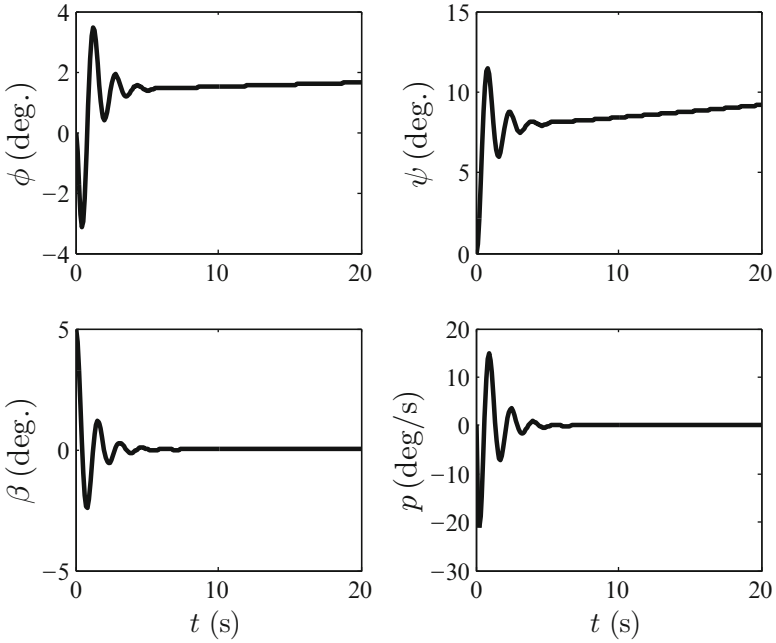


Fig. 4.39 Closed-loop response of a fighter aircraft equipped with a yaw damper to a 5° initial sideslip disturbance

```
b =
       u1
   x1   0
   x2   0
   x3   0
   x4   0
   x5   0
   x6   0

c =
       x1  x2  x3  x4  x5  x6
   y1   0   0   0   0   0   0
   y2   0   0   0   0   0   0

d =
       u1
   y1   0
   y2   0

Continuous-time model.

% Closed-loop initial response:
>> [y,t,x]=initial(sys,[0 0 5*pi/180 0 0 0]',20);

>> margin(sys); % Closed-loop frequency response
```

Note that while the closed-loop DRM damping is more than doubled to 0.222 without much affecting the natural frequency, there is virtually no change in the spiral and pure-rolling modes (as expected). The closed-loop initial response plotted in Fig. 4.39 shows well damped roll rate, yaw rate, and sideslip transients that settle

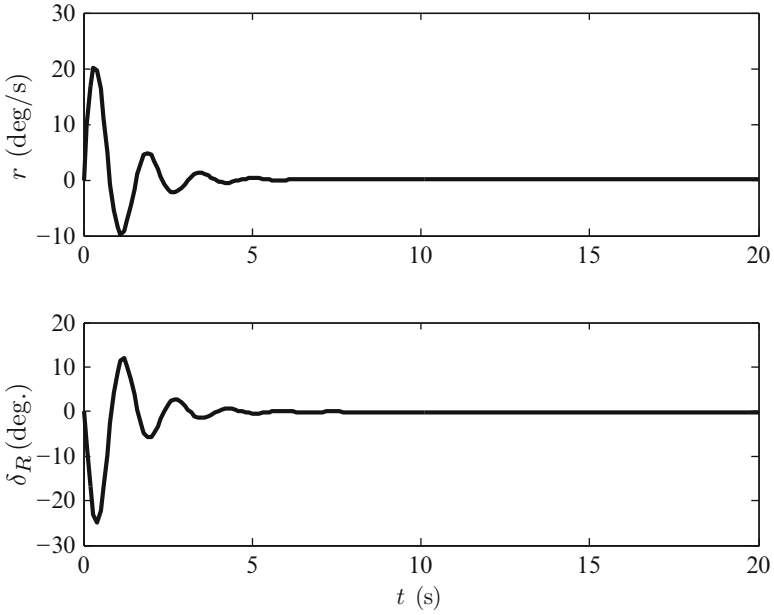


Fig. 4.40 Closed-loop response of a fighter aircraft equipped with a yaw damper to a 5° initial sideslip disturbance (continued)

to zero in about 5 s. However, the roll and yaw angles plotted in Fig. 4.40 show a steady-state error due to the unstable spiral mode, which must be corrected by an aileron input applied either by the pilot, or a heading-hold autopilot. The rudder deflection (Fig. 4.40) does not exceed the limits of ±25°.

4.7.5 Heading Autopilot

The most common task of a lateral autopilot is to either hold a constant heading (velocity azimuth) or to vary it in response to command inputs received from the navigational system. When left on its own, an aircraft with an unstable spiral mode slowly drifts away from a constant desired heading. Even in aircraft with a stable spiral mode, there is usually an insufficient damping as a result of compromise struck with an increased Dutch-Roll damping. Consequently, a lateral atmospheric disturbance causes the heading perturbation to prevail for a long time. Therefore, the primary task of a heading autopilot is to increase spiral stability, while tracking a step heading command. Concerted action by both rudder and ailerons is required by a heading autopilot. For tracking a constant heading angle, ψ_d, it is necessary to have an appropriate feedforward gain matrix, $\mathsf{K_d}$, which satisfies (3.122).

Example 4.19. Design a heading autopilot for the aircraft with lateral-directional stability and control derivatives given in Example 4.14, but in a steady and level

(rather than climbing) flight at the given altitude. Base your design on heading angle and yaw rate feedback and aileron and rudder inputs, such that a step heading command of 2° is achieved in about 10 s. Assume both rudder and aileron servos to be of first-order, with time constant 0.05 s and maximum deflection limits of ±25°.

The lateral-directional dynamics is now represented by the following state-space model (without the servos):

$$\mathsf{A}_{\mathrm{LD}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.0361 & 0 & -0.3248 & 0 & -0.8293 \\ 0 & 0 & -46.0115 & -6.2023 & 0.6608 \\ 0 & 0 & 21.5518 & -0.0757 & -0.6311 \end{pmatrix},$$

and

$$\mathsf{B}_{\mathrm{LD}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -0.0892 \\ 100.0271 & -0.0343 \\ 0.8910 & 0.6367 \end{pmatrix} \; ; \; \mathsf{C}_{\mathrm{LD}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

and $\mathsf{D}_{\mathrm{LD}} = 0$. With this lateral-directional dynamics plant, we require

$$(\mathsf{A}_{\mathrm{LD}} + \mathsf{B}_{\mathrm{LD}}\mathsf{K}_{\mathsf{d}})\,(0, \psi_d, 0, 0, 0)^{\mathrm{T}} = \mathbf{0}, \tag{4.188}$$

which is easily seen to be satisfied with $\mathsf{K}_{\mathsf{d}} = 0$. The feedback loop design is carried out by the linear quadratic regulator (LQR) method for both the regulator and the full-order observer (Kalman filter) with the following MATLAB-CST commands:

```
>> A = [ 0         0         0    1.0000    0
         0         0         0         0    1
    0.0361         0   -0.3248         0   -0.8293
         0         0  -46.0115   -6.2023    0.6608
         0         0   21.5518   -0.0757   -0.6311];

>> B = [ 0         0
         0         0
         0   -0.0892
  100.0271   -0.0343
    0.8910    0.6367];

>> C=[0 1 0 0 0; 0 0 0 0 1] % Heading angle and yaw rate outputs
```

```
>> damp(A)

        Eigenvalue            Damping     Freq. (rad/s)

  0.00e+000                 -1.00e+000      0.00e+000
 -4.58e-003                  1.00e+000      4.58e-003    % Spiral
 -4.35e-001 + 4.29e+000i     1.01e-001      4.31e+000    % Dutch-Roll
 -4.35e-001 - 4.29e+000i     1.01e-001      4.31e+000    %
 -6.28e+000                  1.00e+000      6.28e+000    % Pure roll

% Regulator gain matrix by LQR:
>> K=lqr(A,B,0.1*eye(5),eye(2))

K =
    0.3278    0.2564   -0.0895    0.2703    0.0126
   -0.0095   -0.1851   -0.3526   -0.0002    0.0330

>> Lp=lqr(A',C',eye(5),eye(2));L=Lp' % Kalman filter gain

L =
    0.2010    0.1744
    1.2906    0.4218
   -0.0637    0.4145
    0.1586   -3.4965
    0.4218    3.7978

% Augmented plant with aileron and rudder servos:
>> Abar=[A B;zeros(1,5) -20 0; zeros(1,5) 0 -20];
>> Bbar=[0 0 0 0 0 20 0;0 0 0 0 0 0 20]';
>> Cbar=[C zeros(2,2)];

>> Kd=zeros(2,5) % Feedforward gain matrix

>> Ac=[Abar  -Bbar*K;L*Cbar A-L*C-B*K]; %Closed-loop dynamics matrix

>> Bc=[Bbar*(K+Kd); B*(K+Kd)] % Closed-loop input coefficients matrix
>> damp(Ac) % Closed-loop eigenvalues

        Eigenvalue            Damping     Freq. (rad/s)
 -3.22e+001                  1.00e+000      3.22e+001
 -2.34e+000 + 4.82e+000i     4.36e-001      5.35e+000
 -2.34e+000 - 4.82e+000i     4.36e-001      5.35e+000
 -5.01e-001 + 4.28e+000i     1.16e-001      4.31e+000
 -5.01e-001 - 4.28e+000i     1.16e-001      4.31e+000
 -6.30e+000                  1.00e+000      6.30e+000
 -3.30e-002                  1.00e+000      3.30e-002
 -5.28e-002                  1.00e+000      5.28e-002
 -9.74e-001                  1.00e+000      9.74e-001
 -1.25e+000                  1.00e+000      1.25e+000
 -2.01e+001                  1.00e+000      2.01e+001
 -2.00e+001                  1.00e+000      2.00e+001

% Closed-loop system:
>> sys=ss(Ac,Bc(:,2),[0 1 zeros(1,10)],0)

a =
           x1         x2         x3         x4         x5         x6         x7         x8         x9
  x1        0          0          0          1          0          0          0          0          0
  x2        0          0          0          0          1          0          0          0          0
  x3   0.0361          0    -0.3248          0    -0.8293          0    -0.0892          0          0
  x4        0          0     -46.01     -6.202     0.6608        100    -0.0343          0          0
  x5        0          0      21.55    -0.0757    -0.6311      0.891     0.6367          0          0
  x6        0          0          0          0          0        -20          0     -1.681     -19.79
  x7        0          0          0          0          0          0        -20     0.1912      2.904
  x8        0    0.05181          0          0      1.836          0          0          0    -0.05181
  x9        0      3.305          0          0     0.7279          0          0          0     -3.305
  x10       0   0.006786          0          0      2.302          0          0    0.03525    -0.01974
  x11       0    -0.3283          0          0     -12.09          0          0     -8.406     -98.64
  x12       0     0.7279          0          0      9.901          0          0    -0.06879     -1.517
```

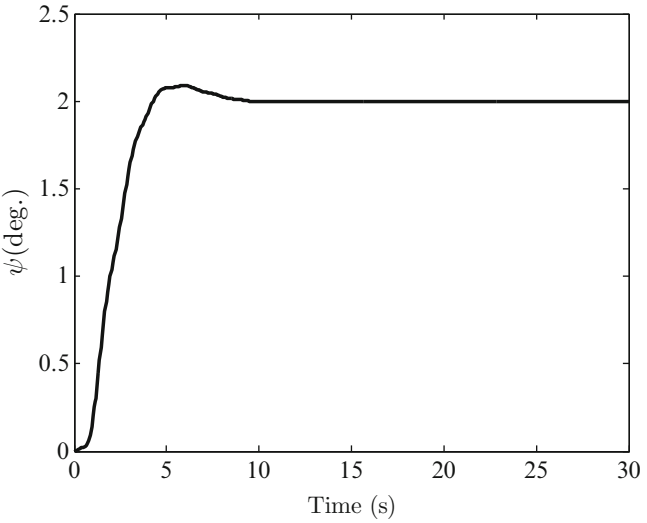


Fig. 4.41 Closed-loop heading angle step response of a fighter aircraft equipped with a heading autopilot to a commanded heading change by 2°

```
             x10         x11         x12
    x1         0           0           0
    x2         0           0           0
    x3         0           0           0
    x4         0           0           0
    x5         0           0           0
    x6    -18.97     -0.2531     0.04754
    x7     7.317     0.02493     -0.8553
    x8         0           1      -1.836
    x9         0           0      0.2721
    x10 -0.3574 -0.0001112      -3.127
    x11  -140.9      -7.468       12.99
    x12   20.94    -0.08618      -10.56

b =
              u1
    x1         0
    x2         0
    x3         0
    x4         0
    x5         0
    x6     19.79
    x7    -2.904
    x8         0
    x9         0
    x10  0.01295
    x11    98.97
    x12   0.7891

c =
          x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11  x12
    y1     0    1    0    0    0    0    0    0    0    0    0    0

d =
          u1
    y1     0

Continuous-time model.
>> [y,t,x]=step(sys,30); % Closed-loop step response

>> y=y*pi/90; x=x*pi/90; % Scaling for the 2 deg. step heading command
```

The resulting step response of the closed-loop system shows that the commanded heading angle is reached within 8 s (Fig. 4.41), with the maximum aileron and rudder deflections of 1.65° and −0.25°, respectively (Fig. 4.42). The other closed-loop state variables – plotted in Fig. 4.43 – show a rapid initial rolling motion, with

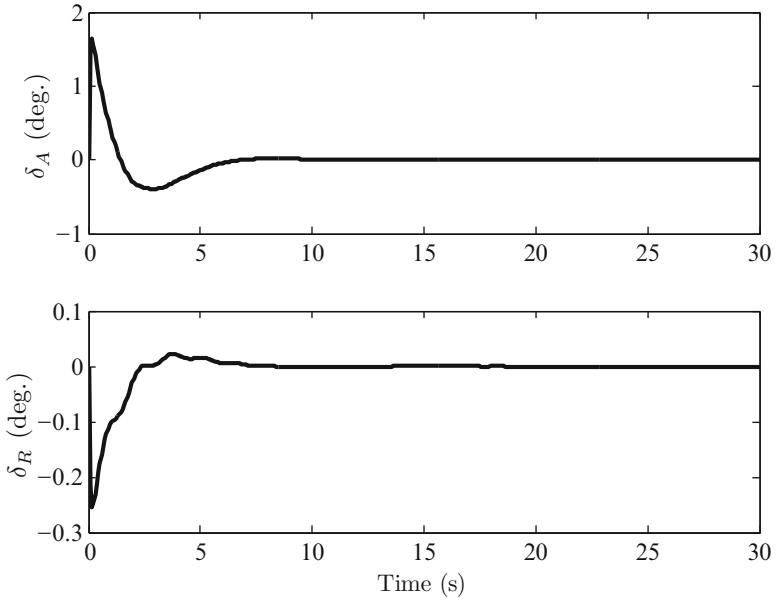


Fig. 4.42 Closed-loop aileron and rudder step response of a fighter aircraft equipped with a heading autopilot to a commanded heading change by 2°

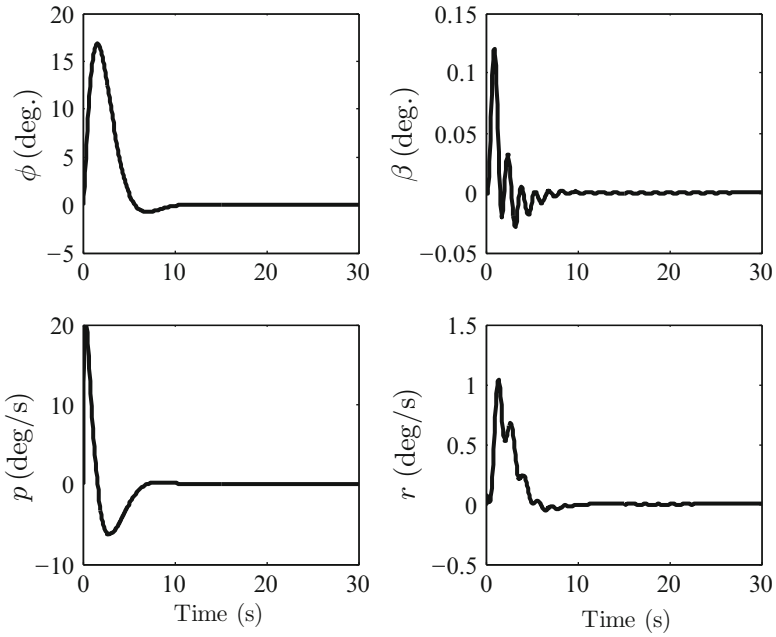


Fig. 4.43 Closed-loop step response of a fighter aircraft equipped with a heading autopilot to a commanded heading change by 2°

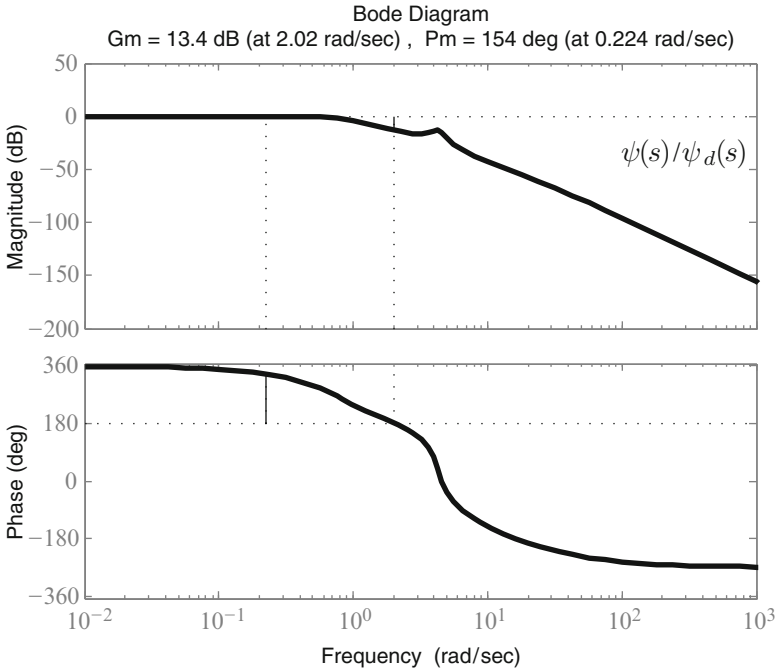


Fig. 4.44 Closed-loop frequency response, $\phi(s)/\phi_d(s)$, of a fighter aircraft equipped with a heading autopilot

much smaller sideslip and yaw rate corrections. The closed-loop system's gain and phase margins are shown in Fig. 4.44 to be 13.4 dB and 154°, respectively, thereby indicating a fair overall robustness. The roll-off with respect to high-frequency measurement noise is close to −40 dB per decade.

4.8 Summary

Aircraft attitude dynamics consists of both rotational and translational motions. For stability and control analysis, only small perturbations from an equilibrium condition need be studied, which lead to an uncoupling of motion taking place within the plane of symmetry (longitudinal dynamics) and that occurring outside it (lateral-directional dynamics). The forces and moments are then approximated by first-order Taylor series expansions about the equilibrium condition, resulting in linear coefficients called stability and control derivatives. Control of longitudinal dynamics is carried out by engine throttle and elevator, each of which has a servo actuator. The throttle input largely affects the long-period (phugoid) mode, whereas the short-period mode can be effectively controlled only by the elevator input. While

single-variable control of airspeed and pitch attitude, respectively, is possible, a more robust design consists both the inputs applied simultaneously, resulting in an active damping of both phugoid and short-period modes was studied. High-speed aircraft experience a marked change of stability derivatives with the altitude and Mach number, such as a decline in the pitch and yaw damping derivatives. Therefore, in order to maintain adequate stability margin throughout the flight envelope, a variation of the feedback gains with the Mach number and dynamic pressure (called gain scheduling) must be allowed. However, this results in an overall nonlinear control system, which must be designed with proper care. The aileron is the primary lateral control for roll angle tracking, whereas rudder is mainly used for directional control by regulating sideslip and yaw rate. Due to an inherent coupling of roll, yaw, and sideslip motions, aileron and rudder can be used individually as single inputs for controlling lateral-directional dynamics. The pure rolling and the Dutch roll modes can be controlled by single feedback loops through aileron and rudder, respectively. However, control of the spiral mode requires a multivariable design approach with both aileron and rudder inputs.

Exercises

4.1. The aircraft given in Example 4.4 has the following additional parameters:

$$m = 85{,}000 \text{ kg}, \quad \bar{c} = 6 \text{ m}, \quad S = 225 \text{ m}^2, \quad J_{yy} = 3.6 \times 10^6 \text{ kg m}^2.$$

If the flight condition in Example 4.4 is at standard sea level ($\rho = 1.225$ kg/m^3), determine the nondimensional longitudinal state dynamics matrix, $\hat{\mathbf{A}}_{\text{Long}}$, the natural frequencies and damping ratios of the longitudinal modes in the controls fixed case. What differences (if any) do you observe compared to the numbers derived in the example?

4.2. Derive the approximate short-period dynamic equations of an aircraft in straight and level flight by neglecting the airspeed variations in the longitudinal dynamics plant. Obtain the approximate short-period natural frequency and damping for the aircraft of Example 4.4, and compare them with the exact number computed in the example.

4.3. Using the data of Example 4.4, carry out an eigenvector analysis (in a manner similar to that presented in Example 4.16) to show that the phugoid mode is indeed dominated by airspeed and pitch angle variations, with a negligible change in the angle-of-attack, while the short-period mode is primarily affected by angle-of-attack and pitch variations.

4.4. Show that the longitudinal plant is controllable with the elevator input alone.

4.5. Show that the longitudinal plant is observable with the pitch rate output alone.

4.6. Re-design the airspeed and altitude hold autopilot of Example 4.9 based upon a full-order observer with airspeed and altitude outputs and the following observer poles: $s_1 = -0.5$, $s_{2,3} = -0.5 \pm 0.5\text{i}$ and $s_{4,5} = -1 \pm \text{i}$. The regulator design is unchanged from that of Example 4.9. What differences are seen in the closed-loop initial response compared to that of Example 4.9?

4.7. Suppose the airspeed hold requirement is removed from the autopilot design of Example 4.9. Using a suitable single output, re-design the observer by pole-placement and analyze the performance and robustness of the resulting closed-loop system.

4.8. For the ILS approach system designed in Example 4.10, find the closed-loop transfer function, $\theta(s)/\delta_{\text{Ec}}(s)$, and compute its frequency response. What are the gain and phase margins of the system?

4.9. Suppose the initial glideslope error is increased to $\theta = 10°$ due to a delay in intercepting the ILS signal for the automatic approach system designed in Example 4.10. Using a *Simulink* block diagram, simulate the closed-loop initial response with elevator servo saturation limits $\pm 25°$ included in the model. What (if any) steady-state error can be observed in the closed-loop pitch angle response?

4.10. For the airplane of Example 4.9, design an automatic landing system with an initial pitch angle of $-3°$, initial approach speed, $U = 55$ m/s, and stalling speed at standard sea level, $U = 42.27$ m/s. Since the engines are maintained to a constant thrust, elevator is the only control input. A radar altimeter provides feedback of altitude above the runway. In addition, airspeed data is also available for measurement and feedback.

4.11. For the PI airspeed control system with throttle input designed in Example 4.12, determine the closed-loop transfer function, $\theta(s)/u_d(s)$, and its frequency response. Plot the pitch angle response to a unit step change in desired airspeed. What (if any) change can be observed in the angle-of-attack, $\alpha(s)/u_d(s)$?

4.12. Re-design the airspeed control system of Example 4.12 with throttle input using a state-space, full-order observer based feedback and feedforward control instead of the PI system. Is it possible to have a zero steady-state error for such a system? Why? Plot the airspeed and pitch angle response to a unit step change in desired airspeed.

4.13. Build a *Simulink* block-diagram in order to simulate the multi-input airspeed control system of Example 4.13 for a climb with a constant indicated airspeed, $U_i = 55$ m/s, beginning from the standard sea level to an altitude of 3 km. Use the following variation of atmospheric density in kg/m^3 with altitude in m [21]:

$$\rho(h) = 1.225\left(1 - \frac{0.0065h}{288.15}\right)^{4.258644}.$$

4.14. Repeat Exercise 4.13 for a climb with a constant Mach number, $\mathcal{M} = 0.16$, beginning from the standard sea level to an altitude of 3 km. Use $\gamma = 1.4$, $\bar{R} = 287$ J/kg/K, and the following variation of atmospheric temperature in K with altitude in m [21]:

$$\bar{T}(h) = 288.15 - 0.0065\,\mathrm{h}.$$

4.15. Derive the feedforward gain equation, (4.166), for airspeed tracking system for a constant commanded airspeed, u_d, (other perturbations being zero) and show that it does not have any solution unless $Z_T/m = M_T/m = 0$.

4.16. Using the aircraft data from Example 4.9, design a suitable autopilot for tracking the altitude profile of Example 4.8.

4.17. Using the wing semi-span, $b/2$, as the characteristic length, derive the nondimensional, lateral-directional state equations.

4.18. Show that the general lateral-directional plant is controllable with the aileron input alone.

4.19. Is the general lateral-directional plant controllable with the rudder input alone?

4.20. For the aircraft of Example 4.18 design a full-order observer and regulator by the LQR method with appropriate choice of cost parameters based on yaw angle feedback and rudder control input. It is required that the spiral mode should be stabilized and the rates and sideslip angle should settle to zero within 20 s for an initial 5° sideslip perturbation, without requiring a rudder deflection greater than ±25°. Plot the closed-loop initial response and determine the gain and phase margins of the closed-loop system.

4.21. Repeat the design of the yaw damper of Example 4.18 for the level flight condition, $\Theta_e = 0$, at the given airspeed and altitude. What differences (if any) do you observe with the yaw damper system designed in Example 4.18 for a 30° climb?

4.22. An aircraft has the following lateral-directional dynamics plant in straight and level flight with airspeed 175 m/s, where all the variables are in radian:

$$\mathsf{A}_{\mathrm{LD}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.07 & 0 & -0.13 & 0 & -0.99 \\ 0 & 0 & -2 & -2.1 & 0.5 \\ 0 & 0 & 1.6 & -0.08 & -0.25 \end{pmatrix}$$

$$\mathsf{B}_{\mathrm{LD}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -0.03 \\ 4.5 & -0.075 \\ -0.05 & 1.2 \end{pmatrix}$$

In addition, the aileron and rudder servos are of first-order with the same time constant of 0.1 s.

(a) Find the lateral-directional modes of this aircraft.
(b) Design a heading autopilot for this plant such that a lateral disturbance of $\beta(0) = 5^\circ$ is dissipated within 15 s without the two control inputs exceeding $\pm 25^\circ$.
(c) Test the design carried out in Part (b) with a step desired heading change of 5°. Plot the closed-loop response of heading, bank, and sideslip angles as well as the rudder and aileron deflections.

4.23. Design a suitable Dutch-Roll damper for the aircraft of Exercise 4.22 using only the yaw rate output and rudder input. Test the design for the initial perturbation specified in Part (b) of Exercise 4.22.

Chapter 5
Automatic Control of Rockets

5.1 Aims and Objectives

- To model a rocket's attitude plant with gimbaled nozzle and reaction jet actuators
- To design and analyze a rocket's roll, pitch, and yaw control systems using single-variable and multi-variable methods

5.2 Introduction

It is crucial for a rocket to be always aligned with the velocity vector due to aerodynamic load considerations. It is easy to appreciate the fragility of the thin structure of a rocket filled with liquid propellants – much like an elongated aluminum can of soda – for which any significant transverse acceleration due to lift (or sideforce) can prove to be destructive. The solid propellant rockets are structurally more robust, but suffer from another fallibility common to all rockets: inherent aerodynamic instability with respect to angle-of-attack, α, (or sideslip, β). A small error in either α or β – if uncorrected – can quickly build-up, leading to excessive transverse loads that are certain to destroy the vehicle. For this reason, all rockets are axisymmetric in shape and mass distribution, which enables an equilibrium flight condition without a net lift and sideforce. In order to correct attitude deviations from equilibrium, most rockets are also equipped with an attitude control system for making the vehicle follow the desired flight path, such that angle-of-attack and sideslip angle are always zero. Such a control system applies its control forces and moments through aerodynamic control surfaces and thrust deflection. Rocket maneuvers are performed by generating a pure torque (without a net lateral force), which causes the large rocket thrust to act in a new direction. Smaller rockets – such as air-to-air missiles – have movable fins mounted fore and aft of the center of mass in a way similar to aircraft lifting surfaces. However, the fins are much smaller than the wings and tails of an aircraft, because they are intended

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_5,

for stability and control rather than for lift generation. For achieving longitudinal (pitch-yaw) stability at lower altitudes, large rockets – such as launch vehicles and ballistic missiles – are also fitted with small aerodynamic fins near the base. If made movable, the fins can apply small control torques in the denser region of the atmosphere (such as in the case of German *V-2* rocket of WW-II). However, since a launch vehicle operates at a wide range of speeds and altitudes, one cannot rely upon the small aerodynamic moments generated by movable fins to control the vehicle throughout its trajectory. Instead, the primary control torque is generated by a combination of thrust-vectoring, i.e., by changing the direction of the thrust vector relative to the axis of symmetry, and through cold gas reaction jets firing normal to the axis. Some vehicles also have separate (but small) vernier rockets for making fine adjustments in overall thrust magnitude and direction.

All rockets are statically unstable in pitch and yaw ($M_\alpha > 0, N_\beta < 0$). They are also unstable in roll when not stabilized by large aerodynamic fins. Hence, rockets require active stabilization via a fully gimbaled, gyro-stabilized platform called an *inertial measurement unit* (IMU) (Chap. 2) consisting of pendulum switches, rate gyros, and accelerometers. The IMU can sense multi-axis angle and angular rate deviations and thus lies at the heart of the rocket's control system. Digital electronics have enabled the replacement of the fully gimbaled platforms by *strap-on IMU* units rigidly attached to the vehicle for better resolution, ruggedness, and smaller weight. These units have fewer moving parts, and mainly rely upon solid-state accelerometers, tuning forks, quartz oscillators, and ring laser gyros. The guidance system continuously sends attitude commands to be tracked by the controller. With the available commands and actual sensor data, the attitude controller drives the nozzle gimbals and/or reaction gas jets to nullify the attitude errors. Apart from sending the nominal attitude, the guidance system also performs minor trajectory corrections by directly driving the gimbal actuators.

5.2.1 Thrust Vectoring for Attitude Control

Creation of attitude control torque by thrust vectoring requires a balancing of the transverse forces. Thus, a minimum number of two equal and opposite control forces is necessary to form a pure couple. While a short-range missile operating entirely inside the lower atmosphere can have the control force produced by an aerodynamic fin to oppose that produced by vectored thrust, this is not feasible for a gravity-turn launch vehicle. In a launch vehicle, thrust-vectoring can be mainly achieved in three distinct ways:[1] (a) gimbaling of a single main engine balanced by reaction jets or vernier rockets, (b) gimbaling of two or more main engines, and (c) differential

[1]Some short-range rockets – such as the German *V-2* and the Russian *SCUD* – alternatively use graphite vanes at nozzle exit to generate limited control moments by rotation and deflection of the exhaust gases.

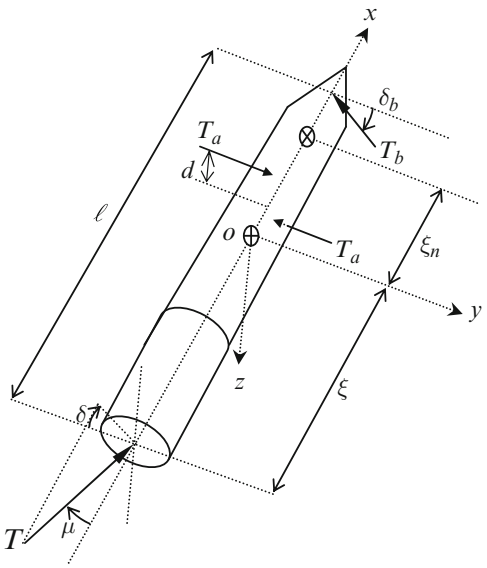


Fig. 5.1 Thrust-vectoring geometry for a rocket equipped with a single gimbaled engine, a balancing reaction jet, and a pair of roll reaction jets

thrust from gimbaled nozzles. Some launch vehicles (such as the erstwhile "Space Shuttle" of NASA) use (a) and (b) alternatively in different stages. The concept of differential thrust for attitude control has not been used due to practical limitations.

5.2.1.1 Reaction Jet System

Gimbaling a single rocket engine can produce pitching and yawing control moments about the center of mass. However, for static stability, one must also have a set of smaller vernier rockets (or gas jets) mounted on the other side of the center of mass that are capable of balancing the sideforce generated by the gimbaled nozzle, by creating a small thrust of their own. In the absence of a balancing thrust, the vehicle has a tendency to move sideways in a destructive manner, if not controlled by a closed-loop system. Consider a rocket with principal body-fixed frame, $oxyz$. Thrust vectoring consists of a gimbaled rocket nozzle at the base that swivels in any desired direction, δ, measured from oz, and deflects by a small thrust angle, μ, measured from ox as shown in Fig. 5.1. The distance of gimbaled nozzle is ξ from the center of mass o and ℓ from the balancing jet. The reaction jets/vernier rockets can be mounted in a ring-like fashion to provide transverse force balance for a particular swivel angle, δ. Assuming a small thrust angle, μ, and a balancing thrust, T_b, normal to the axis ox in the direction δ_b (Fig. 5.1), we have the following pitch-yaw control torque about o:

$$\boldsymbol{\tau} \simeq [T\mu\xi \sin\delta + T_b(\ell - \xi)\sin\delta_b]\,\mathbf{j} - [T\mu\xi\cos\delta + T_b(\ell - \xi)\cos\delta_b]\,\mathbf{k}. \quad (5.1)$$

Force balance requires

$$T\mu\,(\mathbf{j}\cos\delta + \mathbf{k}\sin\delta) = T_b\,(\mathbf{j}\cos\delta_b + \mathbf{k}\sin\delta_b)\,. \tag{5.2}$$

When force balance is not carried out, we have $T_b = 0$, thereby producing a net transverse force

$$\mathbf{F} \simeq T\mu\,(\mathbf{j}\cos\delta + \sin\delta\mathbf{k})\,, \tag{5.3}$$

and the pitch-yaw torque becomes

$$\boldsymbol{\tau} = T\mu\xi\,(\mathbf{j}\sin\delta - \mathbf{k}\cos\delta)\,. \tag{5.4}$$

For simplicity of notation, we define *effective deflection angles*,

$$\begin{aligned}\mu_1 &= \mu\sin\delta\\ \mu_2 &= \mu\cos\delta,\end{aligned} \tag{5.5}$$

which make the transverse force and pitch-yaw control torque the following linear functions for given values of (T, ξ):

$$\begin{aligned}\mathbf{F} &= T\,(\mu_2\mathbf{j} + \mu_1\mathbf{k})\\ \boldsymbol{\tau} &= T\xi\,(\mu_1\mathbf{j} - \mu_2\mathbf{k})\,.\end{aligned} \tag{5.6}$$

In the absence of force balance, the angles μ, δ, must be continuously manipulated by an active control system to maintain a nominal attitude. Instead, for achieving transverse control force balance, we must have

$$T\mu\cos\delta = T_b\cos\delta_b, \tag{5.7}$$

and

$$T\mu\sin\delta = T_b\sin\delta_b, \tag{5.8}$$

which implies

$$\delta = \delta_b. \tag{5.9}$$

Thus, when the transverse force balance is carried out, the control torque is given by

$$\boldsymbol{\tau} = T\mu\ell\,(\mathbf{j}\sin\delta - \mathbf{k}\cos\delta)\,, \tag{5.10}$$

or

$$\boldsymbol{\tau} = T\ell\,(\mu_1\mathbf{j} - \mu_2\mathbf{k})\,. \tag{5.11}$$

Clearly, for a given thrust angle, μ, the ratio of pitching and yawing moments can be adjusted by varying the swivel angle, δ. For a nominal gravity-turn trajectory, we select oxz as the vertical plane. Hence, $\delta = \pi/2$ and we have a pure pitching moment. Note that the control torque with force balance is independent of the location of center of mass, o, that varies as the propellant is consumed. In such a case, the attitude control-law need not adapt with respect to center of mass variation.

In order to correct small, off-nominal deviations from the vertical plane, there must be an additional capability to generate a small rolling control moment by a pair of roll reaction jets or vernier rockets of thrust, T_a, applied at an offset, d, from ox (Fig. 5.1). The net control torque is thus given by

$$\boldsymbol{\tau} = 2T_a d\,\mathbf{i} + T\xi\,(\mu_1\mathbf{j} - \mu_2\mathbf{k})\,, \tag{5.12}$$

with the understanding that ξ should be replaced by ℓ in case of transverse force balance. Due to the fact that $T_a \ll T$, $d \ll \xi$, the rolling control moment is generally an order of magnitude smaller than the pitching and yawing moments. Thus, a launch vehicle equipped with reaction jet control system cannot tolerate significant off-nominal roll deviations.

5.2.1.2 Multiple Gimbaled Nozzle System

Two (or more) rocket engines can be simultaneously gimbaled by different angles to create a control torque without an unbalanced sideforce. In addition, by having a large difference in the gimbal angles, a significant rolling moment can be generated which is not feasible in a reaction jet system. Consider a rocket with a pair of gimbaled rocket nozzles of thrust T_a and T_b, respectively, symmetrically located about ox at the base, with a spacing $2d$ between them. The line joining the centers of the nozzles makes an angle σ with oy, as shown in Fig. 5.2. The nozzles can swivel in desired directions about the longitudinal axis, ox, by angles (δ_a, δ_b) and by small deflection angles (μ_a, μ_b) about the yaw axis, oz (Fig. 5.2). The net transverse force and torque produced by the two nozzles are expressed as follows:

$$\mathbf{F} \simeq (T_a\mu_a \cos\delta_a - T_b\mu_b \cos\delta_b)\mathbf{j} + (T_a\mu_a \sin\delta_a - T_b\mu_b \sin\delta_b)\mathbf{k}. \tag{5.13}$$

$$\begin{aligned}
\boldsymbol{\tau} \simeq\ & [-\xi\mathbf{i} - d(\cos\sigma\mathbf{j} + \sin\sigma\mathbf{k})] \times T_a(\mathbf{i} + \mu_a \cos\delta_a\mathbf{j} + \mu_a \sin\delta_a\mathbf{k}) \\
& + [-\xi\mathbf{i} + d(\cos\sigma\mathbf{j} + sin\sigma\mathbf{k})] \times T_b(\mathbf{i} - \mu_b \cos\delta_b\mathbf{j} - \mu_b \sin\delta_b\mathbf{k}) \\
=\ & d\,[\sin\sigma\,(T_a\mu_a \cos\delta_a + T_b\mu_b \cos\delta_b) - \cos\sigma\,(T_a\mu_a \sin\delta_a + T_b\mu_b \sin\delta_b)]\,\mathbf{i} \\
& + [d\,\sin\sigma(T_b - T_a) + \xi\,(T_a\mu_a \sin\delta_a - T_b\mu_b \sin\delta_b)]\,\mathbf{j} \\
& + [d\,\cos\sigma(T_a - T_b) + \xi\,(T_b\mu_b \cos\delta_b - T_a\mu_a \cos\delta_a)]\,\mathbf{k}.
\end{aligned}$$

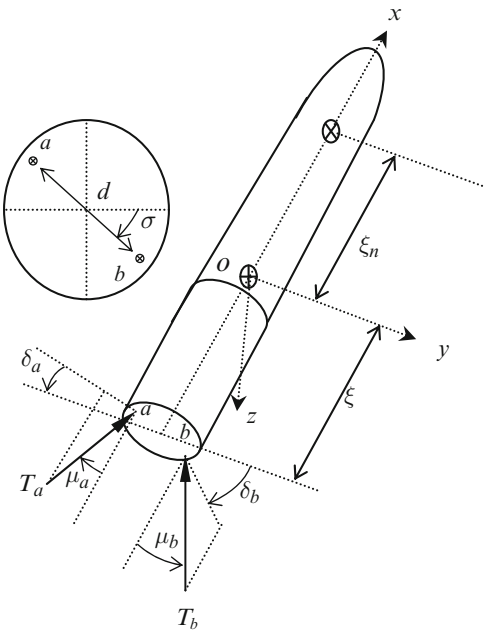


Fig. 5.2 Thrust vectoring geometry for a rocket equipped with a pair of gimbaled engines at the base

For simplifying the notation, we define *effective deflection angles* as

$$\begin{aligned}
\nu_1 &= \frac{1}{T}\left(T_a \mu_a \sin\delta_a - T_b \mu_b \sin\delta_b\right) \\
\nu_2 &= \frac{1}{T}\left(T_a \mu_a \cos\delta_a - T_b \mu_b \cos\delta_b\right) \\
\nu_3 &= \frac{1}{T}\left(T_a \mu_a \sin\delta_a + T_b \mu_b \sin\delta_b\right) \\
\nu_4 &= \frac{1}{T}\left(T_a \mu_a \cos\delta_a + T_b \mu_b \cos\delta_b\right),
\end{aligned} \tag{5.14}$$

which reduce the transverse force to the form of (5.6),

$$\mathbf{F} = T\left(\nu_2 \mathbf{j} + \nu_1 \mathbf{k}\right) \tag{5.15}$$

while the control torque is given by

$$\begin{aligned}
\boldsymbol{\tau} = {} & Td\left(\nu_4 \sin\sigma - \nu_3 \cos\sigma\right)\mathbf{i} + \left[-T\xi\nu_1 - d\,\sin\sigma\left(T_a - T_b\right)\right]\mathbf{j} \\
& + \left[-T\xi\nu_2 + d\,\cos\sigma\left(T_a - T_b\right)\right]\mathbf{k}.
\end{aligned} \tag{5.16}$$

Clearly, without force balance, the control torque is dependent on the changing location of center of mass, ξ. Most rocket engines are incapable of throttling fast

enough to rapidly achieve a thrust differential, $(T_a - T_b)$, required for attitude control. Thus, we generally have $T_a = T_b = T/2$ and the control torque is entirely produced by gimbaling. Almost all launch vehicles use multiple, gimbaled nozzles. The largest such system actually flown was the first stage of the *Saturn-V* rocket of NASA's *Apollo* program, in which four outboard engines achieved attitude control via gimbaling, but without differential thrust. Therefore, transverse force balance could not be achieved when pitch-yaw control was carried out.

5.2.1.3 Gimbaling with Differential Thrust

Suppose a futuristic rocket engine has the combined capability of gimbaling as well as rapid differential thrust of its two nozzles. Then transverse force balance dictates that $\nu_1 = \nu_2 = 0$, i.e.,

$$T_a \mu_a \cos\delta_a = T_b \mu_b \cos\delta_b, \tag{5.17}$$

and

$$T_a \mu_a \sin\delta_a = T_b \mu_b \sin\delta_b, \tag{5.18}$$

which have several possible solutions, the trivial one being

$$\begin{aligned} T_a &= T_b \\ \delta_a &= \delta_b \\ \mu_a &= \mu_b. \end{aligned} \tag{5.19}$$

In order to achieve force balance for an arbitrary control torque, the actuation of the nozzles must be carried out in unison such that (5.17) and (5.18) are satisfied. Clearly, a rigid axle connecting the nozzles, or a fixed gearing arrangement between them is ruled out. There must be a nonlinear, feedback actuator that generates the net thrust and the commanded torque by suitably adjusting the thrust magnitudes and deflection angles of the nozzles. For a total thrust, T, with transverse force balance, the respective nozzle thrusts are related by

$$T_a + T_b \simeq T, \tag{5.20}$$

and the net control torque is given by

$$\boldsymbol{\tau} = 2T_a \mu_a d\,\mathbf{i} \sin(\sigma - \delta_a) + d(T - 2T_a)\left[\mathbf{j}\sin\sigma - \mathbf{k}\cos\sigma\right]. \tag{5.21}$$

In order to conform with the conventional terminology, we can define *effective deflection angles* by

$$\begin{aligned} \mu_1 &= d\left(1 - 2\frac{T_a}{T}\right)\sin\sigma \\ \mu_2 &= d\left(1 - 2\frac{T_a}{T}\right)\cos\sigma, \end{aligned} \tag{5.22}$$

and the control rolling moment by

$$\tau_x = 2T_a \mu_a d. \tag{5.23}$$

Note that with force balance, the pitching and yawing moments are entirely generated by thrust differential, $(T_a - T_b)$, while the rolling moment is produced by the thrust deflection, μ_a. By having a differential thrust modulation, $(T_a - T_b)$, a much larger range of pitching and yawing moments can be generated compared with the reaction jet as well as gimbaled nozzle (without differential thrust) systems.

In summary, for a generic thrust-deflection model without either force balance or thrust differential, we have the following control forces and moments:

$$\begin{aligned} F_y &\simeq T\mu_2 \\ F_z &\simeq T\mu_1 \\ \tau_y &\simeq T\xi\mu_1 \\ \tau_z &\simeq -T\xi\mu_2. \end{aligned} \tag{5.24}$$

With force balance, we have

$$\begin{aligned} F_y &\simeq 0 \\ F_z &\simeq 0 \\ \tau_y &\simeq T\ell\mu_1 \\ \tau_z &\simeq -T\ell\mu_2. \end{aligned} \tag{5.25}$$

The expression for rolling moment, τ_x, depends upon whether the reaction jet or gimbaling with differential thrust is used.

5.3 Attitude Control Plant

Consider the following translational kinetics equations for a rocket, which are the same as those derived in Chap. 4 for an aircraft:

$$\begin{aligned} X - mg\sin\Theta &= m(\dot{U} + QW - RV) \\ Y + mg\sin\Phi\cos\Theta &= m(\dot{V} + RU - PW) \\ Z + mg\cos\Phi\cos\Theta &= m(\dot{W} + PV - QU), \end{aligned} \tag{5.26}$$

where the Euler angle sequence, $(\Psi)_3, (\Theta)_2, (\Phi)_1$, represents the orientation of the principal body frame, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, with respect to an inertial reference frame, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$. However, we are no longer making the assumption that $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ is approximated by

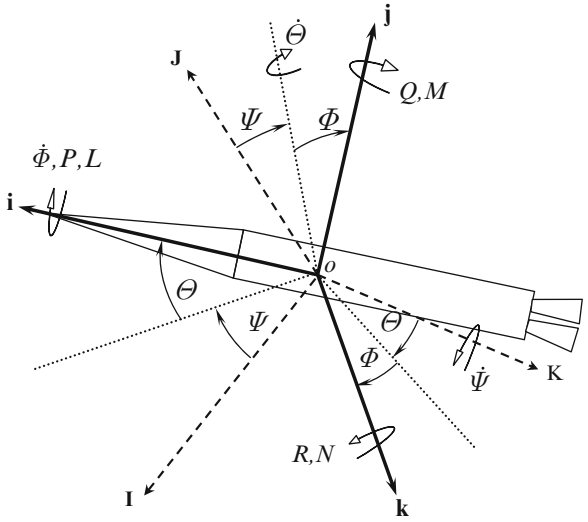


Fig. 5.3 Attitude dynamics of a rocket

the local horizon frame (which we did for the case of aircraft). The typical speed and altitude of a rocket are much higher than those of the aircraft. Consequently, the acceleration terms due to flight path curvature cannot be neglected, and the angular velocity vector,

$$\boldsymbol{\omega} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}, \tag{5.27}$$

is measured in the inertial frame. The form of the rotational kinematics equations are unchanged from Chap. 4,

$$\begin{Bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{Bmatrix} = \frac{1}{\cos\Theta} \begin{pmatrix} \cos\Theta & \sin\Phi\sin\Theta & \cos\Phi\sin\Theta \\ 0 & \cos\Phi\cos\Theta & -\sin\Phi\cos\Theta \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix}, \tag{5.28}$$

provided we remember that the roll angle, Φ, the pitch angle, Θ, and the yaw angle, Ψ, are now measured from the inertial (rather than local horizon) frame.

Another crucial difference between a rocket and an aircraft is that the rocket has a principal axis of symmetry, $\mathbf{i}$, which yields $J_{yy} = J_{zz}$ and $J_{xy} = J_{xz} = J_{yz} = 0$. The attitude dynamics of rockets is greatly simplified by the presence of the axis of symmetry, ox, (Fig. 5.3), resulting in the following rotational kinetics equations:

$$L = J_{xx}\dot{P}, \tag{5.29}$$

$$M = J_{yy}\dot{Q} + \left(J_{xx} - J_{yy}\right)PR, \tag{5.30}$$

$$N = J_{yy}\dot{R} + \left(J_{yy} - J_{xx}\right)PQ. \tag{5.31}$$

5.3.1 Equilibrium Conditions and Small Perturbations

It is clear from (5.26)–(5.31) that the rocket flight equations have the following equilibrium solutions (denoted by the subscript e):

(a) Steady Rolling Motion: $V_e = W_e = Q_e = R_e = 0$ and U_e, P_e constants. The resulting forces and moments at equilibrium are given by

$$\begin{aligned} X_e - mg\sin\Theta_e &= 0 \\ Y_e + mg\sin\Phi_e\cos\Theta_e &= 0 \\ Z_e + mg\cos\Phi_e\cos\Theta_e &= 0 \\ \dot{\Phi}_e &= P_e \\ \dot{\Theta}_e &= 0 \\ \dot{\Psi}_e &= 0, \end{aligned} \tag{5.32}$$

and $L_e = M_e = N_e = 0$.

(b) Steady Pitching Motion: $\Phi_e = V_e = W_e = P_e = R_e = 0$ and U_e, Q_e constants. The resulting forces and moments at equilibrium are given by

$$\begin{aligned} X_e - mg\sin\Theta_e &= 0 \\ Y_e &= 0 \\ Z_e + mg\cos\Theta_e &= -mQ_eU_e \\ \dot{\Phi}_e &= 0 \\ \dot{\Theta}_e &= Q_e \\ \dot{\Psi}_e &= 0, \end{aligned} \tag{5.33}$$

and $L_e = M_e = N_e = 0$. Since the pitch and yaw axes are indistinguishable from one another due to symmetry, here we have selected the body frame such that the constant angular velocity is given by $Q_e\mathbf{j}$. (We could have easily replaced the pitch axis with the yaw axis, and called the equilibrium solution the yawing motion. However, this is not the standard practice.)

In the linearized dynamic models given in (a) and (b) it is assumed that the aerodynamic force and moment perturbations are linearly dependent on the flow variables[2] (u, V, W, p, q, R) as well as the control inputs (μ_1, μ_2, τ_x). As for aircraft in Chap. 4, these relationships are given by the following truncated Taylor series expanded about a given equilibrium solution:

[2]Since $V = W = R = 0$ for each of the two equilibria, we will denote the perturbations in sideslip (V), downward speed (W), and yaw-rate, (R), by capital letters.

Thus, a rocket's attitude control system can be designed for maintaining the vehicle close to either of the two equilibrium points. As in the case of aircraft, the linearized equations of motion in terms of small perturbation variables, $u, v, w, \phi, \theta, \psi, p, q, R, \bar{X}, \bar{Y}, \bar{Z}$, are expressed as follows:

$$\begin{aligned}
\bar{X} &= X_u u + X_\alpha \alpha + X_{\dot{\alpha}} \dot{\alpha} + X_q q + X_{\mu_1} \mu_1 + X_{\mu_2} \mu_2 \\
\bar{Z} &= Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}} \dot{\alpha} + Z_q q + Z_{\mu_1} \mu_1 \\
\bar{M} &= M_u u + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q q + M_{\mu_1} \mu_1 \\
\bar{Y} &= Y_\beta \beta + Y_{\dot{\beta}} \dot{\beta} + Y_p p + Y_r R + Y_{\mu_2} \mu_2 \\
\bar{L} &= L_\beta \beta + L_{\dot{\beta}} \dot{\beta} + L_p p + L_r R + \tau_x \\
\bar{N} &= N_\beta \beta + N_{\dot{\beta}} \dot{\beta} + N_p p + N_r R + N_{\mu_2} \mu_2,
\end{aligned} \tag{5.34}$$

where

$$\alpha \simeq \frac{W}{U_e}; \qquad \beta \simeq \frac{V}{U_e}, \tag{5.35}$$

are the changes in the angle-of-attack and sideslip angle, respectively. Due to the absence of a significant lift (or sideforce), the unsteady aerodynamic stability derivatives, $X_{\dot{\alpha}}, Z_{\dot{\alpha}}, Y_{\dot{\beta}}$, and angular rate force derivatives, X_q, Z_q, Y_p, Y_r, are negligible for a rocket. Furthermore, the small fins do not cause a significant roll-yaw cross-coupling ($N_p \simeq 0, L_r \simeq 0$), or a roll-sideslip coupling ($L_\beta \simeq 0, L_{\dot{\beta}} \simeq 0$). Also due to the pitch-yaw symmetry, we have $Z_\alpha = Y_\beta$, $M_\alpha = -N_\beta$, and $M_q = N_r$. Finally, the change in forward force caused by a small thrust deflection can be assumed to be negligible, resulting in $X_{\mu_1} \mu_1 \simeq X_{\mu_2} \mu_2 \simeq 0$.

Note that the longitudinal forces and moments depend only upon the longitudinal variables, and lateral-directional forces and moments depend only upon the lateral-directional variables. Therefore, while the motion about the roll equilibrium is essentially an inertia coupled six degree-of-freedom motion, we have a decoupling of perturbation about the pure pitching equilibrium into *longitudinal* and *lateral* dynamics:

1. Motion about Roll Equilibrium:

$$X_u u + X_\alpha \alpha - mg\theta \cos \Theta_e = m\dot{u}$$

$$Z_\alpha \beta + Y_{\mu_2} \mu_2 + mg(\phi \cos \Phi_e \cos \Theta_e - \theta \sin \Phi_e \sin \Theta_e) = m(\dot{V} + RU_e - P_e W)$$

$$Z_u u + Z_\alpha \alpha + Z_{\mu_1} \mu_1 - mg(\phi \sin \Phi_e \cos \Theta_e + \theta \sin \Theta_e \cos \Phi_e) = m(\dot{W} + VP_e - qU_e)$$

$$\dot{\phi} = p + q \sin \Phi_e \tan \Theta_e + R \cos \Phi_e \tan \Theta_e$$

$$\dot{\theta} = q \cos \Phi_e - R \sin \Phi_e$$

$$\dot{\psi} = q \sin \Phi_e \sec \Theta_e + R \cos \Phi_e \sec \Theta_e$$

$$L_p p + \tau_x = J_{xx} \dot{p}$$

$$M_u u + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q q + M_{\mu_1} \mu_1 = J_{yy} \dot{q} + \left(J_{xx} - J_{yy}\right) P_\mathrm{e} R$$
$$-M_\alpha \beta - M_{\dot{\alpha}} \dot{\beta} + N_r R + N_{\mu_2} \mu_2 = J_{yy} \dot{R} + \left(J_{yy} - J_{xx}\right) P_\mathrm{e} q.$$

Clearly, we have a linear, time-varying system with $\Phi_\mathrm{e}(t)$ as a varying coefficient. Note that here we have $q = Q$.

2. Motion about Pitch Equilibrium:

(a) Longitudinal Dynamics:

$$\begin{aligned} X_u u + X_\alpha \alpha - mg\theta \cos \Theta_\mathrm{e} &= m(\dot{u} + Q_\mathrm{e} W) \\ Z_u u + Z_\alpha \alpha + Z_{\mu_1} \mu_1 - mg\theta \sin \Theta_\mathrm{e} &= m(\dot{W} - qU_\mathrm{e} - Q_\mathrm{e} u) \\ M_u u + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q q + M_{\mu_1} \mu_1 &= J_{yy} \dot{q} \\ \dot{\theta} &= q. \end{aligned} \tag{5.36}$$

(b) Lateral Dynamics:

$$\begin{aligned} \dot{\phi} &= p + R \tan \Theta_\mathrm{e} \\ \dot{\psi} &= R \sec \Theta_\mathrm{e} \\ Z_\alpha \beta + Y_{\mu_2} \mu_2 + mg\phi \cos \Theta_\mathrm{e} &= m(\dot{V} + RU_\mathrm{e}) \\ L_p p + \tau_x &= J_{xx} \dot{p} \\ -M_\alpha \beta - M_{\dot{\alpha}} \dot{\beta} + M_q R + N_{\mu_2} \mu_2 &= J_{yy} \dot{R} + \left(J_{yy} - J_{xx}\right) Q_\mathrm{e} p. \end{aligned} \tag{5.37}$$

Here, the system is linear and time-varying with $\Theta_\mathrm{e}(t)$ as the varying coefficient. Also, we have $p = P$.

5.3.2 Stability About Pitch Equilibrium

The stability of the two rocket dynamics equilibria is very important. If the motion is unstable, it quickly degenerates from being a pure rolling (or pitching) motion into a wildly tumbling and translatory motion, which ultimately leads to a wayward flight or structural failure. However, the perturbed motion can be stabilized by closed-loop control such that all perturbations other than the primary rotation are suppressed. If the perturbation variables, u, V, W, q, R, are brought to zero by a feedback regulator, a constant roll rate, P_e, could be maintained at a given speed, U_e, and pitch angle, Θ_e. Similarly, by bringing u, V, W, p, R, ϕ to zero, a constant pitch rate, Q_e, can be achieved at a given speed, U_e.

As we have seen above, there is an essential difference between motion about the two equilibria: motion about roll equilibrium is inherently coupled in all the degrees-of-freedom, whereas the motion about pitch equilibrium can be decoupled

into pitch dynamics and roll-yaw dynamics. Since control of coupled degrees-of-freedom is practically very difficult, the roll equilibrium is not feasible for a rocket in atmospheric flight (although it can be easily achieved in space flight (Chap. 6)). Thus, in this chapter we are essentially confined to the discussion of motion about the pitch equilibrium.

All large rockets have a dedicated roll control system that manages to keep the roll rate virtually zero. The pitch and yaw control systems are designed to track a small, reference pitch rate ($Q = Q_d(t)$) commanded by the guidance system, while keeping the yaw rate virtually zero. In such a case, the rocket's dynamic plant is further decoupled into what is typically called the *roll*, *pitch*, and *yaw* dynamics, given by the following (where $U \simeq U_e$):

1. Roll dynamics:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{p} \end{Bmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & \dfrac{L_p}{J_{xx}} \end{pmatrix} + \begin{pmatrix} 0 \\ \dfrac{1}{J_{xx}} \end{pmatrix} \tau_x. \tag{5.38}$$

2. Pitch dynamics:

$$\begin{Bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{Bmatrix} = \begin{pmatrix} \dfrac{X_u}{m} & \dfrac{X_\alpha}{m} & -g\cos\Theta_e & 0 \\ \dfrac{Z_u}{mU} & \dfrac{Z_\alpha}{mU} & -\dfrac{g}{U}\sin\Theta_e & 1 \\ 0 & 0 & 0 & 1 \\ \dfrac{M_u}{J_{yy}} + \dfrac{M_{\dot{\alpha}} Z_u}{J_{yy} mU} & \dfrac{M_\alpha}{J_{yy}} + \dfrac{M_{\dot{\alpha}} Z_\alpha}{J_{yy} mU} & -\dfrac{M_{\dot{\alpha}} g \sin\Theta_e}{J_{yy} U} & \dfrac{M_q + M_{\dot{\alpha}}}{J_{yy}} \end{pmatrix} \begin{Bmatrix} u \\ \alpha \\ \theta \\ q \end{Bmatrix} + \begin{bmatrix} 0 \\ \dfrac{T}{mU} \\ 0 \\ \dfrac{T\xi}{J_{yy}}\left(1 + \dfrac{M_{\dot{\alpha}}}{mU}\right) \end{bmatrix} \mu_1. \tag{5.39}$$

3. Yaw dynamics:

$$\begin{Bmatrix} \dot{\psi} \\ \dot{\beta} \\ \dot{R} \end{Bmatrix} = \begin{pmatrix} 0 & 0 & \sec\Theta_e \\ 0 & \dfrac{Z_\alpha}{mU} & -1 \\ 0 & -\dfrac{M_\alpha}{J_{yy}} & \dfrac{M_q}{J_{yy}} \end{pmatrix} \begin{Bmatrix} \psi \\ \beta \\ R \end{Bmatrix} + \begin{bmatrix} 0 \\ \dfrac{T}{mU} \\ -\dfrac{T\xi}{J_{yy}}\left(1 + \dfrac{M_{\dot{\alpha}}}{mU}\right) \end{bmatrix} \mu_2. \tag{5.40}$$

The success of the rocket guidance and control system thus depends upon the ability of each control subsystem (roll, pitch, yaw) to regulate the roll and yaw rates, while tracking a reference pitch rate. The decoupling of roll, pitch, and yaw dynamics indicates that one can design separate roll, pitch, and yaw control loops for a rocket. It is clear from (5.39) and (5.40) that the short-period pitch (i.e., excluding airspeed variation) and yaw dynamics are nearly identical. Therefore, one can design similar feedback loops for pitch and yaw. Of the three control loops, the roll controller must have the highest bandwidth (speed) for quickly reducing the roll rate, while the pitch and yaw controllers should maintain the vehicle's attitude along the nominal trajectory.

Example 5.1. Consider the *Vanguard* rocket with the following stability and control parameters [2] at the condition of maximum dynamic pressure, $\bar{q} = 28305\ \mathrm{N/m^2}$, with airspeed, $U_\mathrm{e} = 392\ \mathrm{m/s}$, Mach number 1.4, and mass, $m = 6513.2\ \mathrm{kg}$:

$$\begin{aligned} b &= 1.1433\ \mathrm{m} \\ S &= 1.0262\ \mathrm{m^2} \\ \xi &= 8.2317\ \mathrm{m} \\ C_{l_p} &= C_{m_{\dot{\alpha}}} = C_{z_u} = C_{m_u} = 0 \\ \Theta_\mathrm{e} &= 68.5^\circ \\ C_{z_\alpha} &= -3.13\ /\mathrm{rad} \\ C_{m_\alpha} &= 11.27\ /\mathrm{rad} \\ T &= 133202.86\ \mathrm{N} \\ J_{xx} &= 5215.1\ \mathrm{kg\,m^2} \\ J_{yy} &= 156452.8\ \mathrm{kg\,m^2} \\ \frac{b}{2U_\mathrm{e}} C_{m_q} &= -0.321\ \mathrm{s}. \end{aligned} \tag{5.41}$$

Here $b/2$ is the characteristic length and S the reference area used for nondimensionalizing the stability derivatives, e.g.,

$$C_{m_\alpha} = \frac{M_\alpha}{\bar{q} S (b/2)}; \qquad C_{m_q} = \frac{U_\mathrm{e} M_q}{\bar{q} S (b/2)^2}.$$

The dimensional stability derivatives are thus obtained to be the following:

$$\begin{aligned} Z_\alpha &= Y_\beta = -90916\ \mathrm{N/rad} \\ M_\alpha &= -N_\beta = 187132.5\ \mathrm{N\,m/rad} \\ M_q &= N_r = -5330.039\ \mathrm{s}. \end{aligned} \tag{5.42}$$

The rocket has pitch and yaw thrust deflection capability in powered flight such that μ_1, μ_2 can be simultaneously changed by ± 5 deg [12]. Since the vehicle is not equipped with fins and aerodynamic control surfaces, it is crucial to maintain a non-rolling attitude, $P = 0$, in the presence of disturbances, which is achieved by dedicated roll reaction jets. The rolling moment generated by the roll reaction jets is $\tau_x = \pm 625.36$ N m, whose sign can be switched as required. The roll-control system is simply a *bang-bang* switch (Chap. 1) that applies the control torque in the desired direction as soon as roll angle error crosses $\pm 3°$ [12]. Neglecting the airspeed variations (which are typically much slower in time scale, and smaller in magnitude) caused by attitudinal motion, we have the following roll, pitch and yaw dynamics:

1. Roll dynamics:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{p} \end{Bmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.1199 \end{pmatrix} \text{sgn}(u_1). \tag{5.43}$$

2. Pitch dynamics:

$$\begin{Bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{Bmatrix} = \begin{pmatrix} -0.0356 & -0.0233 & 1 \\ 0 & 0 & 1 \\ 1.1961 & 0 & -0.0341 \end{pmatrix} \begin{Bmatrix} \alpha \\ \theta \\ q \end{Bmatrix} + \begin{pmatrix} 0.052171 \\ 0 \\ 7.0084 \end{pmatrix} \mu_1. \tag{5.44}$$

3. Yaw dynamics:

$$\begin{Bmatrix} \dot{\psi} \\ \dot{\beta} \\ \dot{R} \end{Bmatrix} = \begin{pmatrix} 0 & 0 & 2.7285 \\ 0 & -0.0356 & -1 \\ 0 & -1.1961 & -0.0341 \end{pmatrix} \begin{Bmatrix} \psi \\ \beta \\ R \end{Bmatrix} + \begin{pmatrix} 0 \\ 0.052171 \\ -7.0084 \end{pmatrix} \mu_2. \tag{5.45}$$

Here, u_1 is the electrical signal input to the roll jet actuator, whose sign determines the direction of jet firing. The roll dynamics plant has a double pole at origin, $s_{1,2} = 0$, indicating an unstable open-loop plant. The eigenvalues of pitch and yaw dynamics are given by:

```
>> A=[-0.0356 -0.0233  1.0000; 0  0  1; 1.1961  0  -0.0341]; %Pitch dynamics
>> damp(A)

 Eigenvalue      Damping     Freq. (rad/s)
 -1.14e+000     1.00e+000      1.14e+000
  1.05e+000    -1.00e+000      1.05e+000
  2.34e-002    -1.00e+000      2.34e-002

>> Ay=[0  0  2.7285; 0  -0.0356  -1.0000;0  -1.1961  -0.0341]; %Yaw dynamics
>> damp(Ay)

 Eigenvalue      Damping     Freq. (rad/s)
  0.00e+000    -1.00e+000      0.00e+000
 -1.13e+000     1.00e+000      1.13e+000
  1.06e+000    -1.00e+000      1.06e+000
```

Hence, pitch and yaw plants are also unstable about the pitch equilibrium and have nearly identical eigenvalues.

5.4 Roll Control

The roll control system of a launch vehicle is quite similar to that of a reaction jet-controlled, single-axis spacecraft, although some aerodynamic damping in roll, $L_p < 0$, may be available due to stabilizing fins. Small roll perturbations can be caused by an asymmetric atmospheric gust acting on the aerodynamic fins, or by internal swirl moments generated by propellant consumption.

Example 5.2. For *Vanguard* rocket of Example 5.1, design a suitable roll control system for maintaining the vehicle in equilibrium roll attitude.

We see in Example 5.1 that the rocket is unstable in roll due to the absence of aerodynamic fins, thereby requiring active feedback stabilization. The roll controller is a switching system that maintains the vehicle within $\pm 3^\circ$ of roll error by applying a constant rolling moment via reaction jets. Such a control system is thus quite simple to implement and was discussed at length in Example 1.8.

Equation (5.43) is expressed as follows:

$$\ddot{\phi} = 0.1199 \operatorname{sgn}(u_1), \tag{5.46}$$

where $u_1(t)$ is the electrical signal input to the roll jet actuator determining the direction of control jets. Being nonlinear in nature, a switching control system can encounter the problem of *chatter*, i.e., rapid actuation even when a small error around the desired equilibrium state is detected, leading to a wastage of control effort. In order to avoid chatter, a dead-zone of $\pm 3^\circ$ is deliberately built in, resulting in the following control law:

$$u_1(t) = \begin{cases} > 0 & (\phi < -3^\circ) \\ 0 & (-3 \le \theta \le 3^\circ) \\ < 0 & (\theta > 3^\circ) \end{cases}$$

For simulating the closed-loop system, we refer the reader to Example 1.8. The closed-loop simulation results for 2 min of flight (assuming no change of roll parameters with time) are plotted in Figs. 5.4 and 5.5 for the initial perturbation of $\phi(0) = -2^\circ$ and $p(0) = 0.01$ rad/s. Note the subtle damping provided by the switching controller in the roll-rate response, which causes a delaying of successive jet pulses, thereby stabilizing the system about the zero roll state.

A linear feedback control law has an advantage over the switching controller in which it can achieve a more precise control with a smaller overall effort. However, it is much more difficult to implement a linear control law for a reaction jet system, because it requires a precise modulation of control torque magnitude by valves. Consider the following transfer function of the roll dynamics plant:

$$G(s) = \frac{\phi(s)}{\tau_x(s)} = \frac{\frac{1}{J_{xx}}}{s^2 - \frac{L_p}{J_{xx}} s}, \tag{5.47}$$

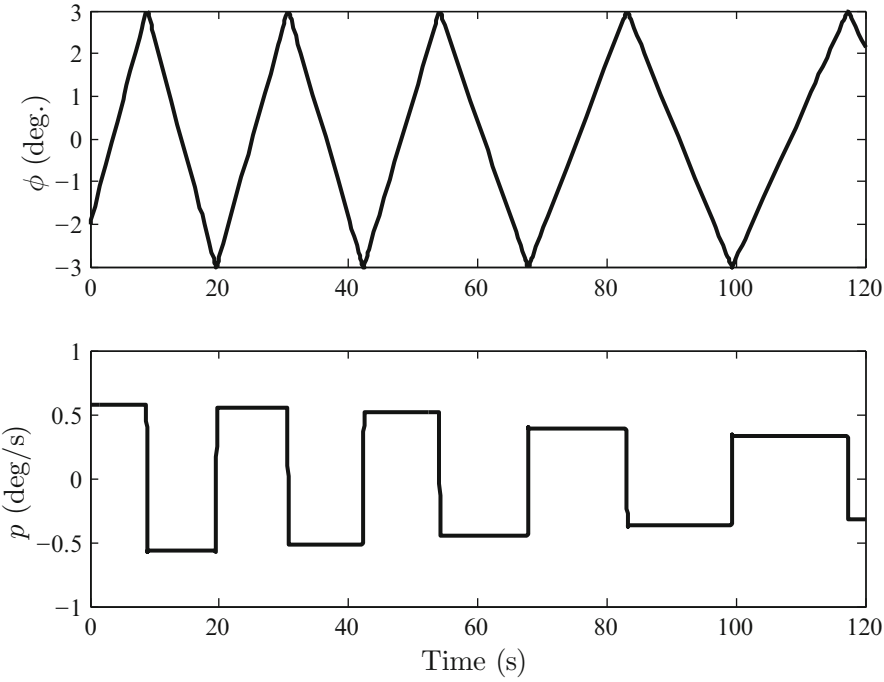


Fig. 5.4 Closed-loop initial response of the switching roll control system for *Vanguard* rocket

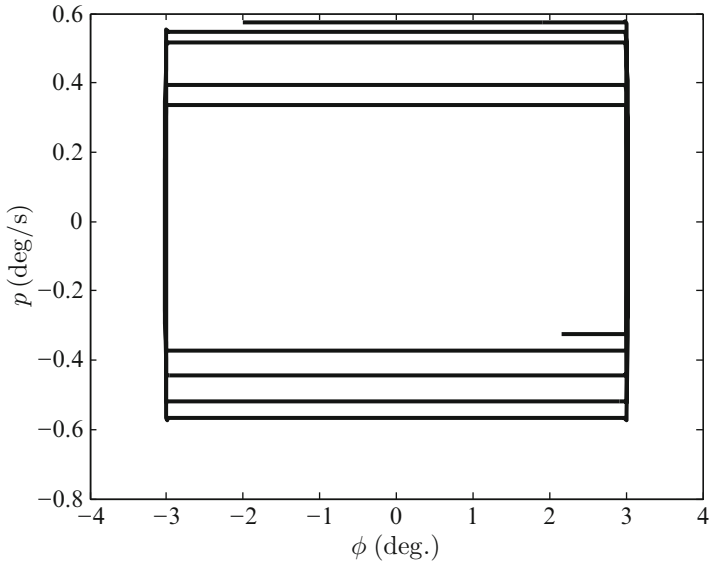


Fig. 5.5 Phase-plane plot, p vs. ϕ, of the switching roll control system for *Vanguard* rocket

where the yaw perturbation, ψ, is assumed zero. Since the plant possesses a pure integration in the open-loop, we can achieve a zero steady-state error to a step roll command, ϕ_d, by proportional feedback control,

$$\tau_x(s) = KE(s) = K[\phi_d s - \phi(s)], \tag{5.48}$$

which makes the closed-loop transfer function

$$\frac{\phi_{(s)}}{\phi_d(s)} = \frac{KG(s)}{1 + KG(s)} = \frac{\frac{K}{J_{xx}}}{s^2 - \frac{L_p}{J_{xx}}s + \frac{K}{J_{xx}}}, \tag{5.49}$$

asymptotically stable. As there can be large variations in the values of J_{xx} and L_p during flight, the feedback gain, K, must be carefully selected such that closed-loop performance does not deteriorate significantly at any point. The closed-loop characteristic equation,

$$s^2 - \frac{L_p}{J_{xx}}s + \frac{K}{J_{xx}} = s^2 + 2\zeta\omega s + \omega^2, \tag{5.50}$$

shows that while the settling-time,

$$t_s \simeq \frac{4}{\zeta\omega} = -\frac{8J_{xx}}{L_p}, \tag{5.51}$$

is unaffected by feedback gain, K, the value of the closed-loop damping ratio, ζ (thus the transient overshoots and control input magnitude), can be selected as desired through K as follows:

$$\zeta = -\frac{L_p}{2\sqrt{KJ_{xx}}}. \tag{5.52}$$

Of course, one can have a better closed-loop control using combined roll and roll rate feedback (i.e., a PD controller),

$$\tau_x(s) = (K_1 + sK_2)E(s), \tag{5.53}$$

with the closed-loop transfer function

$$\frac{\phi_{(s)}}{\phi_d(s)} = \frac{K_1 + sK_2}{J_{xx}s^2 + \left(K_2 - L_p\right)s + K_1}, \tag{5.54}$$

with a handle on both the settling-time,

$$t_s \simeq \frac{4}{\zeta\omega} = \frac{8J_{xx}}{K_2 - L_p}, \tag{5.55}$$

Table 5.1 Roll moment of inertia and damping in roll for Stage-I of a launch vehicle

$t(s)$	J_{xx} (kg m^2)	L_p (N m s/rad)
0	4937.6	−6.8801
6.0606	4856.5	−1469.8
12.121	4775.5	−2277.1
18.182	4694.4	−2338.5
24.242	4613.3	−1923.9
30.303	4532.3	−1358.5
36.364	4451.2	−854.08
42.424	4370.1	−489.62
48.485	4289.1	−260.37
54.545	4208	−130.15
60.606	4126.9	−61.817
66.667	4045.9	−28.148
72.727	3964.8	−12.381
78.788	3883.7	−5.2952
84.848	3802.7	−2.2145
90.909	3721.6	−0.9102
96.97	3640.5	−0.36923
103.03	3559.5	−0.14842

and the damping-ratio,

$$\zeta = \frac{K_2 - L_p}{2\sqrt{K_1 J_{xx}}}. \tag{5.56}$$

However, adding the roll rate feedback increases the input magnitude requirement, while introducing susceptibility (hence reduced robustness) to rate gyro noise. Using a suitable nonlinear switching strategy where a pendulum senses roll rate magnitude above a given threshold, one can improve closed-loop performance without requiring rate feedback. The design can be made more robust with respect to angular measurement noise by adding a suitable lag compensator as a pre-filter.

Example 5.3. Using the nominal data given in Table 5.1 for the first stage of a launch vehicle, design a roll control system to achieve a change in the roll angle by 1 rad before the first stage separation at $t = 103.03$ s. The vehicle is equipped with a roll control gas jet actuator with first-order time constant of 0.2 s, which can exert a maximum torque of ± 150 N m with a deadband of ± 5 N m (i.e., actuator does not respond to commanded torque less than 5 N m in magnitude). The closed-loop system must have adequate robustness with respect to roll angle measurement noise.

Before beginning the design, we analyze the plant's real pole, $s = L_p/J_{xx}$, plotted with time using spline interpolation in Fig. 5.6. Note the large variation in the pole's location during the first 60 s. If one takes an average value, −0.25, the following approximate plant settling-time can be expected

$$t_s \simeq -\frac{8}{0.25} = 32 \text{ s}.$$

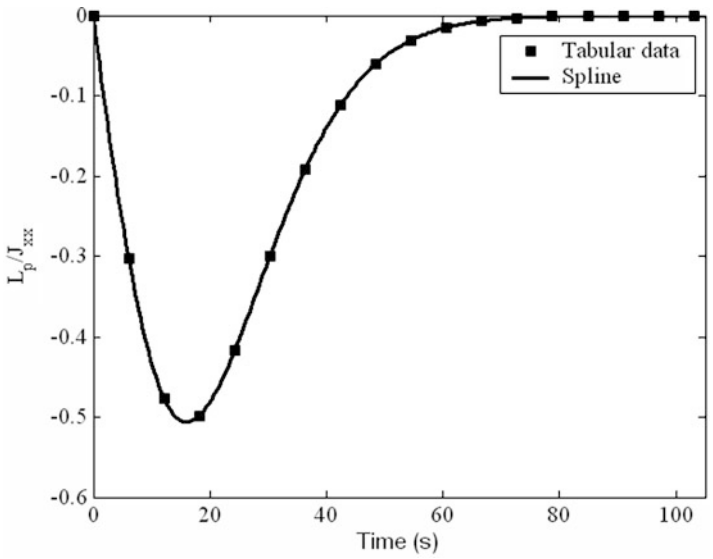


Fig. 5.6 Pole, $s = L_p/J_{xx}$, of roll dynamics plant for a gravity-turn launch vehicle

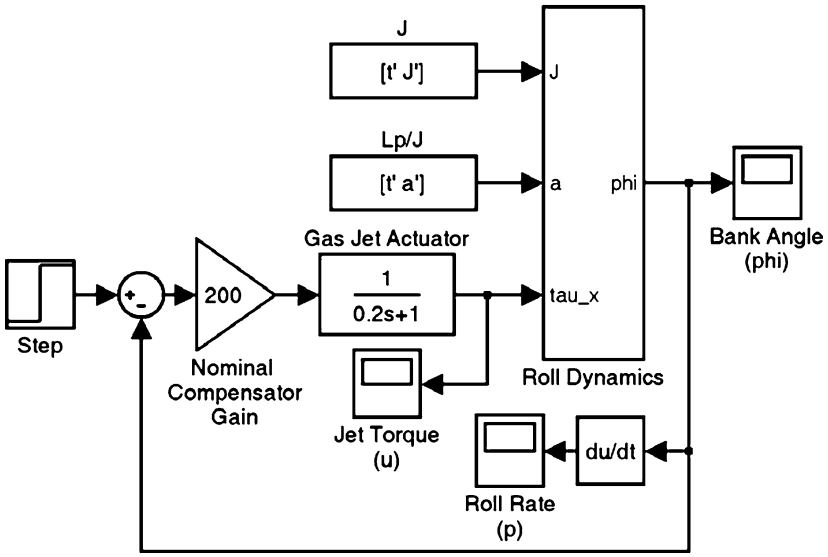


Fig. 5.7 Simulink block diagram of roll control system with proportional angle feedback for a launch vehicle

We will first examine proportional feedback with a constant gain, K, for which the Simulink block diagram is shown in Fig. 5.7. The roll dynamics plant used in the model is depicted in Fig. 5.8. The tabulated values of J_{xx} and $a = L_p/J_{xx}$ are

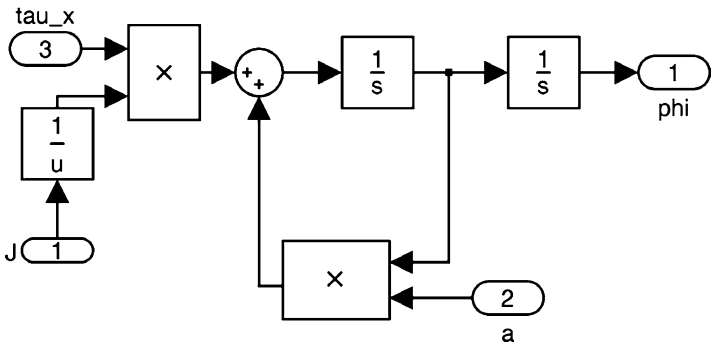


Fig. 5.8 Simulink block diagram of roll dynamics plant for a launch vehicle

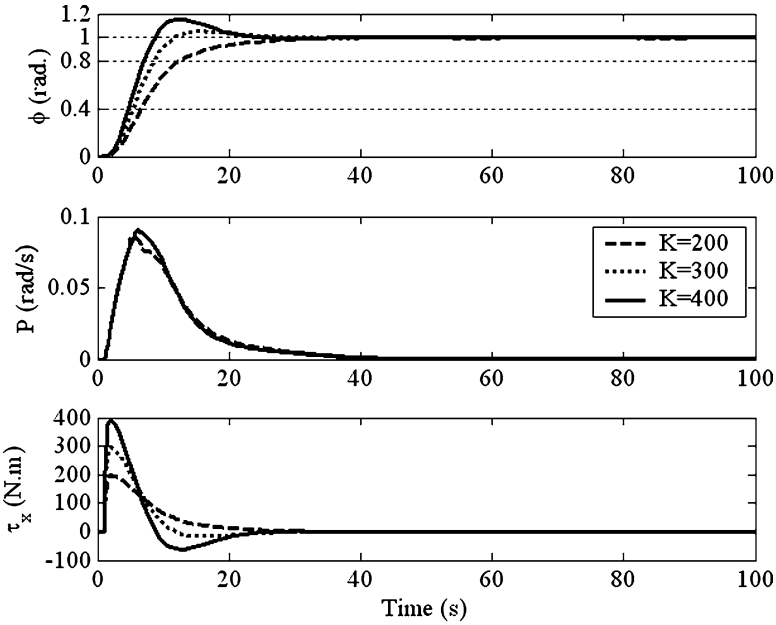


Fig. 5.9 Closed-loop response of roll control system with proportional feedback for a launch vehicle

stored in the MATLAB workspace as row vectors at the given time instants, t, and made available to the Simulink model as shown. Taking three different values of the gain, $K = 200, 300, 400$, the simulated closed-loop response without actuator nonlinearity and sensor noise is plotted in Fig. 5.9. Note that the maximum torque limit is exceeded in all cases, while $K = 200$ requires the minimum peak torque and also has the smallest overshoot, but an increased settling time. A further reduction in K tends to increase the error at $t = 100$ s, which is unacceptable. Therefore, we select $K = 200$ as our nominal design.

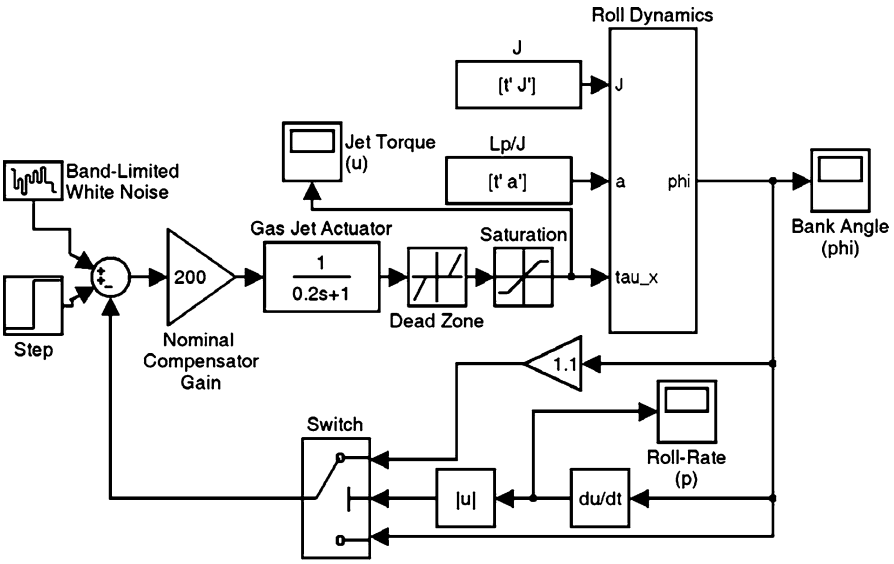


Fig. 5.10 Simulink block diagram of roll control system with proportional angle feedback and a roll rate-activated switch for a launch vehicle

An improvement over the nominal design is obtained by taking the roll rate magnitude into consideration. While not actually feeding back P, we use a pendulum-based switch (Fig. 5.10) that is activated when the roll rate magnitude exceeds 0.05 rad/s, in which case the feedback gain is increased by 10% for quickly reducing the transient error. Otherwise, the nominal gain ($K = 200$) is used. A comparison of the closed-loop response of the switched control system with actuator nonlinearities and that of the nominal linear design (without sensor noise) is plotted in Fig. 5.11. Note the slight increase in the settling time of the switched case with saturated control torque and deadband, as well as the reduction in the maximum pitch rate. Consequently, the transient angle response is much smoother than the nominal design.

Addition of sensor noise within the system's bandwidth can substantially degrade the closed-loop performance, especially in a time-varying plant controlled with a fixed feedback gain (as in the present case). In order to make the system robust with respect to noise in roll angle measurement, we add a lag compensator (Chap. 3) with the transfer function

$$H(s) = \frac{0.2s + 1}{s + 1}$$

as shown in the Simulink block diagram of Fig. 5.12. The lag compensator essentially moves the actuator pole from $s = -5$ to $s = -1$ with a unit DC gain, thereby decreasing control activity in response to the higher frequency noise input. In other words, the compensator decreases the control bandwidth by introducing phase lag and gain roll-off. The Bode plot of the lag compensator is given in Fig. 5.13 showing

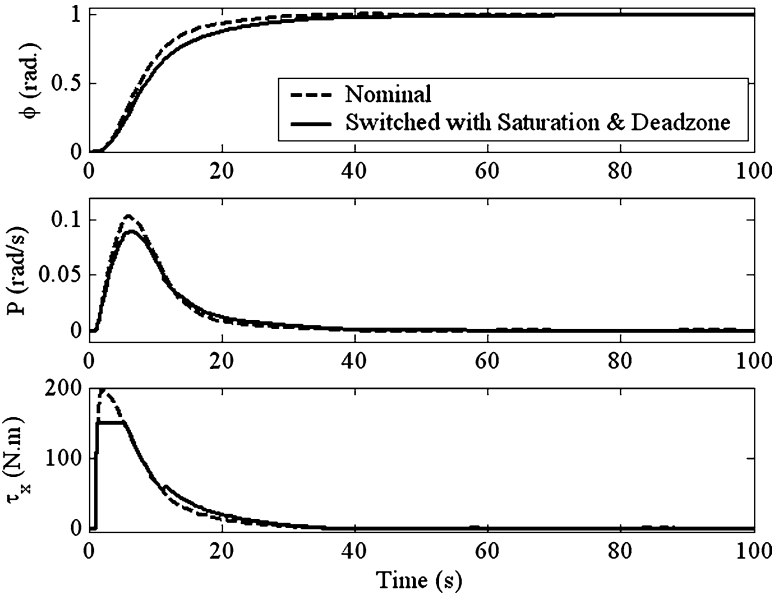


Fig. 5.11 Comparison of switched and unswitched roll control systems with proportional angle feedback for a launch vehicle

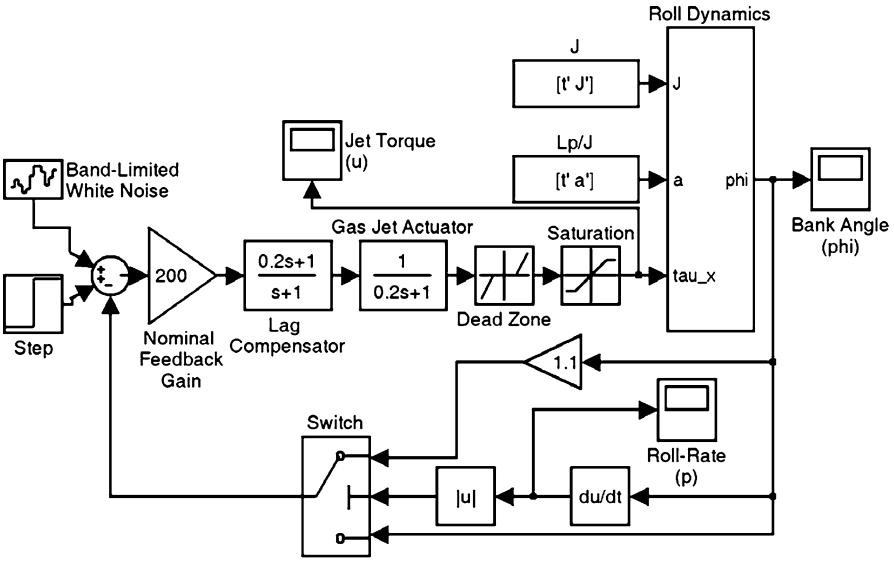


Fig. 5.12 Simulink block diagram of roll control system for a launch vehicle with proportional angle feedback, lag compensator, actuator nonlinearities, a roll rate-activated switch, and sensor noise

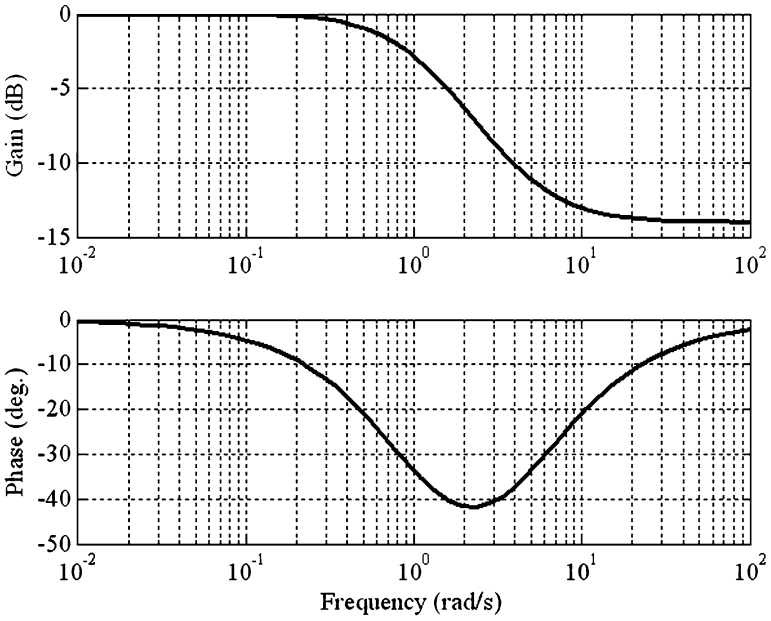


Fig. 5.13 Frequency response of the lag compensator for roll control system of a launch vehicle

a gain reduction by 14 dB in the frequency range $0.2 \leq \omega \leq 10$ rad/s and a phase dip of $-42°$ near $\omega = 2$ rad/s. The noise intensity for the simulation is specified by setting noise power spectral density as 10^{-4} rad in the *Band-Limited White Noise* block (Fig. 5.12), which corresponds to a rough angular resolution of 0.573°. A good IMU with optical angle encoder can produce a resolution about a fifth of this value. However, simulation for such a crude measurement would reveal the robustness (or lack thereof) of the closed-loop system. The noise sample interval is also conservatively selected to be 0.1 s, which is only half the actuator time constant. The response of the closed-loop system with (Fig. 5.12) and without (Fig. 5.10) the lag compensator is compared in Figs. 5.14 and 5.15. Note the significant improvement in the smoothness of the closed-loop torque input as well as a slight reduction in the peak roll rate caused by the lag compensator, albeit with an increase in settling time by 2–3 s.

5.5 Pitch-Yaw Control

A large rocket is designed to operate in a vertical plane in such a way that the normal acceleration required for maneuvering is naturally provided by the gravity component. This kind of trajectory is called a *gravity-turn*, and is used both for launching a payload to an orbit and for ballistic reentry missions. In order to fly a gravity-turn trajectory, the angle-of-attack and angle of sideslip must be kept zero,

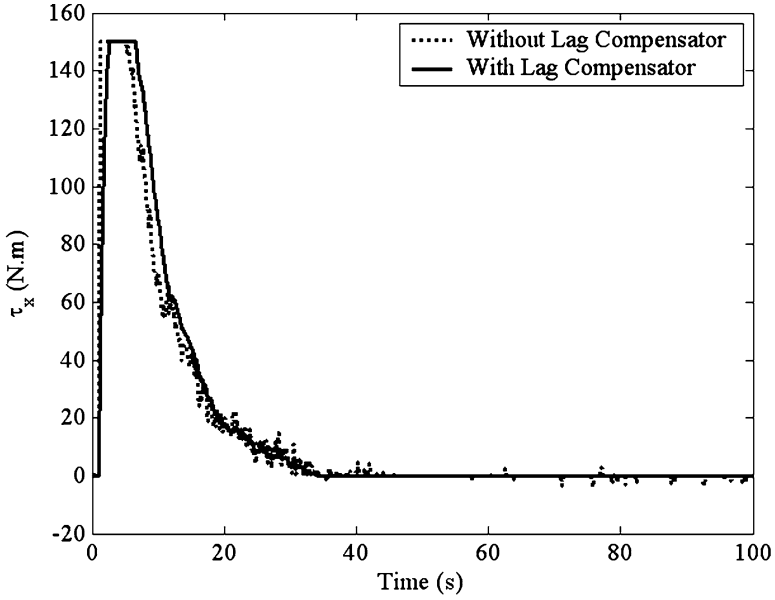


Fig. 5.14 Comparison of closed-loop rolling moment input of roll control systems with and without lag compensator

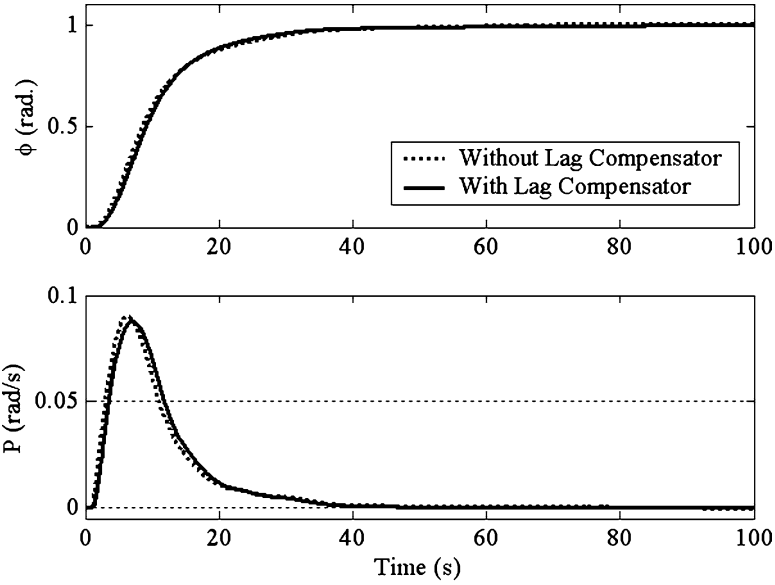


Fig. 5.15 Comparison of closed-loop roll angle and rate response of roll control systems with and without lag compensator

which requires matching the pitch angle, Θ, with the flight path angle (Chap. 2). The task of the control system is to align the longitudinal plane (or pitch plane) of the vehicle with the vertical plane of flight by nullifying roll and yaw motions, while generating the required pitch rate through gimbaling. The nominal trajectory for a given mission is either pre-computed and stored onboard, or supplied to the vehicle via a radio link from the ground station. We will study the design of the pitch control system, and note that yaw control is quite similar (if not identical) to the pitch regulator.

We have the following equations of zero-lift translation (Chap. 2) in a vertical plane with radius,[3] $r(t)$, latitude, $\delta(t)$, airspeed, $v(t)$, and flight path angle, $\gamma(t)$, relative to a planet-fixed frame with origin at the center of the spherical planet:

$$\begin{aligned}
\dot{r} &= v \sin\gamma \\
\dot{\delta} &= \frac{v}{r}\cos\gamma \\
\frac{(T-D)}{m} - g\sin\gamma &= \dot{v} + \Omega^2 r\cos\delta(\cos\gamma\cos\psi\sin\delta - \sin\gamma\cos\delta) \\
g\cos\gamma &= -v\dot{\gamma} + \frac{v^2}{r}\cos\gamma + 2\Omega v\sin\gamma\cos\delta \\
&\quad + \Omega^2 r\cos\delta(\sin\gamma\cos\psi\sin\delta + \cos\gamma\cos\delta), \qquad (5.57)
\end{aligned}$$

where ψ is the constant flight direction, T the thrust, and D the atmospheric drag. The equations include the Coriolis and centripetal acceleration due to planetary rotation, Ω. The nominal trajectory,

$$[r_d(t), \delta_d(t), v_d(t), \gamma_d(t)],$$

is an optimal solution of these equations for specific boundary conditions.

Consider the stability axis, $(\mathbf{i_d}, \mathbf{j_d}, \mathbf{k_d})$, with $\mathbf{i_d}$ always along the nominal flight direction and $\mathbf{j_d}$ normal to the plane of flight. The axis, $\mathbf{k_d}$, completes the right-handed triad, $\mathbf{k_d} = \mathbf{i_d} \times \mathbf{j_d}$. Such a frame describes the nominal attitude of the vehicle required for tracking the nominal trajectory. With a nominal pitch rate, $Q_d(t)$, the angular velocity of the stability axes is given by

$$\boldsymbol{\omega}_{\mathrm{d}} = Q_d \mathbf{j_n}. \qquad (5.58)$$

The nominal pitch rate is continuously updated by the guidance system such that it always equals the rate of change of flight path angle (5.57):

$$\begin{aligned}
Q_d = \dot{\gamma}_d &= \left(\frac{v_d}{r_d} - \frac{g_0 r_0^2}{v_d r_d^2}\right)\cos\gamma_d + 2\Omega v_d \sin\gamma_d \cos\delta_d \\
&\quad + \Omega^2 r_d \cos\delta_d(\sin\gamma_d\cos\psi_d\sin\delta_d + \cos\gamma_d\cos\delta_d). \qquad (5.59)
\end{aligned}$$

[3] In order to avoid confusion, we have denoted yaw-rate perturbation as $R(t)$.

Tracking a reference pitch rate command is called a *pitch program* and is used in one way or the other by all launch vehicles. By integrating the pitch rate signal, the desired pitch angle, Θ_d, can also be obtained at each time instant. However, instead of maintaining a pitch equilibrium, the pitch program is intended for varying the pitch rate in a specific fashion, which requires generating a nominal pitching moment through gimbaling. The nominal gimbal profile can be calculated by putting $\alpha = 0$ and $\dot{\alpha} = 0$ in the pitching moment equation:

$$\mu_{1_d} = \frac{J_{yy}\dot{Q}_d - M_q Q_d}{T\xi}. \tag{5.60}$$

By neglecting the instantaneous variation of $Q_d(t)$ due to planetary rotation, radius, and airspeed, we write

$$\dot{Q}_d \simeq \frac{\partial Q_d}{\partial \gamma_d}\dot{\gamma}_d = -Q_d\left(\frac{v_d}{r_d} - \frac{g_0 r_0^2}{v_d r_d^2}\right)\sin\gamma_d = -Q_d^2\tan\gamma_d, \tag{5.61}$$

we have the following gimbaling command for the pitch program:

$$\mu_{1_d} = -\frac{(J_{yy}Q_d\tan\gamma_d + M_q)Q_d}{T\xi}. \tag{5.62}$$

This is quite easy to implement in a guidance computer, where $Q_d(t)$ profile can be stored and the current flight path angle fed back by a rate-integrating gyro. A similar pitch program was implemented in the earliest guided rocket vehicles (such as the German *V-2* rocket) by a clockwork mechanism driving the pitch gimbal via rate-integrating gyro signal. With the availability of digital computers, it is now routine to perform online computations of the desired pitch attitude.

We can now contemplate the guidance and control system for pitch-yaw dynamics, shown as a schematic block diagram in Fig. 5.16. The feedforward part of the system is the pitch program consisting of (5.59) and (5.60), which are supplied by the guidance computer. Additionally, the guidance system can issue gimbal commands, $\mu_{1d}(t), \mu_{2d}(t)$, to correct small deviations from the nominal trajectory. The task of the pitch-yaw controller is to drive the angular rate error, q, R, to zero as quickly as possible by corrective gimbaling. Such a controller uses pitch and yaw angles or rates, as well as normal acceleration, $\mathbf{a_n}(t)$, for feedback. The outer guidance loop senses flight path errors, $\gamma(t), \psi(t)$, via rate-integrating gyros to correct flight path deviations.

Example 5.4. Consider the pitch control of the first stage of *Vanguard* rocket (Example 5.1) for flying a gravity-turn trajectory specified in Table 5.2. The duration of flight is small enough for the planetary rotation effects to be negligible. Thus, we have

$$Q_d = \dot{\gamma}_d = \left(\frac{v_d}{r_d} - \frac{g_0 r_0^2}{v_d r_d^2}\right)\cos\gamma_d, \tag{5.63}$$

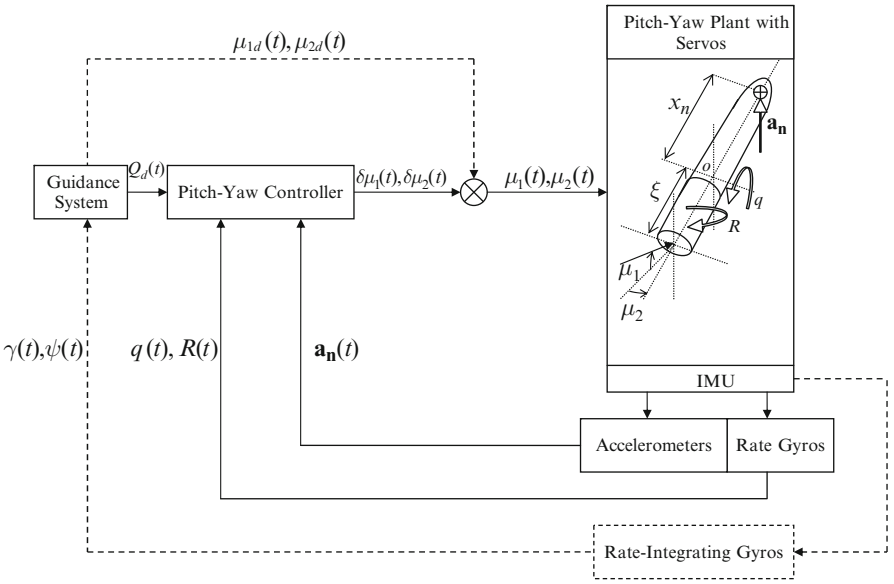


Fig. 5.16 A typical pitch-yaw guidance and control system for a gravity-turn launch vehicle

Table 5.2 Nominal gravity-turn trajectory for Stage I flight of the *Vanguard* rocket

$t\,(s)$	r (km)	v (km/s)	γ (rad)
11.347	6378.6	0.088604	1.5349
14.047	6378.9	0.10625	1.5237
16.747	6379.2	0.12437	1.5115
19.447	6379.5	0.14287	1.4985
22.147	6379.9	0.16166	1.4848
24.847	6380.4	0.18074	1.4704
27.547	6380.9	0.20012	1.4555
30.247	6381.5	0.21983	1.4401
32.947	6382.1	0.24004	1.4242
35.647	6382.8	0.2608	1.408
38.347	6383.5	0.28225	1.3915
41.047	6384.3	0.30443	1.3747
43.747	6385.1	0.32742	1.3577
46.447	6386	0.35114	1.3407
49.147	6386.9	0.37494	1.3235
51.847	6388	0.39865	1.3063
54.547	6389	0.42268	1.289
57.247	6390.1	0.44717	1.2717
59.947	6391.3	0.47264	1.2544
62.647	6392.6	0.49934	1.2372
65.347	6393.9	0.52757	1.22

(continued)

Table 5.2 (continued)

$t\,(s)$	r (km)	v (km/s)	γ (rad)
68.047	6395.2	0.55753	1.2031
70.747	6396.7	0.58946	1.1863
73.447	6398.2	0.62356	1.1698
76.147	6399.8	0.65985	1.1536
78.847	6401.5	0.6983	1.1378
81.547	6403.2	0.73892	1.1223
84.247	6405.1	0.7817	1.1072
86.947	6407	0.82663	1.0926
89.647	6409	0.87368	1.0784
92.347	6411.1	0.92285	1.0646
95.047	6413.4	0.97414	1.0512
97.747	6415.7	1.0276	1.0383
100.45	6418.2	1.0832	1.0258
103.15	6420.7	1.1411	1.0138
105.85	6423.4	1.2014	1.0021
108.55	6426.2	1.2641	0.99093
111.25	6429.1	1.3295	0.98013
113.95	6432.1	1.3975	0.96973
116.65	6435.3	1.4685	0.95973
119.35	6438.6	1.5425	0.95013
122.05	6442.1	1.6196	0.94091
124.75	6445.7	1.7002	0.93206
127.45	6449.5	1.7843	0.92359
130.15	6453.4	1.8722	0.91548
132.85	6457.5	1.964	0.90772
135.55	6461.8	2.0601	0.90031
138.25	6466.2	2.1607	0.89324
140.95	6470.9	2.2662	0.8865
143.65	6475.7	2.377	0.88009
146.35	6480.8	2.4937	0.874

where $r_0 = 6378.14$ km and $g_0 = 9.806\ \text{m/s}^2$. The thrust is maintained constant at $T = 133202.86$ N, while the variations of the mass and the pitch moment of inertia are the following:

$$m = 7727.3[1 - 0.0054(t - 11.347)]\ \text{kg}$$
$$J_{yy} = 215000[1 - 0.006(t - 11.347)]\ \text{kg m}^2.$$

The 1976 US standard atmosphere model [21] is used to generate the dynamic pressure, $\bar{q}$, and Mach number, $\mathcal{M}$, along the nominal trajectory for the first 200 s of flight, which are plotted in Fig. 5.17. The variations of the drag, D, and the stability derivatives, Z_α, M_α, M_q, along the nominal trajectory are plotted in Fig. 5.18. Note the sharp transonic drag rise as well as the large changes in the stability derivatives near $\mathcal{M} = 1$ ($t = 50$ s). Also note the maximum dynamic pressure flight

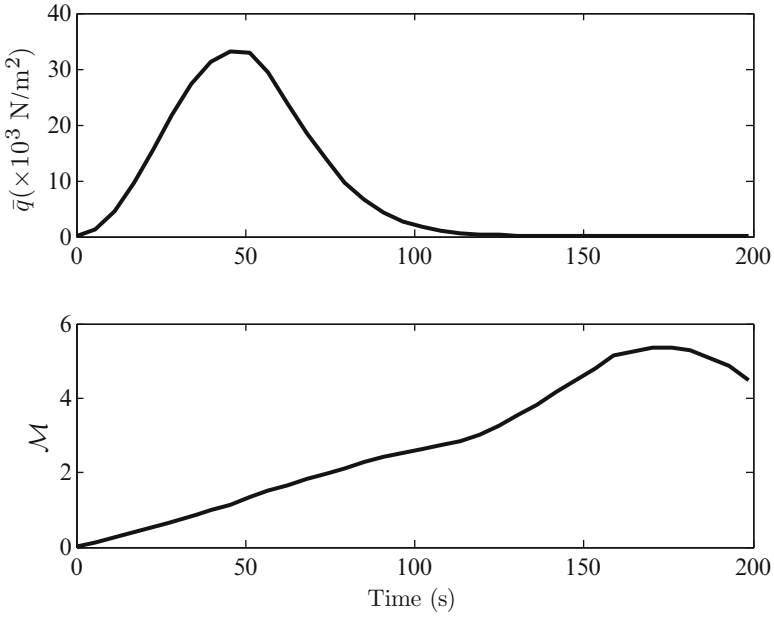


Fig. 5.17 The dynamic pressure and Mach number profiles for the nominal flight of the *Vanguard* rocket

point near $t = 50$ s. The portion of flight lying between the transonic and the maximum dynamic pressure corresponds to the region of the largest aerodynamic loads experienced by the vehicle. There is another smaller drag peak at the onset of hypersonic regime ($t = 125$ s) corresponding to the point of maximum aerodynamic heating.

A Simulink block diagram coded for simulating the ballistic, gravity-turn flight path in the vertical plane is depicted in Fig. 5.19. Note the absence of planetary rotation (thus the latitude) in this basic trajectory model. Here, "u1" and "u2" denote the nominal axial and normal acceleration inputs applied by the rocket thrust and the atmosphere, while "rout," "vout," and "gamout" are the radius, airspeed, and flight path angle, respectively. The initial conditions are specified in the respective integrator blocks at $t = 11.347$ s as per Table 5.2.

With $M_{\dot{\alpha}} \simeq 0$, the pitch dynamics plant about the nominal trajectory is given by

$$\begin{Bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{Bmatrix} = \begin{pmatrix} \dfrac{Z_\alpha}{mv_d} & -\dfrac{g}{v_d}\sin\Theta_d & 1 \\ 0 & 0 & 1 \\ \dfrac{M_\alpha}{J_{yy}} & 0 & \dfrac{M_q}{J_{yy}} \end{pmatrix} \begin{Bmatrix} \alpha \\ \theta \\ q \end{Bmatrix} + \begin{pmatrix} \dfrac{T}{mv_d} \\ 0 \\ \dfrac{T\xi}{J_{yy}} \end{pmatrix} \delta\mu_1, \tag{5.64}$$

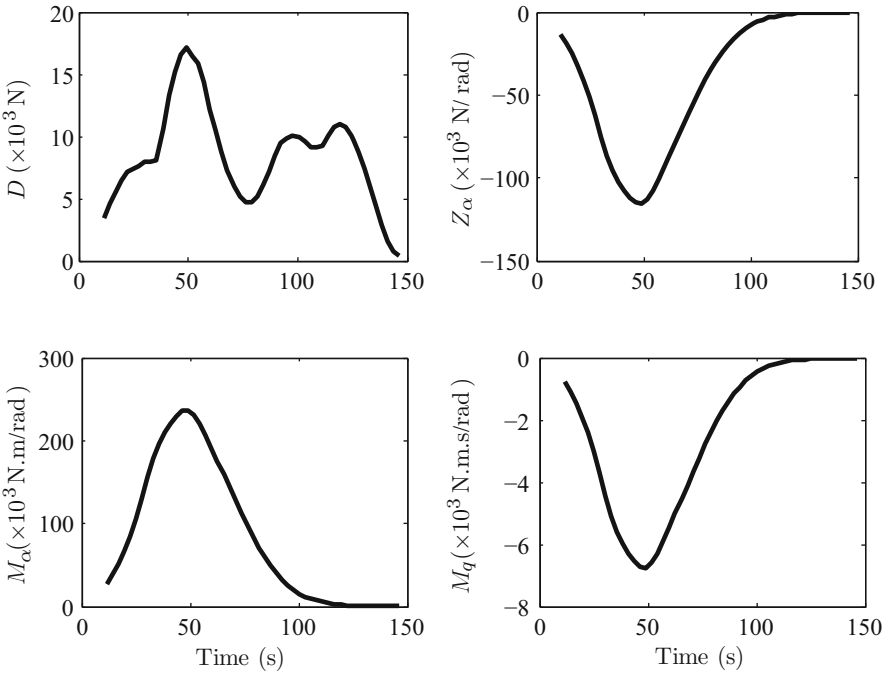


Fig. 5.18 The atmospheric drag and stability derivatives along the nominal trajectory of the *Vanguard* rocket

where $\delta\mu_1$ is the corrective gimbal input for attitude control. The vehicle is equipped with first-order pitch and yaw gimbal servos, each of time constant 0.02 s. Let us design a pitch controller for the rocket such that the angle-of-attack and pitch deviations are brought to within $\pm 1^\circ$ with the gimbal angle not exceeding $\pm 5^\circ$.

We select pitch angle error as the sole output, $y(t) = \theta(t)$, which is measured via a rate-integrating gyro. A regulator and a full-order observer are then designed using the plant dynamics in the maximum dynamic pressure region, $t \simeq 50$ s, by the frozen LQR method (Chap. 3). The plant parameters at the selected design point have already been derived in Example 5.1. The initial response of the closed-loop system (with servo) to a 5.7° initial angle-of-attack perturbation is plotted in Fig. 5.20. The closed-loop frequency response at the design point is plotted in Fig. 5.21, showing a gain margin of 37.9 dB and a phase margin of 175°. While the gain margin may be adequate for the model parameter variation (process noise) envisaged here, it can be improved by adding a pitch-rate feedback loop. A Simulink model (Fig. 5.22) is created for the pitch dynamics plant such that its coefficients, $m, U, J_{yy}, Z_\alpha, M_\alpha$, and M_q, are continuously updated along the nominal trajectory according to Table 5.2, (5.64) and Fig. 5.18. Finally, the pitch control system is simulated along with gimbal saturation limits using the Simulink block diagram of

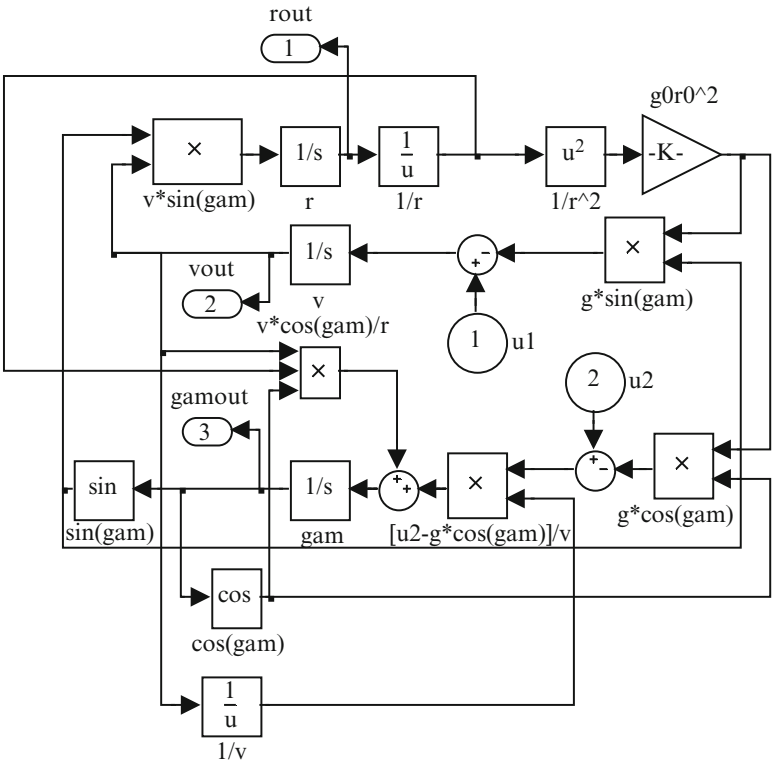


Fig. 5.19 Simulink block diagram for simulating the nominal trajectory of the *Vanguard* rocket

Fig. 5.23, and the results are plotted in Figs. 5.24 and 5.25. The design steps are given by the following MATLAB-CST statements:

```
>> A =[-0.0356   -0.0233    1.0000
                0          0    1.0000
           1.1961          0   -0.0341];

>> B=[0.0522   0    7.0084]';
>> C=[0  1  0]; D=0;
>> [k,S,E]=lqr(A,B,[0.01 0 0;0 0.1 0;0 0 0.1],1) %Regulator design by LQR

k =
    0.1954    0.3456    0.4993

S =
    0.4326   -0.4183    0.0247
   -0.4183    0.5933    0.0524
    0.0247    0.0524    0.0711

E =
   -2.4910
   -0.0397
   -1.0488
```

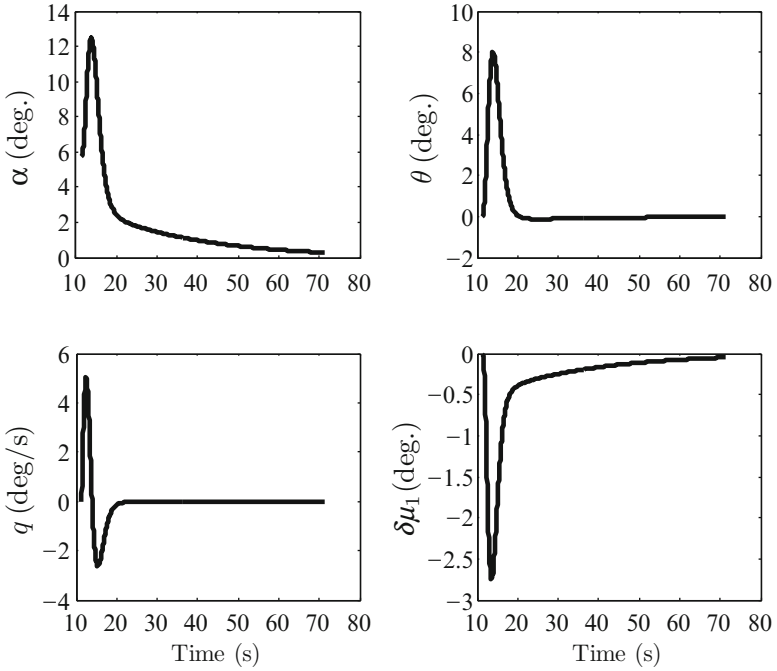


Fig. 5.20 Closed-loop initial response (with servo) of the pitch control system at the design point for Stage I of the *Vanguard* rocket

```
>> Lp=lqr(A',C',eye(3),1);L=Lp' %Observer (Kalman Filter) design by LQR

L =
    4.5545
    3.3159
    4.9975

% Augmented plant with gimbal servo follows:
>> Abar=[A B;zeros(1,3) -50]

Abar =
   -0.0356   -0.0233    1.0000    0.0522
         0         0    1.0000         0
    1.1961         0   -0.0341    7.0084
         0         0         0  -50.0000

>> Bbar=[0 0 0 50]'

Bbar =
     0
     0
     0
    50

>> Cbar=[C 0]

Cbar =
     0     1     0     0
```

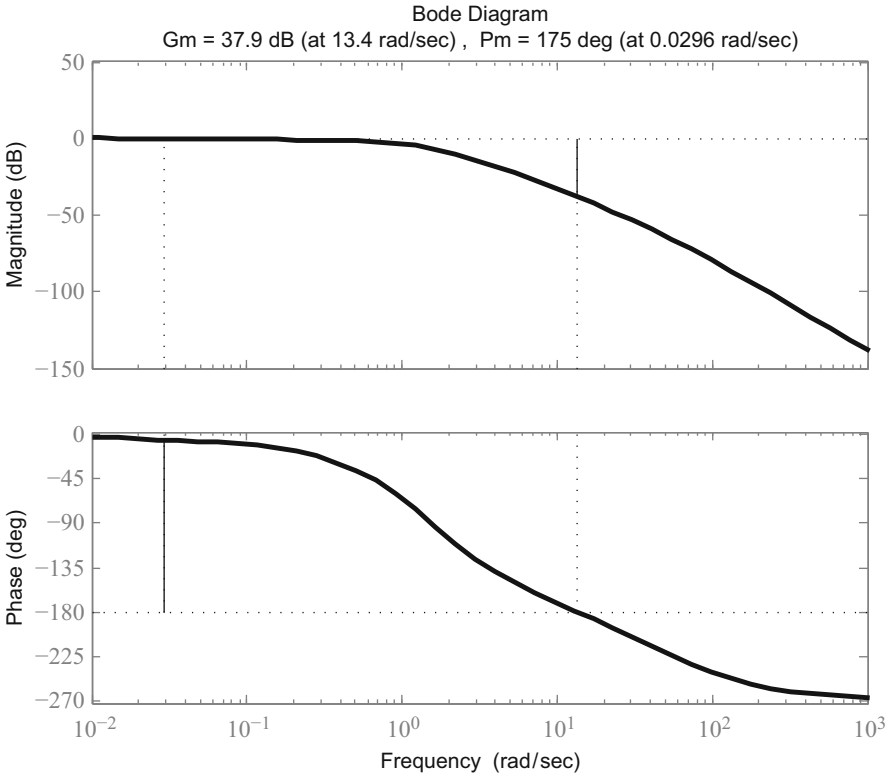


Fig. 5.21 Closed-loop frequency response and gain and phase margins of the pitch control system at the design point for Stage I of the *Vanguard* rocket

```
>> Ac=[Abar  -Bbar*k;L*Cbar A-L*C-B*k] %Servo augmented closed-loop system

Ac =
   -0.0356   -0.0233    1.0000    0.0522         0         0         0
         0         0    1.0000         0         0         0         0
    1.1961         0   -0.0341    7.0084         0         0         0
         0         0         0  -50.0000   -9.7704  -17.2824  -24.9673
         0    4.5545         0         0   -0.0458   -4.5958    0.9739
         0    3.3159         0         0         0   -3.3159    1.0000
         0    4.9975         0         0   -0.1734   -7.4199   -3.5337

>> Bc=[Bbar*k; B*k]
Bc =
         0         0         0
         0         0         0
         0         0         0
    9.7704   17.2824   24.9673
    0.0102    0.0180    0.0261
         0         0         0
    1.3695    2.4224    3.4996

>> sys=ss(Ac,Bc(:,2),[0 1 0 zeros(1,4)],0) %Closed-loop system with pitch
   angle output
```

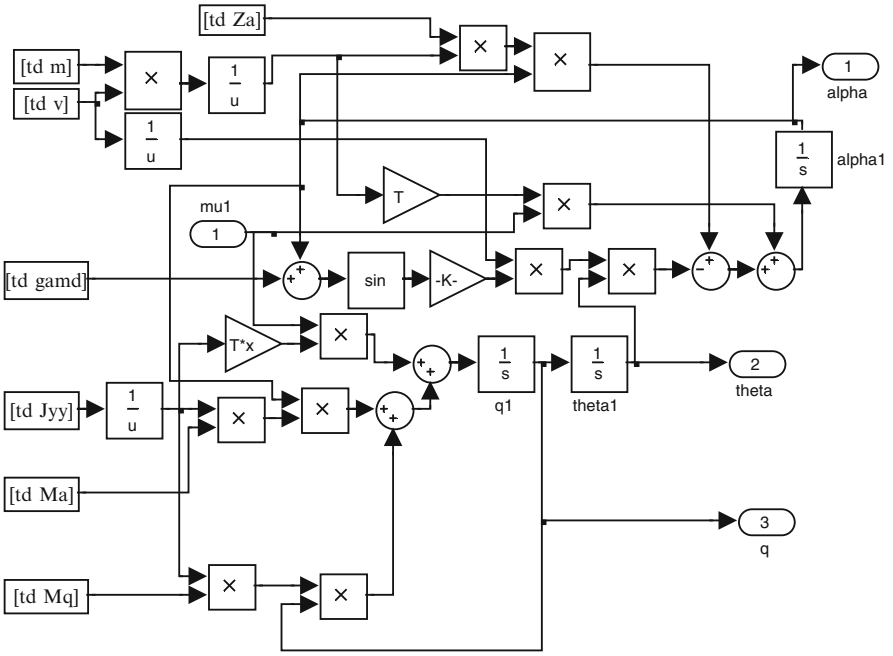


Fig. 5.22 Simulink block diagram for simulating the pitch dynamics plant for Stage I of the *Vanguard* rocket along the nominal trajectory

```
a =
           x1        x2        x3        x4        x5        x6        x7
   x1 -0.03561  -0.02327         1   0.05217         0         0         0
   x2         0         0         1         0         0         0         0
   x3     1.196         0  -0.03407     7.008         0         0         0
   x4         0         0         0       -50     -9.77    -17.28    -24.97
   x5         0     4.554         0         0   -0.0458    -4.596    0.9739
   x6         0     3.316         0         0         0    -3.316         1
   x7         0     4.997         0         0   -0.1734     -7.42    -3.534

b =
           u1
   x1       0
   x2       0
   x3       0
   x4   17.28
   x5  0.01803
   x6       0
   x7   2.422

c =
       x1  x2  x3  x4  x5  x6  x7
   y1   0   1   0   0   0   0   0

d =
       u1
   y1   0

Continuous-time model.
>> [y,t,x]=initial(sys,[0.1 0 0 0 0 0 0]'); %Closed-loop initial response
>> margin(sys) % Closed-loop freq. response with gain and phase margins
```

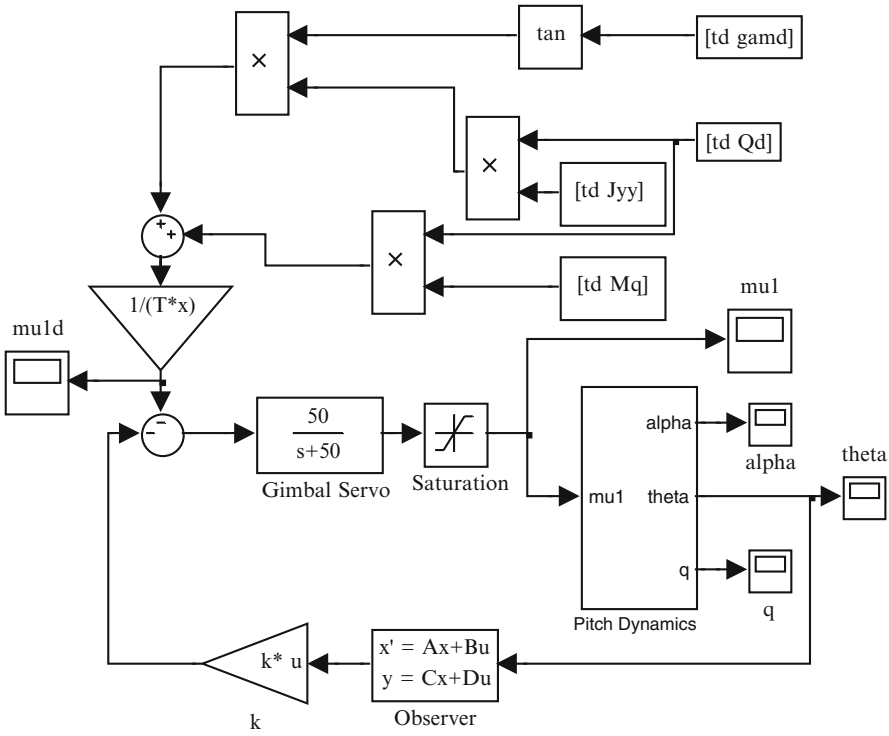


Fig. 5.23 Simulink block diagram for simulating the closed-loop pitch control system for Stage I of the *Vanguard* rocket

The nominal flight path angle, pitch-rate, and nominal gimbal angle are shown in Fig. 5.24. Figure 5.25 shows that the pitch controller – based on fixed regulator and observer gains – is able to quickly reduce the angle-of-attack and pitch-rate errors, and maintains them at nearly zero level throughout the flight, despite a large variation in the inertial and aerodynamic parameters. Only toward the end of the I stage flight, a slight oscillation is evident in the pitch angle due to a significantly reduced aerodynamic damping (M_q). The gimbal angle response never reaches the saturation level for the rather large initial angle-of-attack perturbation (5.7°). Thus, our design is quite successful. However, the pitch controller needs to be simulated in the guidance loop for a better assessment of its performance.

5.6 Summary

Flight of most rockets is largely confined to the vertical (longitudinal) plane and is usually ballistic in nature (zero lift and side force) to avoid excessive transverse loads. The guidance and control system must maintain the vehicle on a ballistic

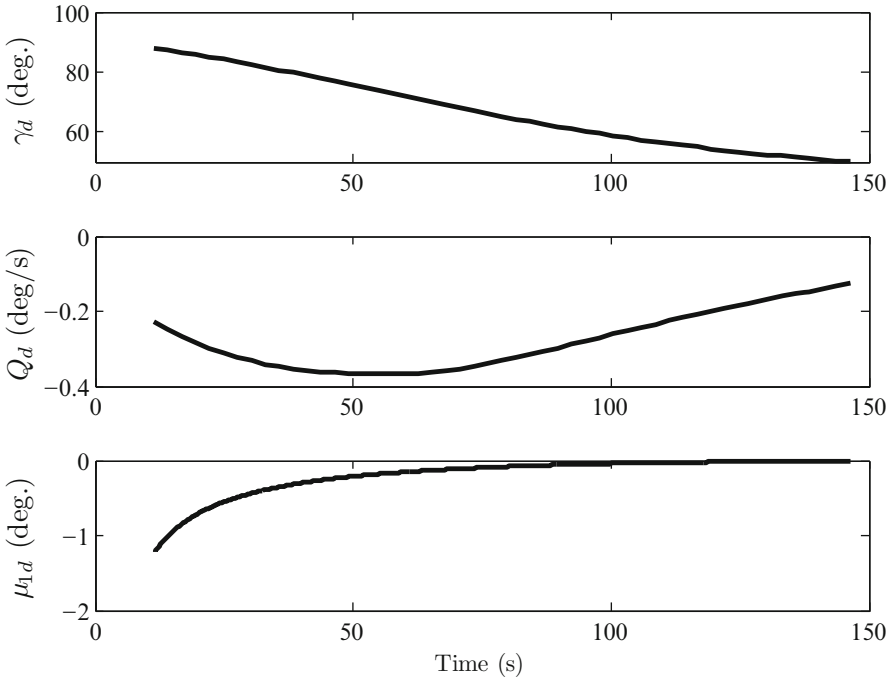


Fig. 5.24 Nominal gravity-turn flight path angle, desired pitch-rate, and gimbal angle along the nominal trajectory of Stage I of the *Vanguard* rocket

trajectory – called the gravity turn – that allows the flight path angle to change naturally from a near vertical to horizontal in a given time, under the sole influence of gravity. Due to the decoupling of roll dynamics from pitch and yaw, as well as pitch and yaw symmetry, it is possible to have separate control loops for roll, pitch, and yaw, and nearly identical regulators for pitch and yaw. While roll control is performed by either switching or linear feedback, pitch and yaw control is primarily based on driving the gimbal actuators via a linear feedback control law based on sensed attitude errors. A typical rocket is a time-varying plant due to the varying inertial and aerodynamic parameters. However, since the duration of each stage is of only a few minutes, a controller based upon fixed regulator and observer gains at the highest dynamic pressure design point is sufficient for attitude control.

Exercises

5.1. Consider a rocket with four gimbaled engines of identical, constant thrust, $T/4$, mounted symmetrically at the circular base in a ring with distance d from the base center, and a distance ξ from the center of mass. The thrust direction

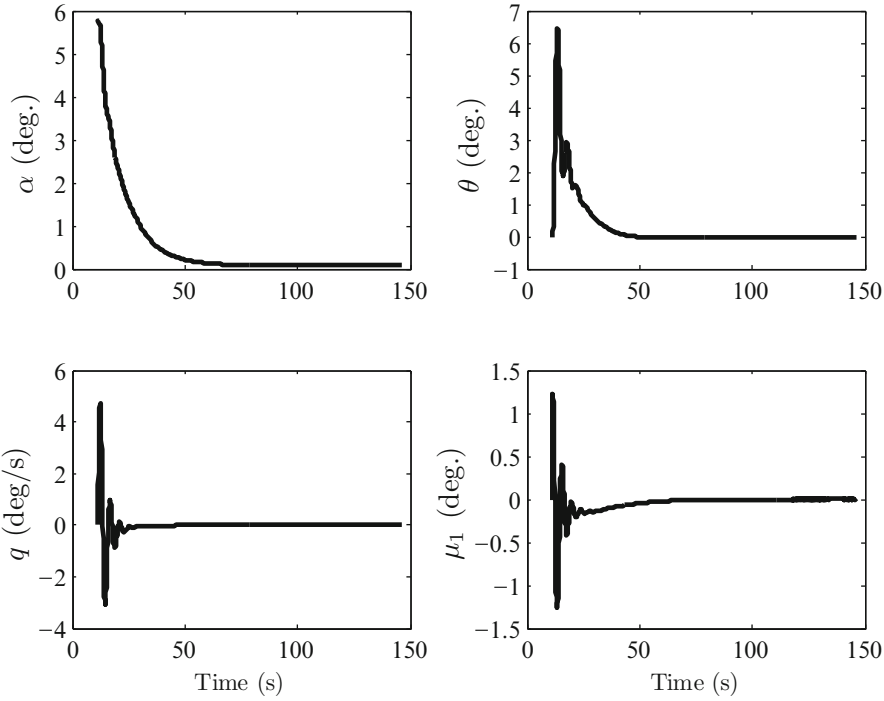


Fig. 5.25 Simulated closed-loop initial response (with servo saturation) of the frozen LQR pitch control system along the nominal trajectory of Stage I of the *Vanguard* rocket

of ith engine is controlled by swivel angle, δ_{i}, and deflection angle, μ_i, such that

$$\mathbf{T_i} \simeq \frac{T}{4}\left(\mathbf{i} - \mu_i \cos\delta_{\mathrm{i}}\mathbf{j} - \mu_i \sin\delta_{\mathrm{i}}\mathbf{k}\right).$$

Selecting the body axis oz along the direction from engine number 1 to engine number 3, and oy from 4 to 2, show that the net torque is given by

$$\begin{aligned}\boldsymbol{\tau} \simeq\ & \frac{T}{4}\xi \sum_{i=1}^{4} \mu_i \left(\cos\delta_{\mathrm{i}}\mathbf{k} - \sin\delta_{\mathrm{i}}\mathbf{j}\right) \\ & + \frac{T}{4} d \left(\mu_4 \sin\delta_4 + \mu_3 \cos\delta_3 - \mu_1 \cos\delta_1 - \mu_2 \sin\delta_2\right)\mathbf{i}.\end{aligned}$$

What is the net torque when gimbaling is carried out for transverse force balance?

5.2. Can a gimbal system be designed to maneuver a rocket sideways, but without rotating it on its axis? (This would be the case of a net transverse force with a zero moment.)

5.3. Consider a roll dynamics plant with $J_{xx} = 300$ kg m^2 and $L_p = -400$ N m s/rad. Design a suitable roll controller for achieving a desired step change of roll angle by 10° in about 1 s, without exceeding the maximum control rolling moment, $|\tau_x| \leq 4$ N m.

5.4. Consider a simplified pitch dynamics plant with $\alpha \approx 0$ and the following transfer function:

$$\frac{\theta(s)}{\mu_{1d}(s)} = -\frac{T\xi/J_{yy}}{s^2 - \frac{M_q}{J_{yy}}s}.$$

where $T = 2{,}000$ N, $J_{yy} = 6{,}000$ kg m^2, $\xi = 1$ m, and $M_q = -2{,}000$ N m s/rad. Design a suitable pitch controller for achieving a desired step change of pitch angle by 10° in about 1 s, without the pitch gimbal angle exceeding ±1°.

5.5. Why do rockets need a guide-rail (or launcher) when being launched vertically, and not when launched at an angle? Try to answer this in the light of translational dynamics in a vertical plane (do not include planetary rotation effects).

5.6. Modify the Simulink block diagram given in Fig. 5.19 to simulate the gravity-turn trajectory of a rocket from Earth's surface with a constant thrust of 10,000 N, a linearly varying mass from launch $t = 0$ until burnout, $t = 300$ s, given by

$$m(t) = 1{,}000 - 2.5t \text{ (kg)}; \qquad (0 \leq t \leq 300 \text{ s}),$$

and the following approximate aerodynamic drag given by the exponential atmospheric model:

$$D(t) = 0.25\mathrm{e}^{-h/6700}v^2 \text{ (N)}; \qquad (0 \leq t \leq 300 \text{ s}),$$

where h is the altitude in m and v is the airspeed in m/s. Assume initial conditions of $h(0) = 0, v(0) = 10$ m/s and $\gamma(0) = 85°$ (which must be specified in the respective integrator blocks). The normal acceleration input should be set as "u2 = [0 0]" in the MATLAB workspace, while the axial acceleration input, "u1," must be programmed using the relevant mathematical function blocks from the Simulink library. Plot the trajectory variables, $h(t), v(t), \gamma(t)$. What are the final airspeed and flight path angle?

5.7. Redesign the pitch controller for the Vanguard rocket of Example 5.4 by pole placement. Select optimal regulator and observer poles for meeting the design requirements. Compare the closed-loop design with that of Example 5.4.

5.8. Replace the pitch controller design for the Vanguard rocket of Example 5.4 using pitch-rate instead of pitch angle as the output variable. Select suitable cost matrices for LQR design of regulator and observer for meeting the design requirements. What are the gain and phase margins of the closed-loop system? Simulate and compare the closed-loop system with that of Example 5.4.

5.9. Redesign the pitch controller of Example 5.4 by adding the pitch-rate (along with the pitch angle) as an output variable. For the resulting two-output, single-input plant, select suitable cost matrices for LQR design of regulator and observer for meeting the design requirements. Simulate the closed-loop system along the nominal trajectory. Is there any improvement in the performance (transient response) and robustness (gain and phase margins at design point) compared to the designs of Example 5.4 and Exercise 5.8?

5.10. Replace the pitch controller design for the Vanguard rocket of Example 5.4 using a normal acceleration output, $a_z = U(\dot{\alpha} - x_n \dot{q})$, from a sensor located a distance, x_n, forward of the center of mass. Select suitable cost matrices for LQR design of regulator and observer for meeting the design requirements. What are the gain and phase margins of the closed-loop system? Simulate and compare the closed-loop system with that of Example 5.4.

5.11. Design a yaw controller for the Vanguard rocket of Example 5.4 using the frozen LQR method (Chap. 3) using the yaw rate as the feedback signal from IMU. Select the cost coefficient matrices such that the maximum gimbal angle of $\pm 5^\circ$ is never exceeded for an initial sideslip angle of 5.7°. Compare your design with that given in Example 5.4 for pitch control. Simulate the closed-loop response with the given yaw gimbal servo.

5.12. Redesign the yaw controller of Exercise 5.11 using the lateral normal acceleration output, $a_y = U(\dot{\beta} + x_n \dot{R})$, from a sensor located a distance, x_n, forward of the center of mass. Compare your design with that of Exercise 5.11 and simulate the closed-loop response along the nominal trajectory.

Chapter 6
Automatic Control of Spacecraft

6.1 Aims and Objectives

- To design and analyze planar orbit control systems, including orbit shape control with radial and tangential thrust, and nonlinear thrust vectoring for de-orbiting a satellite from circular orbit
- To design and analyze nonlinear orbital plane control systems, including those based upon switching thrust direction and smooth thrust modulation
- To design and analyze spacecraft attitude control systems with impulsive thrust, reaction wheels, momentum wheels, and control moment gyros, and taking into account gravity-gradient and solar radiation disturbances

6.2 Introduction

Spacecraft control systems can be classified into: (a) orbit control systems and (b) attitude control systems. Except in a few cases of low-orbits, the orbital and attitude dynamics of a spacecraft are uncoupled (see Chap. 2), thereby enabling an independent design and analysis of orbit and attitude controllers. This is a situation markedly different from the inertial and aerodynamic coupling between translational and rotational dynamics inherent in atmospheric flight vehicles. For this primary reason, spacecraft control system design is a relatively easy task compared to that of aircraft and rockets. However, since the spacecraft orbital and attitude motions are not confined to small perturbations, we have to contend with an essentially nonlinear plant dynamics while designing a spacecraft control system. In this chapter, we will attempt to highlight those aspects of spacecraft control systems that are different from aircraft and rockets. Our focus will be on single-variable control for clarity of basic concepts, but in later sections we will consider specific multivariable control problems.

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0_6, © Springer Science+Business Media, LLC 2011

Orbital control refers to the control of translational motion of the spacecraft in the central gravity field of a massive object (Sun, planet, etc.). We have seen in Chap. 2 that the two-body problem has analytical solutions in the form of well-defined trajectories called *orbits*. In exercising orbital control, one attempts to either modify the shape of the orbit in the fixed orbital plane, or change the plane of the orbit. In either case, the resulting orbit is suitably modified. Another common application of the orbit controller is to maintain a constant plane and shape of the orbit despite small perturbations due to gravitational anomalies (nonspherical planet, third-body gravity) and nonconservative effects (atmospheric drag, solar radiation pressure). Most artificial satellites fall in this category, and it is orbital regulation that we will be mainly concerned with here.

Attitude control refers to the task of rotating the spacecraft about its own center of mass about a desired axis. Changes in spacecraft attitude are necessary for performing its mission objectives, such as observation, mapping, and relay of telecommunication signals. The most common orientation for such satellites is the *Nadir pointing* attitude where a payload camera or antenna directly points downward, toward the planetary center. On the other hand, a spacecraft on a lunar or interplanetary voyage must re-configure its attitude from time to time, in order to point toward a fixed point for navigation. Furthermore, all spacecraft require re-orientation maneuvers when firing rockets in the desired direction for orbital control. A large spacecraft in a low-orbit must maintain a specific orientation relative to the orbit in order to have the smallest aerodynamic drag. Such an attitude can be maintained by passive means (i.e., without requiring frequent rocket firings) through the use of gravity-gradient stabilization. Thus, attitude stability and control are as crucial to a spacecraft's mission as orbital control.

6.3 Planar Orbit Control with Vectored Rocket Thrust

Consider a spacecraft of mass, m, equipped with a rocket thruster capable of producing a thrust, T, of a variable magnitude. The attitude of the spacecraft can be controlled by a separate attitude control system such that the thrust vector always lies in the orbital plane (Chap. 2) described by the position and velocity of the center of mass, $(\mathbf{r}(t), \mathbf{v}(t))$, and makes an angle, α, from the local horizontal, $\mathbf{i}_\theta$, as shown in Fig. 6.1. It is desired to control the shape of the orbit by varying the magnitude and direction of the rocket thrust. Presently, we shall neglect the rotary inertia and assume that a desired thrust angle is achieved by rotating the spacecraft instantaneously.

The equation of translational motion can be written in a vector form as follows:

$$\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} + \frac{\mu}{r^3}\mathbf{r} = \frac{T}{m}\left(\mathbf{i}_\mathbf{r}\sin\alpha + \mathbf{i}_\theta\cos\alpha\right), \tag{6.1}$$

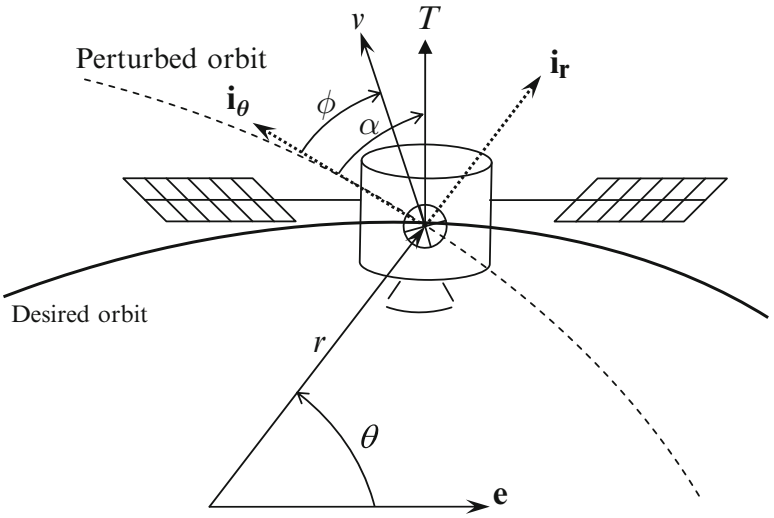


Fig. 6.1 A spacecraft with vectored rocket thrust applied in the orbital plane

or, in a scalar form by sorting the radial and circumferential velocity components as follows:

$$\ddot{r} - \frac{h^2}{r^3} + \frac{\mu}{r^2} = \frac{T}{m}\sin\alpha, \tag{6.2}$$

$$\frac{\dot{h}}{r} = \frac{T}{m}\cos\alpha. \tag{6.3}$$

A common application of an orbit controller is to maintain a given (desired) orbital shape despite disturbances. For the nominal orbit (Keplerian motion), a zero nominal thrust is required and the orbit control reduces to a regulation problem. A nominal two-body orbit (Chap. 2) can be expressed as follows:

$$\ddot{r}_n - \frac{h_n^2}{r_n^3} + \frac{\mu}{r_n^2} = 0, \tag{6.4}$$

$$\dot{h}_n = 0, \tag{6.5}$$

the solution to which is given by

$$r_n = \frac{\dfrac{h_n^2}{\mu}}{1 + e_n\cos\theta_n}, \tag{6.6}$$

where the true anomaly, $\theta_n(t)$, is determined by solving Kepler's, hyperbolic Kepler's, or Barker's equation, depending upon the nominal eccentricity, e_n. Thus,

the nominal trajectory is known and is given by $(r_n(t), h_n)$. A Taylor series expansion of the perturbed orbit about the nominal orbit value with only the linear terms retained, is the following:

$$r(t) \simeq r_n(t) + \delta r(t), \tag{6.7}$$

$$h(t) \simeq h_n + \delta h(t), \tag{6.8}$$

where $\delta r \ll r_n$ and $\delta h \ll h_n$ are the small orbital perturbations. Substituting (6.4), (6.5), (6.7), and (6.8) into (6.2) and (6.3) and using binomial expansion, we have

$$\delta\ddot{r} + \left(\frac{3h_n^2}{r_n^4(t)} - \frac{2\mu}{r_n^3(t)}\right) - \frac{2h_n}{r_n^3(t)}\delta h \simeq \frac{T}{m}\sin\alpha, \tag{6.9}$$

$$\frac{\dot{h}}{r_n(t)} \simeq \tfrac{T}{m}\cos\alpha. \tag{6.10}$$

6.3.1 Orbit Control with Radial Rocket Thrust

A common application of planar orbit control is when only the radial thrust is applied thereby changing the radius without affecting the angular momentum. Clearly, the flight path resulting from radial thrust is such that $h = rv\cos\phi$ is a constant. A common application of such a control system is maintaining a satellite in a circular orbit of desired radius, in the presence of radial orbital perturbations. Nearly all practical satellites have circular orbits, e.g., geosynchronous, sun-synchronous, and global positioning system (GPS) satellites, whose mission requirement includes a precise control of the orbital radius.

For the radial control of a satellite in a desired constant orbital radius, r_n, and a constant angular momentum, $h_n = h$, we note that

$$\frac{3h_n^2}{r_n^4} - \frac{2\mu}{r_n^3} = \frac{\mu}{r_n^3} = n^2, \tag{6.11}$$

and taking $\alpha = \pi/2$, express the orbital plant as follows:

$$\dot{\mathbf{x}}(t) = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)u(t) \qquad \mathbf{x}(0) = \mathbf{x_0}, \tag{6.12}$$

where

$$\mathbf{x} = \begin{Bmatrix} \delta r \\ \delta\dot{r} \end{Bmatrix},$$

$$u = \frac{T}{m},$$

$$\mathsf{A} = \begin{pmatrix} 0 & 1 \\ -n^2 & 0 \end{pmatrix},$$

and

$$\mathsf{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The eigenvalues of the dynamics matrix, A, reveal that the plant is stable, but not asymptotically stable:

$$|\lambda \mathsf{I} - \mathsf{A}| = \lambda \left(\lambda^2 + n^2\right) = 0, \tag{6.13}$$

or

$$\lambda_{1,2} = \pm in.$$

Thus, the task of the orbit controller is to provide active damping for reducing the effects of radial perturbations.

Example 6.1. A satellite in earth orbit has initial speed 7.8 km/s when the altitude and flight path angle are 200 km and 10°, respectively. The vehicle is equipped with a rocket thruster capable of applying radial thrust. Design a control system with radial thrust for bringing the satellite into a circular orbit of 200 km altitude within 10 min. What is the maximum radial acceleration input required?

We begin by noting that a radial thruster can provide orbital shape control without affecting the orbital angular momentum, **h**. Hence, the constant angular momentum magnitude in this case is

$$h \;=\; rv\cos\phi \;=\; (6578.14)(7.8)(\cos 10^\circ) \;=\; 50529.9855\ \text{km}^2/\text{s}.$$

The desired circular orbit has radius $r_n = 6578.14$ km for which the orbital frequency is $n = \sqrt{\mu/r_n^3} = 0.00183353$ rad/s.

Since only the radial acceleration input, $u(t) = T(t)/m(t)$, is involved, we can treat the regulator design problem by the SISO Laplace transform method of Chap. 3. The linearized differential equation for a small radial deviation from desired circular orbit, $\delta r = r(t) - r_n$, is derived above to be the following:

$$\delta\ddot{r} + n^2\delta r = u, \tag{6.14}$$

where $u(t)$ is the radial acceleration input. Taking Laplace transform of (6.14) with initial condition $\delta r(0), \delta\dot{r}(0)$, we have

$$s^2\delta R(s) - s\delta r(0) - \delta\dot{r}(0) + n^2\delta R(s) = U(s), \tag{6.15}$$

or

$$\delta R(s) = \frac{U(s) + s\delta r(0) + \delta\dot{r}(0)}{s^2 + n^2}. \tag{6.16}$$

Since the plant lacks damping, we employ a proportional-derivative (PD) controller, $U(s) = -(k_1 + k_2 s)\delta R(s)$, in order to quickly damp out the initial orbital deviation,

$\delta r(0) = 0, \delta\dot{r}(0) = (7.8)(\sin 10°) = 1.35446$ km/s resulting in the following closed-loop response:

$$\delta R(s) = \frac{1.35446}{s^2 + k_2 s + (k_1 + n^2)}. \tag{6.17}$$

Clearly, for an asymptotically stable closed-loop system we must have $k_2 > 0$. With the required closed-loop settling-time of approximately 10 min, $t_s \simeq 4/\zeta\omega_n = 600$, we choose $\zeta = 0.7$, and the controller gains become

$$k_2 = 2\zeta\omega_n = \frac{8}{600} \text{ rad/s}$$

$$k_1 = \omega_n^2 - n^2 = \left(\frac{4}{420}\right)^2 - (0.00183353)^2 = 8.9303 \times 10^{-5} \text{ rad}^2/\text{s}^2.$$

Finally, we compute the closed-loop response using MATLAB's *impulse* function as follows:

```
>> sys1=tf(1.35446,[1 8/600 (4/420)^2])

Transfer function:
          1.354
---------------------------
s^2 + 0.01333 s + 9.07e-005

>> sys2=tf([1.35446 0],[1 8/600 (4/420)^2])

Transfer function:
         1.354 s
---------------------------
s^2 + 0.01333 s + 9.07e-005

>>[y1,t1]=impulse(sys1);%% radius error (delta-r)
>>[y2,t2]=impulse(sys2);%% radial speed error (delta-rdot)

>> sys3=tf(1.35446*[-8/600 -8.9303e-5],[1 8/600 (4/420)^2])

Transfer function:
   -0.01806 s - 0.000121
---------------------------
s^2 + 0.01333 s + 9.07e-005
>> [u,t]=impulse(sys3);%% Radial acceleration input (u)
```

The resulting graphs are plotted in Fig. 6.2 ($\delta r(t), \delta\dot{r}(t)$) and Fig. 6.3 ($u(t)$), respectively. Note that a maximum radial acceleration input of -18.06 m/s^2 is required and the desired settling-time of 600 s is achieved for all transients.

6.3.2 Nonlinear De-orbiting Control System for a Circular Orbit

The task of de-orbiting spacecraft from a circular orbit for re-entry into a planetary atmosphere requires a reduction in the specific total energy that can be achieved by

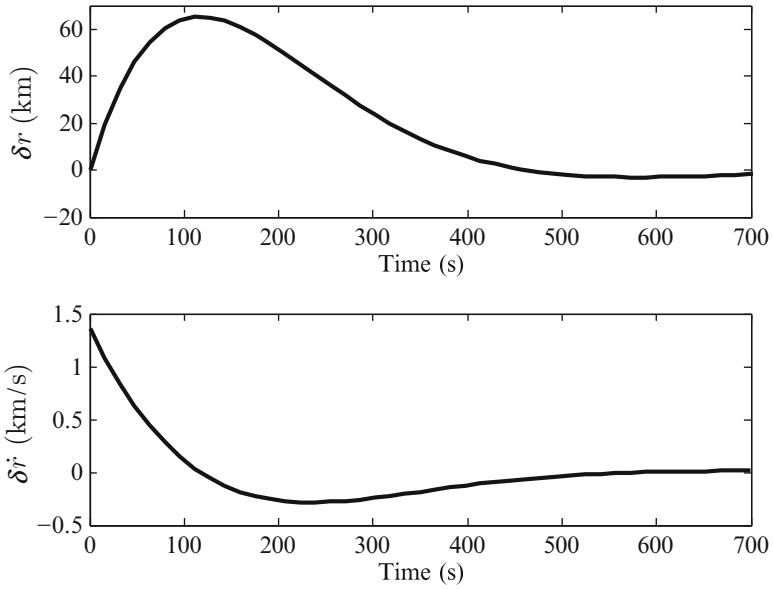


Fig. 6.2 Closed-loop transient response of satellite radius and radial speed errors from desired circular orbit

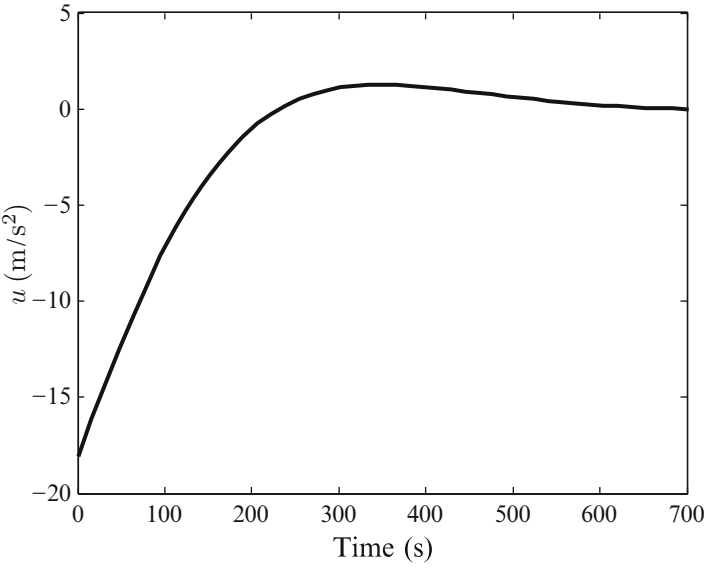


Fig. 6.3 Closed-loop radial acceleration input for orbit control of satellite in a circular orbit

decreasing either the speed, or the radius, or both. We shall consider here a special case where the speed is continually decreased while maintaining the initial orbital radius, r_0. Since the radius is maintained at $r(t) = r_0$, we have $h = r_0 v$, and the equations of planar motion with a vectored rocket thrust – (6.2) and (6.3) – become the following:

$$\frac{\mu}{r_0^2} - \frac{v^2}{r_0} = \frac{T}{m} \sin\alpha, \tag{6.18}$$

$$\dot{v} = \frac{T}{m} \cos\alpha. \tag{6.19}$$

Generally, de-orbiting imposes a particular set of conditions on the final position and velocity at the end of the control interval. The final position and velocity are governed by the duration, magnitude, and direction of thrust application (called *thrust program*), and have a profound effect on the subsequent re-entry trajectory. Therefore, the thrust program must be carefully designed for achieving the terminal conditions.

6.3.2.1 Nonlinear, Two-Point Boundary Value Problem for Thrust Direction

Let us consider the design of a control system with a constant thrust acceleration, $u = T/m = \bar{u}$, for achieving a specified final speed, $v(t_f) = v_f < \sqrt{\mu/r_0}$ at the end of the control interval, $0 \le t \le t_f$. Such a de-orbit trajectory is often the most practical to implement. Clearly, for satisfying both the equations of motion with the maximum possible deceleration, the thrust vector must be inclined such that

$$\alpha = \pi - \beta.$$

Therefore, the equations of motion become the following:

$$\frac{v_0^2 - v^2}{r_0} = \bar{u} \sin\beta, \tag{6.20}$$

$$\dot{v} = -\bar{u} \cos\beta. \tag{6.21}$$

Equation (6.21) can be alternatively expressed as follows:

$$v\dot{v} = -v\bar{u} \cos\beta,$$

or

$$\frac{\mathrm{d}v^2}{\mathrm{d}t} = -2v\bar{u} \cos\beta. \tag{6.22}$$

On differentiating (6.20) with time, we have

$$\frac{\mathrm{d}v^2}{\mathrm{d}t} = -r_0 \bar{u} \dot{\beta} \cos\beta. \tag{6.23}$$

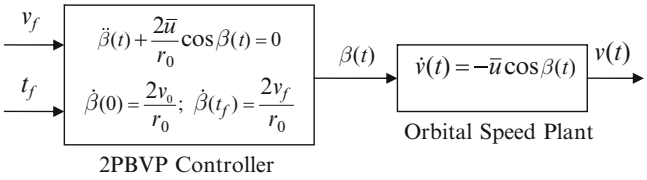


Fig. 6.4 A nonlinear 2PBVP based open-loop control system for de-orbiting a satellite from a circular orbit while maintaining a constant radius

Comparing (6.22) and (6.23) produces the following relationship between the speed and the thrust direction:

$$\dot{\beta} = \frac{2v}{r_0}. \tag{6.24}$$

The thrust direction, $\beta(t)$, must be carefully adjusted by a controller in order to achieve the desired final speed, v_f, at a terminal time, $t = t_f$. Differentiating (6.24) with time, and substituting (6.21) we have the following nonlinear differential equation for the thrust direction:

$$\ddot{\beta} + \frac{2\bar{u}}{r_0}\cos\beta = 0. \tag{6.25}$$

Equation (6.25) must be solved, subject to the initial and final conditions, $v(0) = v_0$, and $v(t_f) = v_f$, which can be written as the following boundary conditions:

$$\dot{\beta}(0) = \frac{2v_0}{r_0}; \qquad \dot{\beta}(t_f) = \frac{2v_f}{r_0}. \tag{6.26}$$

(6.25) and (6.26) constitute a *two-point boundary value problem*, whose solution, $\beta(t)$, is the required control history for meeting the desired end conditions. Thus, we have a nonlinear, open-loop control system for de-orbiting the satellite, as shown in the schematic block-diagram of Fig. 6.4.

A two-point boundary value problem (2PBVP) is a system of ordinary differential equations (ODEs) with solution variables and their derivatives specified at two different points, called the boundaries. However, when solution variables and their derivatives are specified at a single point, we have an initial value problem (IVP). 2PBVPs are more difficult to solve than IVPs.

Iterative numerical schemes are necessary for solving a 2PBVP. There are two broad 2PBVP iterative solution methods: (a) shooting methods and (b) collocation methods. A shooting method is based upon posing the boundary value problem as if it were an initial value problem (IVP), and then changing the initial conditions by trial and error until the boundary conditions are satisfied with a given accuracy. Thus, a shooting method consists of a repeated integration of the governing differential equations for various initial conditions, using a standard numerical

scheme such as the fourth-order, fifth-stage Runge–Kutta (RK-4(5)) method [19]. Simple shooting methods solve the IVP over the entire control interval, $(0, t_f)$, and can face convergence problems even if the governing system of equations is stable and well conditioned. In order to remove such difficulties, multiple shooting methods have been developed [19] that partition the control interval into several grid points, and shooting is carried out in stages.

In contrast with the shooting methods, a collocation method approximates the solution by a linear combination of a number of piecewise continuous polynomials that are usually defined on a mesh of collocation points [13, 16]. The approximate solution is then substituted into the system of ODEs such that the system is exactly satisfied at each collocation point. The number of collocation points plus the number of boundary conditions must equal the number of unknown coefficients in the approximate solution. The most common choice of approximation is a linear combination of spline functions. In order to achieve a given accuracy, the collocation points must be carefully selected.

MATLAB contains an in-built 2PBVP solver called *bvp4c.m* that is based upon a collocation scheme [15]. It integrates a system of ODEs of the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(x, \mathbf{y}),$$

on the closed interval, $a \leq x \leq b$, subject to general two-point boundary conditions of the form

$$\mathbf{g}_a[\mathbf{y}(a)] = \mathbf{0}; \qquad \mathbf{g}_b[\mathbf{y}(b)] = \mathbf{0},$$

where $\mathbf{g}_a(.), \mathbf{g}_b(.)$ are vector functionals.

Example 6.2. Using the inbuilt MATLAB 2PBVP solver, *bvp4c.m*, find the nominal thrust direction profile for an Earth satellite with $r_0 = 6{,}600$ km, $v_f = 0.9v_0$, and $t_f = 600$ s, if the constant thrust acceleration magnitude is the following:

$$\bar{u} = \frac{1.01}{r_0}|v_0^2 - v_f^2|.$$

The thrust direction differential equations for state variables, $\beta(t), \dot{\beta}(t)$, are specified in a MATLAB code called *deorbode.m* (Table 6.1), whereas the necessary boundary conditions are given in the MATLAB M-file *deorbbc.m* (Table 6.2). The solution of the 2PBVP with the given initial orbit, terminal time, and final speed is carried out through *bvp4c.m* in a MATLAB code called *deorbitconstrad.m* (Table 6.3). The resulting thrust direction angle and orbital speed are plotted in Fig. 6.5. Note an almost linear thrust direction time history, with the initial and final values of $\beta(0) = -1.447°$ and $\beta(600) = 74.762°$, respectively. This is due to the fact that there is only a ten percent decrease in the speed during the control interval, thereby having a nearly constant slope, $\dot{\beta}$ (6.24).

Table 6.1 Program *deorbode.m* for generating the thrust direction state equations for de-orbiting a satellite from a circular orbit while maintaining a constant radius

```
% Program 'deorbode.m' for generating the thrust direction
% state equations for de-orbiting a satellite
% from a circular orbit while maintaining a
% constant radius.
% (c) 2009 Ashish Tewari
% To be used by 2PBVP solver 'bvp4c.m'

function dydx=deorbode(x,y)
global u;
global r0;
global mu;
global v0;
global vf;

beta=y(1); % Thrust direction angle (rad.)
dtr=pi/180;

% Nonlinear thrust direction state equations
dydx=[y(2)
      -2*u*cos(beta)/r0];
```

Table 6.2 Program *deorbbc.m* for specifying the thrust direction boundary conditions for de-orbiting a satellite from a circular orbit while maintaining a constant radius

```
% Program 'deorbbc.m' for specifying the thrust direction
% boundary conditions for de-orbiting a satellite
% from a circular orbit while maintaining a
% constant radius.
% (c) 2009 Ashish Tewari
% To be used by 2PBVP solver 'bvp4c.m'

function res=deorbbc(ya,yb)

global u;
global r0;
global mu;
global v0;
global vf;

%Thrust direction boundary conditions
res=[ya(2)-2*v0/r0
    yb(2)-2*vf/r0];
```

6.3.2.2 Closed-Loop Regulation of Thrust Direction with Small Disturbances

When disturbances in the form of process and measurement noise are present in the open-loop control system of Fig. 6.4, substantial errors in the final speed and radius can occur from the nominal trajectory. If uncorrected, such errors can cause significant deviations in the subsequent re-entry trajectory that may prove to be disastrous for the vehicle. In such a case, a closed-loop regulator is required for maintaining the system on the nominal trajectory by driving small deviations to zero in the steady state.

Let the perturbed thrust direction be given by

$$\beta(t) = \beta_n(t) + \delta\beta(t); \qquad \delta\beta(0) = \delta\beta_0, \tag{6.27}$$

Table 6.3 Program for solving the 2PBVP for the thrust direction profile necessary for de-orbiting a satellite from a circular orbit while maintaining a constant radius

```
% Program for solving the 2PBVP for the thrust
% direction profile necessary for de-orbiting a satellite
% from a circular orbit while maintaining a
% constant radius.
% (c) 2009 Ashish Tewari
% Invokes 2PBVP solver 'bvp4c.m'
% Requires 'deorbode.m' and 'deorbbc.m'

global r0; r0=6600; %Radius (km)
global mu; mu=398600.4; %Gravitational constant (km^3/s^2)
global v0; v0=sqrt(mu/r0); % Initial speed (km/s)
global vf; vf=0.9*v0; % Final speed (km/s)
% Constant thrust acceleration:
global u; u=1.01*abs(v0^2-vf^2)/r0; %(km/s^2)
dtr=pi/180;
tf=600;
% Collocation points and initial guess:
initguess = [asin((v0^2-vf^2)/(r0*u)) 2*vf/r0];
solinit = bvpinit(linspace(0,tf,50),initguess);

% 2PBVP solution for thrust direction:
sol = bvp4c(@deorbode,@deorbbc,solinit);
x = linspace(0,tf); % Time vector (s)
y = deval(sol,x); % Solution vector:
                  % y(1,:): beta (rad.); y(2,:): d(beta)/dt (rad/s)
subplot(211),plot(x,y(1,:)/dtr),xlabel('Time (s)'),
ylabel('\beta (deg.)'),
hold on,
subplot(212),plot(x,r0*y(2,:)/2),xlabel('Time (s)'),
ylabel('v (km/s)')
```

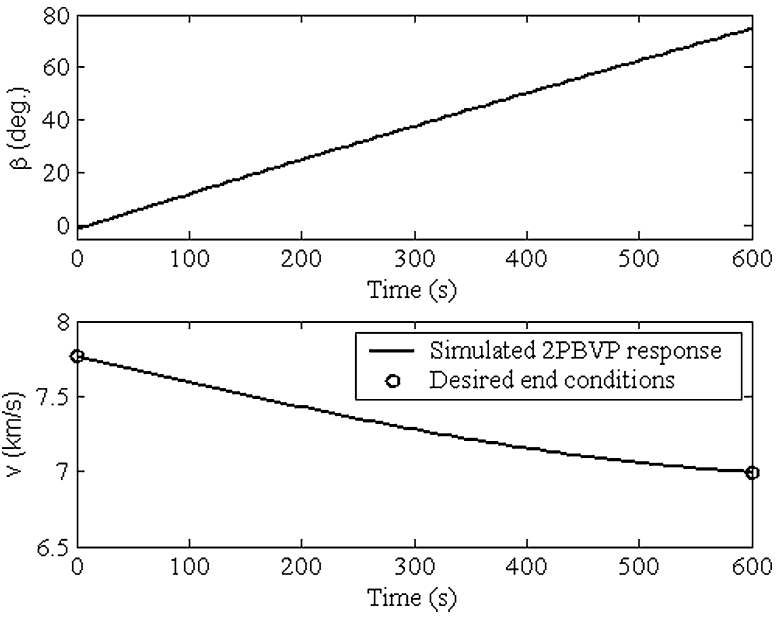


Fig. 6.5 Simulated response of a nonlinear 2PBVP-based open-loop control system for de-orbiting a satellite from a circular orbit, while maintaining a constant radius

where β_n is the nominal thrust direction computed above from the solution of a nonlinear 2PBVP, and $\delta\beta$ is a small perturbation. In order to study the stability of the perturbed motion, we confine our attention to an initial perturbation, $\delta\beta_0$, due to process and measurement noise acting at previous times. Substituting (6.27) into (6.25), we have

$$\ddot{\beta}_n + \delta\ddot{\beta} + \frac{2\bar{u}}{r_0}(\cos\beta_n \cos\delta\beta - \sin\beta_n \sin\delta\beta) = 0,$$

or, making the small angle approximation, $\cos\delta\beta \simeq 1$, $\sin\delta\beta \simeq \delta\beta$, we can write the approximate linear differential equation for perturbation as follows:

$$\delta\ddot{\beta} - \frac{2\bar{u}\sin\beta_n}{r_0}\delta\beta = 0. \tag{6.28}$$

Clearly, the motion is stable (but not asymptotically stable) for $\sin\beta_n < 0$, and unstable otherwise. In order to achieve unconditional asymptotic stability, a closed-loop control of speed with thrust angle input is necessary.

Consider a small error from the nominal speed, $v_n(t)$, given by

$$\delta v(t) = v(t) - v_n(t); \qquad \delta v(0) = \delta v_0. \tag{6.29}$$

Substitution of (6.29) into (6.21) gives us the following approximate, linear, time-varying speed plant:

$$\delta\dot{v}(t) = \bar{u}\sin\beta_n(t)\delta\beta(t). \tag{6.30}$$

For this first-order plant, we choose a proportional feedback controller of the form

$$\delta\beta(t) = -k(t)\delta v(t), \tag{6.31}$$

where $k(t)$ is a time-varying gain depending upon the nominal trajectory. Substituting (6.31) into (6.30), we have

$$\delta\dot{v}(t) + [k(t)\bar{u}\sin\beta_n(t)]\,\delta v(t) = 0; \qquad \delta v(0) = \delta v_0. \tag{6.32}$$

It is clear that for asymptotic stability, one must have

$$k(t)\bar{u}\sin\beta_n(t) > 0$$

at all times. As we know from Example 6.2, it is possible to have $\beta_n(t) \leq 0$ at some time that could be problematic. A possible choice for the controller gain could be the following:

$$k(t) = \begin{cases} \dfrac{1}{\tau\bar{u}\sin\beta_n(t)}; & \beta_n(t) \neq 0 \\ 0; & \beta_n(t) = 0 \end{cases} \tag{6.33}$$

Here, $\tau > 0$ is a pre-selected time-constant of the desired feedback response.[1]

[1] If one neglects the behavior close to $\beta_n(t) = 0$, the closed-loop response with the controller gain of (6.33) is the following:

$$\delta v(t) = \delta v_0 \mathrm{e}^{-t/\tau}.$$

The closer we are to the $\beta_n(t) = 0$ point, the larger the magnitude of $k(t)$ results from (6.33), causing a large deflection, $\delta\beta(t)$. In order to restrict the gain to reasonable values, the difficulty around $\beta_n(t) = 0$ is avoided by having a dead-zone, wherein no feedback is applied unless the magnitude of $\beta_n(t)$ is larger than a given value. Thus, (6.33) is modified to include a dead-zone as follows:

$$k(t) = \begin{cases} \dfrac{1}{\tau\bar{u}\sin\beta_n(t)}; & |\beta_n(t)| \geq \epsilon \\ 0; & |\beta_n(t)| < \epsilon \end{cases} \tag{6.34}$$

where ϵ is a small positive number called deadband. The deadband is carefully selected by considering the maximum allowable magnitude of the control input, such as

$$|\delta\beta|_{\max} = \frac{|\delta v_0|_{\max}}{\tau\bar{u}\sin\epsilon}.$$

The deadband delays the application of the control input, which can have the two-fold effect of: (a) prolonging the settling-time of the transient, and (b) increasing the output error if the system is unstable. Hence, a larger deadband requires a smaller controller gain, but could result in a larger input magnitude that acts for a longer time.

A schematic block diagram of the closed-loop system is shown in Fig. 6.6. Note the presence of a feedforward controller for generating the nominal control, $\beta_n(t)$, in addition to the feedback regulator with a time-varying gain. The unmodeled process noise, $p(t)$, and measurement noise, $m(t)$, are shown as random disturbances.

Example 6.3. For the nominal trajectory obtained in Example 6.2, design a constant thrust acceleration magnitude, feedback regulator such that an initial speed error decays in less than 20 s. Use $\delta v_0 = 0.01$ km/s.

A MATLAB code called *deorbitreg.m* (Table 6.4) is written in order to generate the closed-loop equations of motion for the perturbed trajectory with the nonlinear speed plant, feedback speed regulator (6.34) with a given deadband values, ϵ, and a feedforward integral controller that generates the nominal thrust direction, as shown in the block-diagram of Fig. 6.6. The closed-loop state equations generated by *deorbitreg.m* are integrated using the intrinsic fourth-order, fifth-stage, Runge–Kutta solver of MATLAB, *ode45.m* with the calling program, *rundeorbitreg.m* (Table 6.5) that also post-processes the results to produce the speed error profile, $\delta v(t)$, and the thrust correction angle, $\delta\beta(t)$, using another subroutine called *deorbitbeta.m* (Table 6.6).

Results for three different deadband values, $\epsilon = 0.2, 0.3, 0.4$, are plotted in Figs. 6.7 and 6.8. Note that a reduction of the deadband causes an increase in the thrust deflection magnitude, but a decrease in the settling-time.

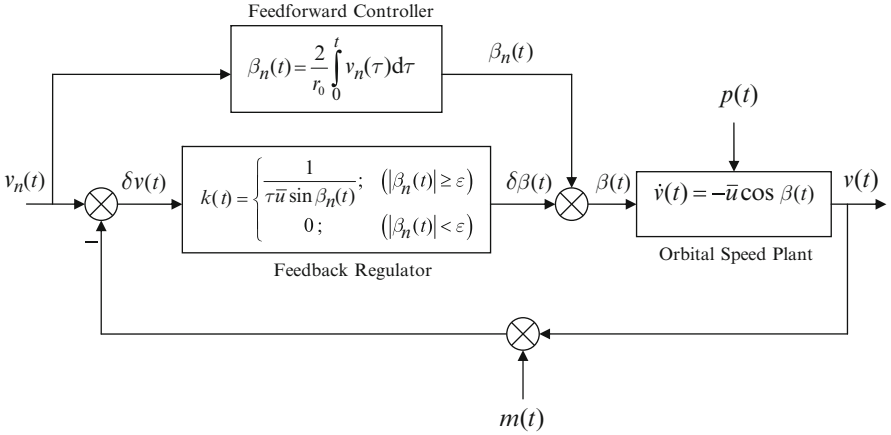


Fig. 6.6 A closed-loop control system with feedback regulator and feedforward controller for de-orbiting a satellite from a circular orbit while maintaining a constant radius

Table 6.4 Program *deorbitreg.m* for generating the closed-loop equations of motion for deorbiting with the nonlinear speed plant, a feedback speed regulator with deadband, and a feedforward integral controller for the nominal thrust direction

```
% Program 'deorbitreg.m' for generating the closed-loop equations
% of motion for deorbiting with the nonlinear speed plant,
% a feedback speed regulator with deadband,  and a
% feedforward integral controller for the nominal thrust
% direction.
% (c) 2009 Ashish Tewari
function xdot=deorbitreg(t,x)
global r0;
global mu;
global v0;
global vf;
global u;
global eps;
global tn;
global vn;
global betan;
% Interpolation for nominal speed & thrust direction at time,t:
beta=interp1(tn,betan,t);
v=interp1(tn,vn,t);
% Feedback regulator with deadband:
T=100;
if abs(beta)<=eps
    k=0;
else
k=1/(T*u*sin(beta));
end
dbeta=-k*(x-v); % Thrust direction correction law (rad)
xdot=-u*cos(beta+dbeta); % Nonlinear acceleration (km/s^2)
```

Table 6.5 Program for integrating the closed-loop equations of motion for deorbiting with feedback/feedforward control system using Runge–Kutta 4(5) algorithm

```
% Program for integrating the closed-loop state equations
% of motion for deorbiting with feedback/feedforward
% control system using Runge--Kutta 4(5) algorithm, 'ode45.m'.
% (c) 2009 Ashish Tewari
global r0; r0=6600; %Radius (km)
global mu; mu=398600.4;
global v0; v0=sqrt(mu/r0);
global vf; vf=0.9*v0;
global u; u=1.01*abs(v0^2-vf^2)/r0; % km/s^2
global eps; eps=0.4; % deadband
% Nominal trajectory stored at specific times instants:
global tn;
tn=[0 6.0606 12.1212 18.1818 24.2424 30.3030 36.3636 42.4242 ...
  48.4848 54.5455 60.6061 66.6667 72.7273 78.7879 84.8485 ...
  90.9091 96.9697 103.0303 109.0909 115.1515 121.2121 ...
  127.2727 133.3333 139.3939 145.4545 151.5152 157.5758 ...
  163.6364 169.6970 175.7576 181.8182 187.8788 193.9394 ...
  200.0000 206.0606 212.1212 218.1818 224.2424 230.3030 ...
  236.3636 242.4242 248.4848 254.5455 260.6061 266.6667 ...
  272.7273 278.7879 284.8485 290.9091 296.9697 303.0303 ...
  309.0909 315.1515 321.2121 327.2727 333.3333 339.3939 ...
  345.4545 351.5152 357.5758 363.6364 369.6970 375.7576 ...
  381.8182 387.8788 393.9394 400.0000 406.0606 412.1212 ...
  418.1818 424.2424 430.3030 436.3636 442.4242 448.4848 ...
  454.5455 460.6061 466.6667 472.7273 478.7879 484.8485 ...
  490.9091 496.9697 503.0303 509.0909 515.1515 521.2121 ...
  527.2727 533.3333 539.3939 545.4545 551.5152 557.5758 ...
  563.6364 569.6970 575.7576 581.8182 587.8788 593.9394 ...
  600.0000];
global betan;
betan=[-0.0253 -0.0110 0.0033 0.0175 0.0317 0.0459 0.0600 0.0742 ...
    0.0883 0.1024 0.1165 0.1306 0.1446 0.1586 0.1726 0.1866 ...
    0.2006 0.2146 0.2285 0.2424 0.2563 0.2702 0.2840 0.2979 ...
    0.3117 0.3255 0.3393 0.3530 0.3668 0.3805 0.3942 0.4079 ...
    0.4216 0.4352 0.4489 0.4625 0.4761 0.4897 0.5033 0.5168 ...
    0.5303 0.5439 0.5574 0.5708 0.5843 0.5978 0.6112 0.6246 ...
    0.6380 0.6514 0.6648 0.6781 0.6915 0.7048 0.7181 0.7314 ...
    0.7447 0.7580 0.7712 0.7845 0.7977 0.8109 0.8241 0.8373 ...
    0.8505 0.8636 0.8768 0.8899 0.9030 0.9161 0.9292 0.9423 ...
    0.9554 0.9684 0.9815 0.9945 1.0076 1.0206 1.0336 1.0466 ...
    1.0596 1.0726 1.0855 1.0985 1.1115 1.1244 1.1373 1.1503 ...
    1.1632 1.1761 1.1890 1.2019 1.2148 1.2277 1.2406 1.2534 ...
    1.2663 1.2791 1.2920 1.3048];
global vn;
vn=[7.7714 7.7607 7.7501 7.7394 7.7288 7.7182 7.7075 7.6969 ...
    7.6863 7.6757 7.6651 7.6546 7.6440 7.6335 7.6230 7.6125 ...
    7.6021 7.5917 7.5813 7.5710 7.5606 7.5504 7.5401 7.5299 ...
    7.5198 7.5097 7.4996 7.4896 7.4796 7.4697 7.4599 7.4501 ...
    7.4403 7.4307 7.4210 7.4115 7.4020 7.3926 7.3832 7.3739 ...
    7.3647 7.3556 7.3465 7.3375 7.3286 7.3197 7.3110 7.3023 ...
    7.2937 7.2852 7.2768 7.2685 7.2602 7.2521 7.2440 7.2360 ...
    7.2282 7.2204 7.2127 7.2051 7.1976 7.1903 7.1830 7.1758 ...
    7.1687 7.1618 7.1549 7.1482 7.1415 7.1350 7.1286 7.1222 ...
    7.1160 7.1100 7.1040 7.0981 7.0924 7.0868 7.0813 7.0759 ...
    7.0706 7.0655 7.0604 7.0555 7.0507 7.0461 7.0416 7.0371 ...
    7.0329 7.0287 7.0247 7.0208 7.0170 7.0134 7.0098 7.0065 ...
    7.0032 7.0001 6.9971 6.9942];
% Runge-Kutta 4(5) integration for perturbed speed
[t,X]=ode45('deorbit_reg',[0 600],v0+0.01);
% Post-processing the results to generate
% speed error, dv, and thrust correction angle, dbeta
[dbeta,dv]=deorbit_beta(t,X);
```

Table 6.6 Program *deorbitbeta.m* for generating the speed error and the thrust correction angle from integrated closed-loop results for deorbiting a spacecraft from a circular orbit

```
% Program 'deorbitbeta.m' for generating the speed error
% and the thrust correction angle from integrated
% closed-loop results for deorbiting a spacecraft
% from a circular orbit.
% (c) 2009 Ashish Tewari
function [dbeta,dv]=deorbitbeta(t,V)
global r0;
global mu;
global v0;
global vf;
global u;
global eps;
global tn;
global vn;
global betan;

T=100;
dv=[];dbeta=[];
for i=1:size(t,1)
beta=interp1(tn,betan,t(i,1));
v=interp1(tn,vn,t(i,1));
if abs(beta)<=eps
    k=0;
else
k=1/(T*u*sin(beta));
end
dv(i,:)=V(i,1)-v;
dbeta(i,:)=-k*(V(i,1)-v);
end
```

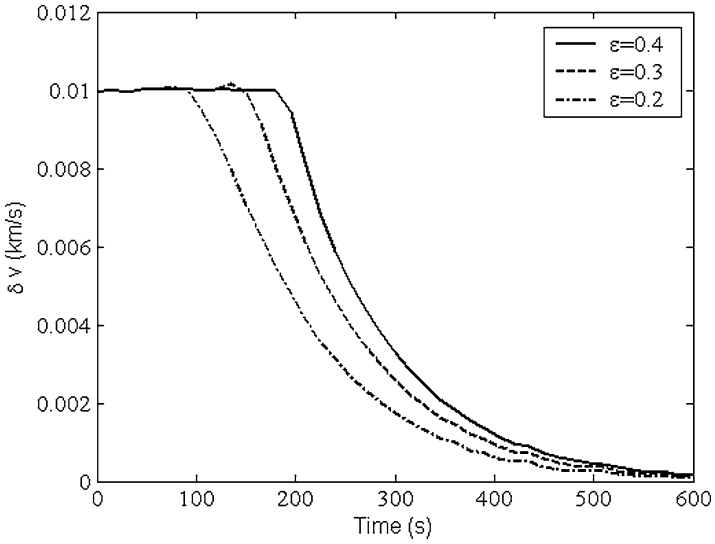


Fig. 6.7 Speed error response of the closed-loop, regulated system for de-orbiting a satellite from a circular orbit while maintaining a constant radius

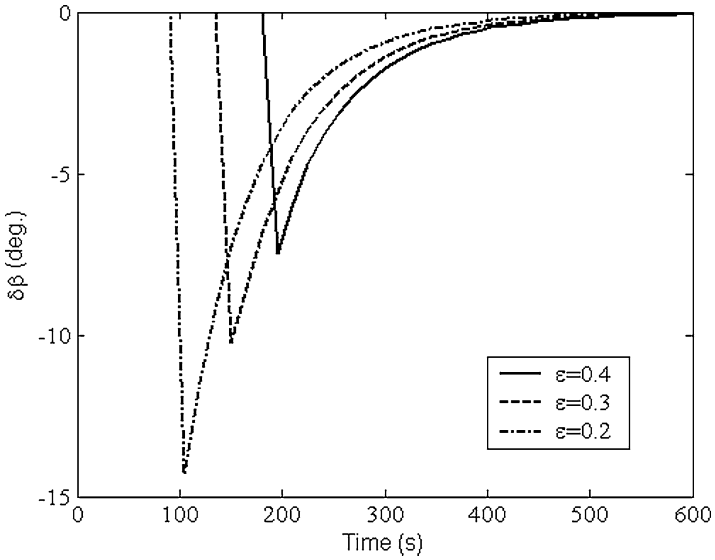


Fig. 6.8 Thrust direction correction input of the closed-loop, regulated system for de-orbiting a satellite from a circular orbit while maintaining a constant radius

6.4 Orbital Plane Control with Vectored Rocket Thrust

Consider a spacecraft with a nominal orbital plane given by the angular momentum vector, $\mathbf{h_n}$. Suppose there is an error in the orbital plane by an angle α such that the actual angular momentum vector is given by

$$\mathbf{h} = \mathbf{h_n} + \delta\mathbf{h}, \tag{6.35}$$

but the magnitude of angular momentum is unchanged from the nominal value,

$$h = |\mathbf{h}| = |\mathbf{h_n} + \delta\mathbf{h}| = h_n. \tag{6.36}$$

Clearly, a control torque must be applied to the orbital plane in order to restore the nominal angular momentum direction,

$$\mathbf{i_h}^n = \frac{\mathbf{h_n}}{h_n}.$$

Let the normal acceleration input causing rotation of the orbital plane be $\mathbf{u}(t)$, as shown in Fig. 6.9. Then the dynamic orbital equation becomes the following:

$$\dot{\mathbf{v}} = \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{u}(t) - \frac{\mu\mathbf{r}}{r^2}. \tag{6.37}$$

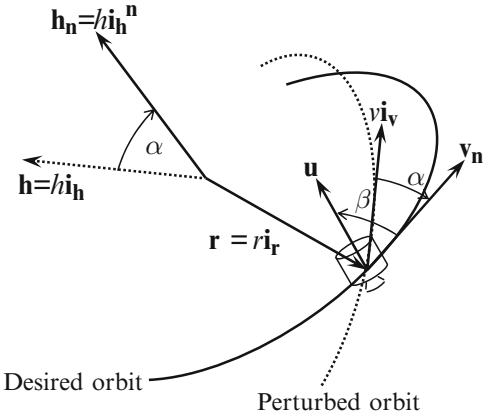


Fig. 6.9 A spacecraft with thrust controlled orbital plane

The angular acceleration of the orbital plane is obtained by taking a cross product of (6.37) with the radius vector $\mathbf{r}(t)$:[2]

$$\dot{\mathbf{h}} = \frac{\mathrm{d}\mathbf{h}}{\mathrm{d}t} = \mathbf{r}(t) \times \mathbf{u}(t). \tag{6.38}$$

Often, the duration in which the plane change occurs is quite small in comparison with the orbital period, leading to an impulsive (or instantaneous) plane change approximation. Here, we shall make a similar assumption that the plane change takes place so quickly that there is little time for a change in either the orbital radius vector or the speed due to planetary gravitation. Since the magnitude of the angular momentum (thus orbital shape) is also unaffected by the acceleration input normal to orbital plane, we have

$$\mathbf{r} = \mathbf{r_n}; \qquad v = |\,\mathbf{v_n}\,| = v_n,$$

whereby the direction of $\mathbf{u}(t)$ is given by the angle, β, measured from the original velocity vector as shown in Fig. 6.9.

The displaced orbital plane can be defined by the local horizon frame, $(\mathbf{i_r}, \mathbf{i}_\theta, \mathbf{i_h})$, as shown in Fig. 6.10, and its orientation relative to the nominal plane,

[2]
$$\dot{\mathbf{h}} = \dot{\mathbf{r}} \times \mathbf{v} + \mathbf{r} \times \dot{\mathbf{v}} = \mathbf{v} \times \mathbf{v} + \mathbf{r} \times \dot{\mathbf{v}} = \mathbf{r} \times \dot{\mathbf{v}},$$
$$\mathbf{r} \times \dot{\mathbf{v}} = \mathbf{r} \times \mathbf{u} - \frac{\mu(\mathbf{r} \times \mathbf{r})}{r^2} = \mathbf{r} \times \mathbf{u}.$$

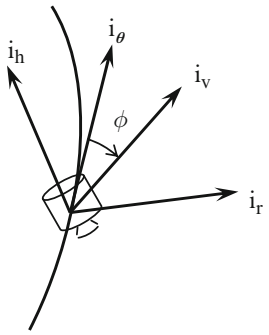


Fig. 6.10 Local horizon frame, $(\mathbf{i}_\mathbf{r}, \mathbf{i}_\theta, \mathbf{i}_\mathbf{h})$, representing the orbital plane

$(\mathbf{i}_\mathbf{r}, \mathbf{i}_\theta^\mathbf{n}, \mathbf{i}_\mathbf{h}^\mathbf{n})$, (Fig. 6.9) is represented by principal rotation angle, $-\alpha$, about the nodal vector, $\mathbf{i}_\mathbf{r}$:

$$\begin{Bmatrix} \mathbf{i}_\mathbf{r} \\ \mathbf{i}_\theta^\mathbf{n} \\ \mathbf{i}_\mathbf{h}^\mathbf{n} \end{Bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{Bmatrix} \mathbf{i}_\mathbf{r} \\ \mathbf{i}_\theta \\ \mathbf{i}_\mathbf{h} \end{Bmatrix}. \tag{6.39}$$

Hence, the perturbed angular momentum can be expressed as follows:

$$\mathbf{h} = h_n \mathbf{i}_\mathbf{h} = h_n \left(\mathbf{i}_\mathbf{h}^\mathbf{n} \cos\alpha - \mathbf{i}_\theta^\mathbf{n} \sin\alpha \right). \tag{6.40}$$

Referring to Fig. 6.9, we can express the normal acceleration input as follows:

$$\mathbf{u} = u \left(\mathbf{i}_\mathbf{v} \cos\beta + \mathbf{i}_\mathbf{h} \sin\beta \right), \tag{6.41}$$

where the velocity direction, $\mathbf{i}_\mathbf{v}$, is resolved in the local horizon frame (Fig. 6.10) through the flight path angle, ϕ according to

$$\mathbf{i}_\mathbf{v} = \mathbf{i}_\mathbf{r} \sin\phi + \mathbf{i}_\theta \cos\phi. \tag{6.42}$$

Therefore, we have

$$\begin{aligned} \dot{\mathbf{h}} &= \mathbf{r} \times \mathbf{u} = r \mathbf{i}_\mathbf{r} \times \mathbf{u} \\ &= ru \left[(\cos\alpha \cos\phi \cos\beta - \sin\beta \sin\alpha) \, \mathbf{i}_\mathbf{h}^\mathbf{n} \right. \\ &\qquad \left. - (\sin\beta \cos\alpha + \sin\alpha \cos\beta \cos\phi) \, \mathbf{i}_\theta^\mathbf{n} \right]. \end{aligned} \tag{6.43}$$

Taking the time derivative of (6.40) and comparing with (6.43) produces the following governing equations for the plane change maneuver:

$$\dot{\alpha} = \frac{ru \sin\beta}{h}, \tag{6.44}$$

and

$$\tan\alpha = -\frac{\cos\alpha\cos\beta\cos\phi - \sin\alpha\sin\beta}{\cos\alpha\sin\beta + \sin\alpha\cos\beta\cos\phi}. \tag{6.45}$$

Here, $u(t)$ is the normal acceleration input magnitude and $\beta(t)$ is its direction.

Equation (6.45) relates the direction, $\beta(t)$, of the normal acceleration input with the angular perturbation, $\alpha(t)$ and the flight path angle, $\phi(t)$. Since the shape of the orbit is unaffected by the normal acceleration input, the flight path angle can be approximated by its nominal value, $\phi_n(t)$ that is determined from the true anomaly, $\theta_n(t)$, as follows:

$$\phi_n = \tan^{-1}\left\{\frac{e_n\sin\theta_n}{1+e_n\cos\theta_n}\right\}. \tag{6.46}$$

Thus, given the perturbation, α, one must solve (6.45) in order to calculate at what angle the plane change input must be applied, such that the shape of the orbit is unaffected. Such a solution requires an iterative, numerical scheme due to the transcendental nature of (6.45).

However, a simpler solution of the thrust direction angle is provided by considering the case of $\phi = 0$, for which (6.45) becomes the following:

$$\tan\alpha = -\cot(\alpha+\beta), \tag{6.47}$$

or $\beta = -\text{sgn}(\alpha)\pi/2$, where sgn(.) denotes the signum function. Interestingly, while derived for $\phi = 0$, this value of β identically satisfies (6.45) even for a non–zero flight path angle! Thus, we take

$$\beta(t) = -\text{sgn}(\alpha)\frac{\pi}{2}$$

as the unique thrust direction to be applied in all cases of plane change maneuvers. Consequently, (6.44) becomes the following:

$$\dot{\alpha} = -\frac{ru}{h} = -\frac{ru}{h_n}. \tag{6.48}$$

Equation (6.48) is rendered linear by neglecting the change of radius in the small time interval of plane change maneuver, i.e., $r \simeq r_n = \text{const.}$:

$$\dot{\alpha} = -\frac{r_n}{h_n}u. \tag{6.49}$$

Such a model is invaluable in control system design with a continuous input magnitude, $u(t)$. Taking the Laplace transform of (6.49) with initial condition, $\alpha(0) = \alpha_0$, we have

$$\alpha(s) = \frac{1}{s}\left[\alpha_0 - \frac{r_n}{h_n}u(s)\right], \tag{6.50}$$

or,

$$\alpha(t) = \alpha_0 - \frac{r_n}{h_n}\int_0^t u(\tau)\mathrm{d}\tau \qquad (t \geq 0). \tag{6.51}$$

6.4.1 Constant Acceleration, Switching Control of Orbital Plane

The most common application of plane change maneuver is with a constant rocket acceleration input acting for a fixed duration, t_f, whereby $u(t) = \bar{u} = \text{const.}$, and the orbital plane inclination becomes

$$\alpha(t) = \alpha_0 - \frac{r_n \bar{u} t}{h_n}; \qquad (t \leq t_\mathrm{f}) \tag{6.52}$$

Clearly, the duration of rocket firing, t_f, can be adjusted as follows in order to make the final orbital inclination vanish:

$$t_\mathrm{f} = \frac{h_n \mid \alpha_0 \mid}{r_n \bar{u}}. \tag{6.53}$$

The direction of the thrust application is given by

$$\beta(t) = -\mathrm{sgn}[\alpha(t)]\frac{\pi}{2} \tag{6.54}$$

requiring a feedback of the inclination error, $\alpha(t)$. Thus, we have a nonlinear controller for the plane change maneuver, requiring a switching of the thrust direction, $\beta(t)$, and turning the rocket motor with a constant thrust acceleration, $\bar{u}$, on and off at the desired instants.

While simple to design, a switching controller could be difficult to implement where measurement and process noise, as well as timing errors cause a rapid switching of $\beta(t)$ about the nominal $\alpha = 0$ state, called *chatter*. A successful implementation of a switching controller is thus based upon precision timing as well as precise measurement of the angular deviation, $\alpha(t)$. In order to reduce the susceptibility of the system to small but random error in the output, $\alpha(t)$, a dead-zone is normally built in around the nominal $\alpha = 0$ state. No control input is generated when the system falls into the dead-zone, i.e.,

$$\mathbf{u}(t) = \mathbf{0}; \qquad |\alpha(t)| \leq \epsilon;,$$

where $\epsilon > 0$ is a small number called the deadband. However, with a dead-zone, one can never precisely achieve the nominal $\alpha = 0$ state. A schematic block diagram of the switching orbital plane control system is given in Fig. 6.11.

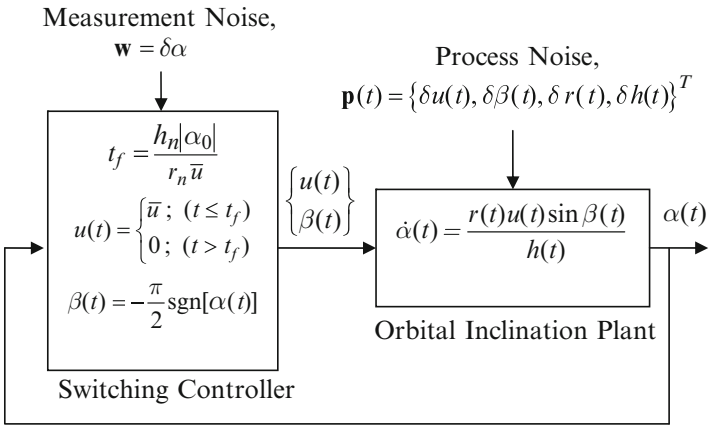


Fig. 6.11 A switching control system for orbital inclination with a constant rocket thrust

Example 6.4. A spacecraft has the following initial orbital elements relative to the Earth's equatorial, celestial frame:

$$a(0) = 7{,}000 \text{ km}; \qquad e(0) = 0.5; \qquad \tau(0) = -2{,}000 \text{ s},$$
$$i(0) = 32°; \qquad \omega(0) = 200°; \qquad \Omega(0) = 50°.$$

It is desired to decrease the inclination of the orbital plane by 2° without changing the shape of the orbit, such that the final orientation is the following:

$$i = 30°; \qquad \omega = 200°; \qquad \Omega = 50°.$$

Using a constant rocket thrust acceleration, design a switching control system for this task.

Since only the orbital inclination is changed, we have constant orbital elements $a, e, \tau, \omega, \Omega$ at their respective initial values. We begin by encoding the calculation of position and velocity for the given elliptical orbit in a MATLAB code *orbit.m* (Table 6.7) that calls *trajE.m* (Table 6.8), *rotation.m* (Table 6.9), and *kepler.m* [21]. The nominal position and velocity at $t = 0$ are thus obtained by the following calling statements:

```
>> dtr=pi/180;
>> orb=[7000; 0.5; 30*dtr; 200*dtr; 50*dtr; -2000];
>> mu=398600.4; %Gravitational constant of Earth (km^3/s^2)
>> [rn,vn]=orbit(mu,orb,0) %Nominal position, velocity @t=0
```

which produce the following output:

```
rn =
          6572.16071498624
          7178.16526461591
         -242.793976241903

vn =
         -1.82024587036952
          4.03988173258303
          2.30430605093637
```

Table 6.7 Program *orbit.m* for position and velocity of a body in a specified elliptical orbit

```
function [r,v]=orbit(mu,orb,t)
% program for position and velocity of a body in 'orb' elliptical orbit.
% Elements of 'orb': 1x1: a; 2x1:e; 3x1:i; 4x1:w; 5x1: Om; 6x1: tau.
% Requires 'kepler.m','trajeE.m', and 'rotation.m'.
% (c) 2009 Ashish Tewari
a=orb(1);e=orb(2);i=orb(3);w=orb(4);Om=orb(5);tau=orb(6);
n=sqrt(mu/a^3);
M=-n*tau;
E=kepler(e,M);
r0=a*(1-e*cos(E));
R0=a*[cos(E)-e;sqrt(1-e^2)*sin(E);0];
V0=sqrt(mu*a)*[-sin(E);sqrt(1-e^2)*cos(E);0]/r0;
[R,V]=trajE(mu,0,R0,V0,t);
if abs(i)>=1e-6
C=rotation(i,Om,w);
else
    C=eye(3);
end
r=C*R;
v=C*V;
```

Next, another MATLAB code called *planechange.m* (Table 6.10) is written in order to simulate the orientation error, $\alpha(t)$, of the orbital plane by numerically integrating the following switch controlled nonlinear, orbital dynamics equations, with the use of Runge–Kutta 4(5) method programmed in the MATLAB intrinsic code *ode45.m*:

$$\dot{\mathbf{r}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{u} - \frac{\mu \mathbf{r}}{r^2},$$

where the constant normal acceleration control input is the following:

$$\mathbf{u}(t) = \begin{cases} -\bar{u}\ \mathrm{sgn}[\alpha(t)]\mathbf{i_h}; & \left(t \le \dfrac{h_n \mid \alpha_0 \mid}{r_n \bar{u}}\right) \\ \\ 0; & (t > 0) \end{cases}$$

For the present, a dead-zone around the nominal $\alpha = 0$ state is not enforced. The call to *planechange.m* is made by the following sequence of statements:

```
dtr=pi/180;
mu=398600.4;
orb0=[7000; 0.5; 32*dtr; 200*dtr; 50*dtr; -2000];
[rn0,vn0]=orbit(mu,orb0,0);
OPTIONS = odeset('RelTol', 1e-8);
[t,x]=ode45('planechangeopen',[0 250],[rn0;vn0],OPTIONS);
```

The post-processing of the integration results is carried out with the another MATLAB code called *postproplane.m* (Table 6.11). Results thus generated for two different values of input acceleration magnitude, $\bar{u} = 1\ \mathrm{m/s^2}$ and $\bar{u} = 10\ \mathrm{m/s^2}$, are compared in Figs. 6.12–6.15. While the desired angular momentum is precisely

Table 6.8 Program *trajE.m* for calculating perifocal position and velocity for a given elliptical orbit using Lagrange's coefficients

```
% Program 'trajE.m' for calculating perifocal position and velocity
% for a given elliptical orbit using Lagrange's coefficients.
% This code requires 'kepler.m'
% (c) 2006 Ashish Tewari
function [R,V]=trajE(mu,t0,R0,V0,t)
eps=1e-10;
r0=norm(R0);
v0=norm(V0);
alpha=dot(R0,V0);
H=cross(R0,V0);
h=norm(H);
p=h*h/mu;
ecv=cross(V0,H)/mu-R0/r0;
e=norm(ecv);
ecth0=p/r0-1;
esth0=norm(cross(ecv,R0))/r0;
if abs(ecth0)>=eps;
th0=atan(esth0/ecth0);
if ecth0<0
    if esth0>=0;
    th0=th0+pi;
    end
elseif esth0<0
    th0=th0+2*pi;
end
elseif esth0>=0
    th0=pi/2;
else
    th0=3*pi/2;
end
ainv=-(v0*v0)/mu+2/r0;
a=1/ainv;
n=sqrt(mu/a^3);
E0=2*atan(sqrt((1-e)/(1+e))*tan(0.5*th0));
tau=t0+(-E0+e*sin(E0))/n;
M=n*(t-tau);
E=kepler(e,M);
r=a*(1-e*cos(E));
f=1+a*(cos(E-E0)-1)/r0;
g=a*alpha*(1-cos(E-E0))/mu+r0*sqrt(a/mu)*sin(E-E0);
fd=-sqrt(mu*a)*(sin(E-E0))/(r*r0);
gd=1+a*(cos(E-E0)-1)/r;
R=f*R0+g*V0;
V=fd*R0+gd*V0;
```

Table 6.9 Program *rotation.m* for generating the rotation matrix between perifocal and celestial frames using 3-1-3 Euler angles

```
function C=rotation(i,Om,w)
% Rotation matrix of 3-1-3 Euler angles (radians)
% i=orbital inclination;
% Om=right ascension of ascending node;
% w=argument of periapsis.
% (c) 2006 Ashish Tewari
L1=cos(Om)*cos(w)-sin(Om)*sin(w)*cos(i);
L2=-cos(Om)*sin(w)-sin(Om)*cos(w)*cos(i);
L3=sin(Om)*sin(i);
M1=sin(Om)*cos(w)+cos(Om)*sin(w)*cos(i);
M2=-sin(Om)*sin(w)+cos(Om)*cos(w)*cos(i);
M3=-cos(Om)*sin(i);
N1=sin(w)*sin(i);
N2=cos(w)*sin(i);
N3=cos(i);
C=[L1 L2 L3;M1 M2 M3;N1 N2 N3];
```

Table 6.10 Program *planechange.m* for generating the nonlinear state equations with a constant acceleration, switching controller for orbital plane control with an elliptical nominal orbit

```
% Program 'planechange.m' for generating the nonlinear state
% equations with a constant acceleration, switching controller
% for orbital plane control with an elliptical nominal
% orbit.
% (c) 2009 Ashish Tewari
function xdot=planechange(t,x)
dtr=pi/180;
mu=398600.4;

%Nominal orbit
rn =[6.572160714986236e+003
     7.178165264615914e+003
    -2.427939762419030e+002];
vn= [-1.820245870369515e+000
     4.039881732583030e+000
     2.304306050936373e+000];
hn=cross(rn,vn); %Nominal angular momentum

%Actual orbit:
R=x(1:3,1);
V=x(4:6,1);
r=norm(R);
v=norm(V);
H=cross(R,V);
h=norm(H);

%Angular error of orbit plane:
ithetan=cross(hn,rn)/(norm(hn)*norm(rn));
sinalfa=-dot(H/h,ithetan);

%Switching orbit plane control
alfa0=2*dtr; %Initial inclination error
u=0.001; %Constant input (km/s^2)
tf=abs(alfa0)*norm(hn)/(norm(rn)*u);
beta=-sign(sinalfa)*pi/2; %Thrust direction
if t>tf
    u=0;
end
U=u*(cos(beta)*V/v+sin(beta)*H/h);
R=x(1:3,1);
V=x(4:6,1);

% Equations of motion:
xdot(1:3,1)=V;
xdot(4:6,1)=-mu*R/r^3 + U;
```

achieved in both the cases with a zero inclination error (Figs. 6.12 and 6.13), Figs. 6.14 and 6.15 plotted for $\bar{u} = 10$ m/s^2 and $\bar{u} = 1$ m/s^2, respectively, show a rapid switching of the thrust direction between $\pm\pi/2$ is required near the end of the control interval, $t = t_f$. The switching takes place because the approximate value of t_f, derived with a linear approximation, is not precise in making $\alpha = 0$ at $t = t_f$, thereby causing a slight, initial overshoot of the $\alpha = 0$ condition. Since a deadband is not enforced in the present example, rapid switching due to imprecise timing is unavoidable. It is clear from Figs. 6.14 and 6.15 that more switchings occur for a smaller input magnitude.

Table 6.11 The postprocessing program, *postproplane.m*, of the plane change integration results with a constant acceleration, switching controller for orbital plane control with an elliptical nominal orbit

```
% (c) 2009 Ashish Tewari
dtr=pi/180;
rn =[6.572160714986236e+003
    7.178165264615914e+003
   -2.427939762419030e+002];
vn= [-1.820245870369515e+000
    4.039881732583030e+000
    2.304306050936373e+000];
hn=cross(rn,vn);
R=[];V=[];h=[];alfa=[];beta=[];
R=x(:,1:3);
V=x(:,4:6);
n=size(t,1);
alfa=[];h=[];
for i=1:n
r=norm(R(i,:));
v=norm(V(i,:));
H=cross(R(i,:),V(i,:))
h(i,:)=H;
ithetan=cross(hn,rn)/(norm(hn)*norm(rn));
sinalfa=-dot(H/norm(H),ithetan);
alfa(:,i)=asin(sinalfa);
beta(:,i)=-sign(sin(sinalfa))*pi/2;
end
```

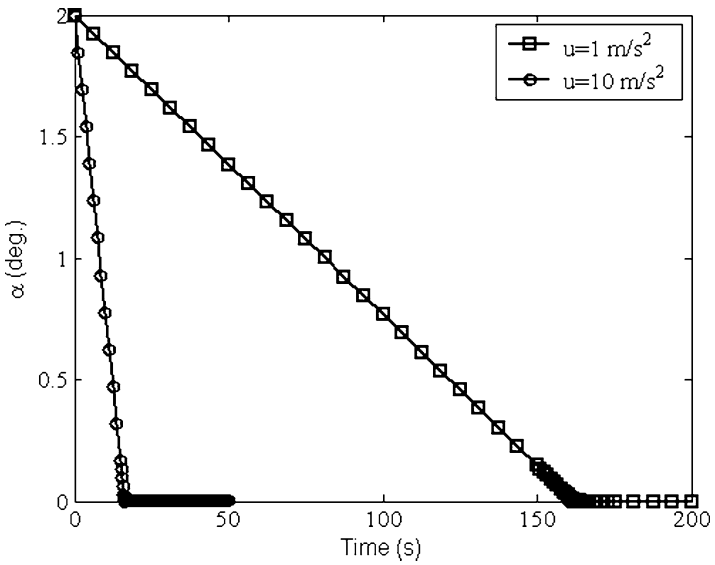


Fig. 6.12 Simulated angular displacement of orbital plane from the nominal with a constant acceleration, switching control system

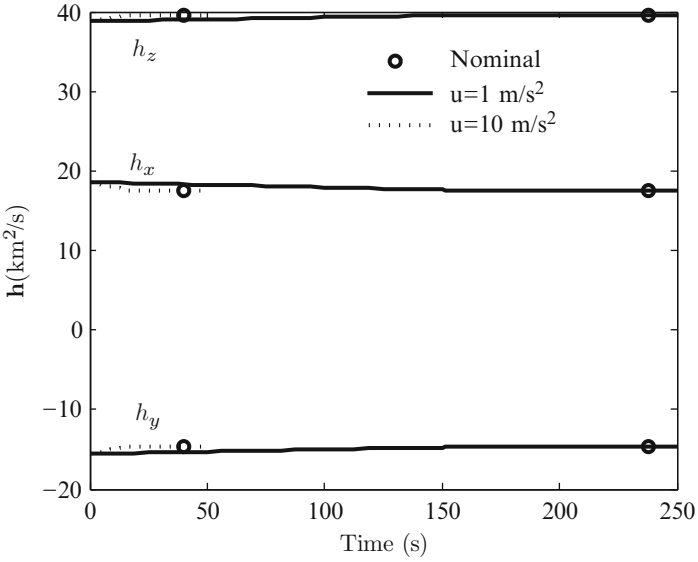


Fig. 6.13 Simulated angular momentum vector with a constant acceleration, switching, orbital plane control system

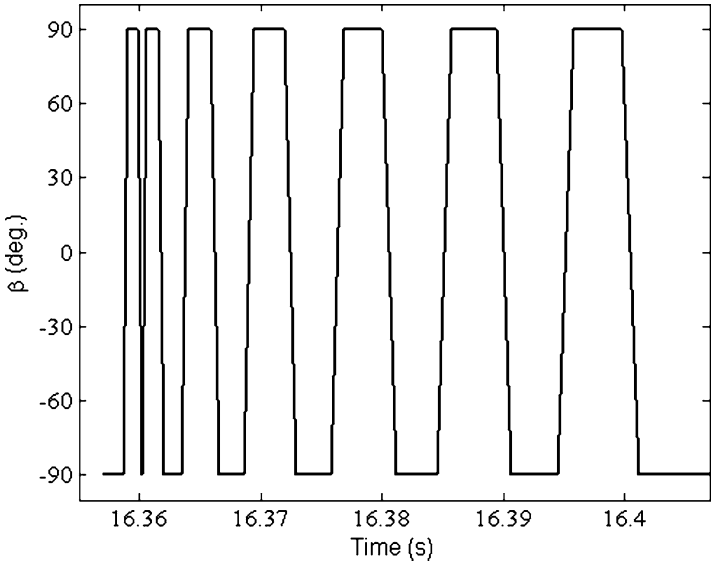


Fig. 6.14 Simulated thrust direction angle near $t = t_\mathrm{f}$ for the constant acceleration, switching, orbital plane control system with $\bar{u} = 10\ \mathrm{m/s}^2$

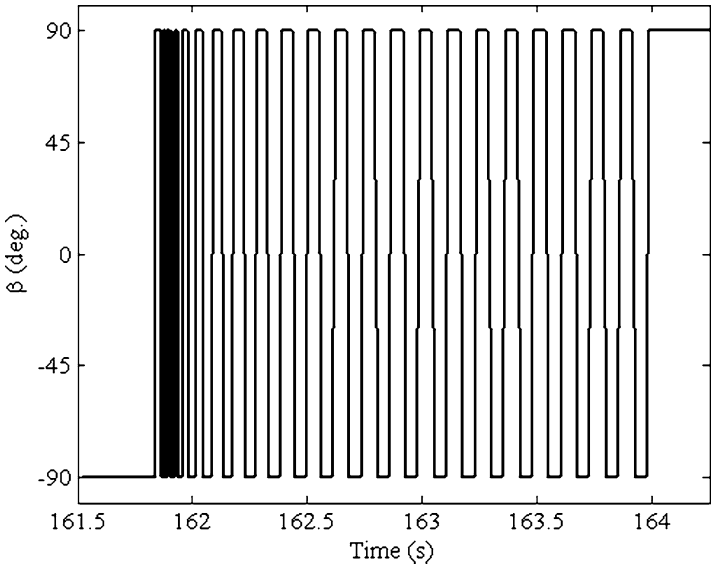


Fig. 6.15 Simulated thrust direction angle near $t = t_f$ for the constant acceleration, switching, orbital plane control system with $\bar{u} = 1$ m/s^2

6.4.2 Feedback Regulated Thrust, Orbital Plane Control

In order to remove the rapid thrust switching problem of the switching control and to make the control system robust with respect to disturbances, the thrust direction is kept fixed, while the acceleration input magnitude, $u(t)$, is regulated by a closed-loop system. Since the plant is of first-order, it is expected that a linear feedback regulator with a constant gain, k, where

$$u(t) = k\alpha(t), \tag{6.55}$$

and a fixed thrust direction,

$$\beta(t) = -\frac{\pi}{2},$$

should produce asymptotic stability. For the selection of the gain, k, consider the Laplace transform of the linearized plant, (6.49), along with the linear, feedback control law of (6.55), resulting in the following closed-loop, initial response:

$$\alpha(s) = \frac{\alpha_0}{s + k\frac{r_n}{h_n}}, \tag{6.56}$$

or,

$$\alpha(t) = \alpha_0 e^{-kr_n t/h_n}, \tag{6.57}$$

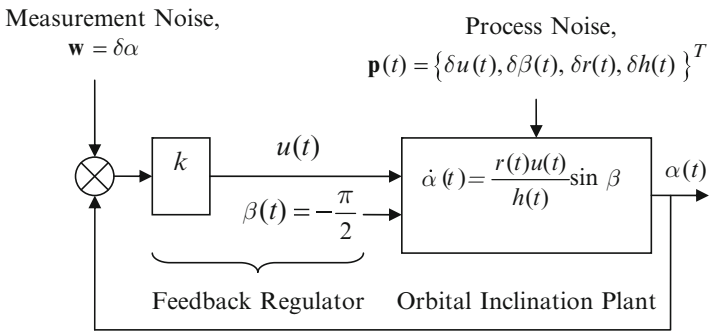


Fig. 6.16 A thrust regulated control system for orbital plane with a constant feedback gain, k

with a settling-time (Chap. 3),

$$t_s = 4\frac{h_n}{k r_n}. \tag{6.58}$$

Therefore, for a desired settling-time, t_s, the proportional feedback gain is the following:

$$k = 4\frac{h_n}{t_s r_n}. \tag{6.59}$$

When applied to the actual nonlinear plant, (6.48), the steady-state error can be reduced to zero asymptotically, provided the radius does not change markedly in the control interval $0 \leq t \leq t_s$. A schematic block diagram of the thrust regulated control system is shown in Fig. 6.16.

Example 6.5. Replace the switching control system designed in Example 6.4 with a thrust regulated controller for $t_s = 40$ s and the nominal and initial orbit given in Example 6.4.

We begin by replacing statements pertaining to switching control (lines 31–35) of the MATLAB code *planechange.m* (Table 6.11) by the following statements concerning the thrust regulated controller:

```
%Thrust regulated plane control
alfa=asin(sinalfa);
T=40;
k=4*norm(hn)/(T*norm(rn));
u=k*alfa; % km/s^2
beta=-pi/2;
```

Other aspects of the computation remain unchanged, and the results of the nonlinear simulation carried out for $2t_s$ are shown in Figs. 6.17–6.19. The desired nominal orbital plane is accurately achieved in an exponential manner (Figs. 6.17 and 6.18), while the thrust acceleration magnitude falls exponentially (as expected) from $u = 16.38$ m/s^2 at $t = 0$ to nearly zero at $t = 80$ s (Fig. 6.19).

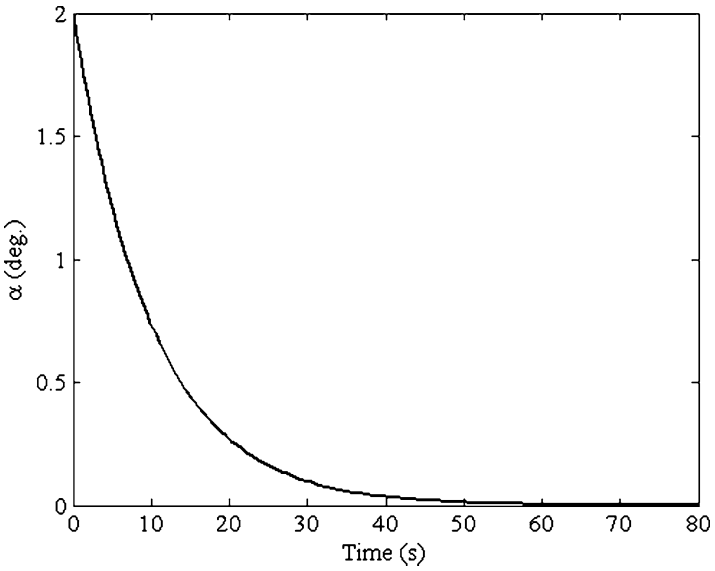


Fig. 6.17 Simulated angular displacement of orbital plane from the nominal with a constant gain regulated thrust acceleration magnitude

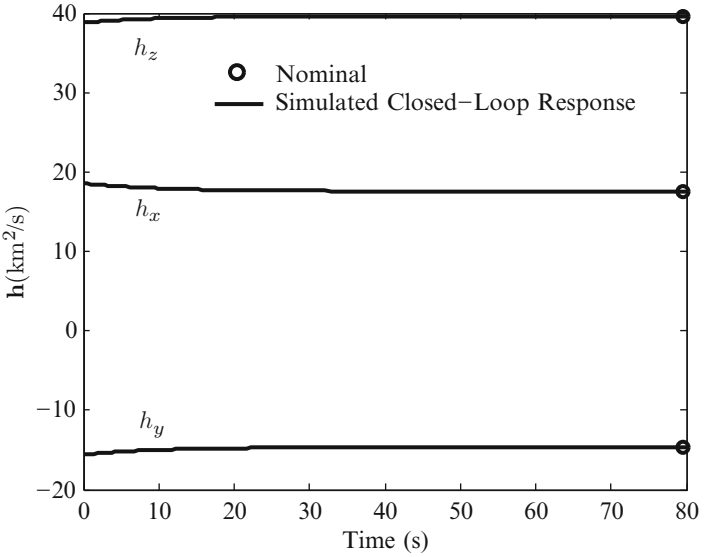


Fig. 6.18 Simulated angular momentum vector with a constant gain regulated thrust acceleration magnitude

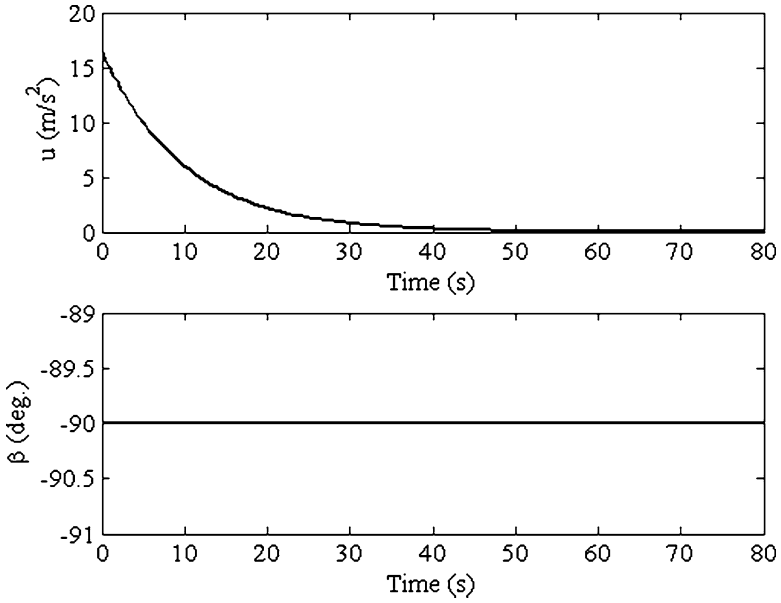


Fig. 6.19 Simulated thrust magnitude and thrust direction angle for orbital plane control system with a regulated thrust acceleration magnitude

6.5 Attitude Control by Torque Impulses

A complex multi-axis attitude maneuver can be designed as a series of single-axis rotations carried out by the firing of rocket thrusters that are arranged in equal and opposite pairs about each principal axis. Consider a rigid spacecraft with moment of inertia, J_z, equipped with a pair of attitude thrusters capable of exerting a large torque for an infinitesimal duration, $\Delta t \to 0$, about the axis of desired rotation, $\mathbf{k}$. The impulse causes an instantaneous change in the angular momentum by $\Delta H_z = \tau_z(0)$. Here, $\tau_z(0)$ is *not* the magnitude of the torque, but its time integral (total area of torque vs. time graph). Using the unit impulse function, $\delta(t)$ (Chap. 3), we say that a *torque impulse*, $\tau_z(t) = \tau_z(0)\delta(t)$, has been applied about $\mathbf{k}$. Thus, the rotational kinetics equations (Chap. 2) reduce to the following:

$$\begin{aligned} \dot{P} &= 0 \\ \dot{Q} &= 0 \\ J_z \dot{R} &= \tau_z(0)\delta(t), \end{aligned} \tag{6.60}$$

where $\boldsymbol{\omega} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$ is the angular velocity of the spacecraft relative to an inertial reference frame. In terms of the angular displacement, ψ, about $\mathbf{k}$, the last of (6.60) can be expressed as follows:

$$\ddot{\psi} = \frac{\tau_z(0)}{J_z}\delta(t), \tag{6.61}$$

whose solution is easily obtained by successive integration by Laplace transform (Chap. 3) as follows:

$$\begin{aligned} R(t) = \dot{\psi} &= R(0) + \frac{\tau_z(0)}{J_z}u_s(t) \\ \psi(t) &= \psi(0) + R(0)t + \frac{\tau_z(0)}{J_z}r(t), \end{aligned} \tag{6.62}$$

where $\psi(0)$, $R(0)$ refer to the initial condition immediately before the application of torque impulse, $u_s(t) = \int \delta(t)\mathrm{d}t$ is the unit step function (Chap. 3) applied at $t = 0$, and $r(t) = \int u_s(t)\mathrm{d}t$ is the *unit ramp function*, also applied at $t = 0$. We can write a unit ramp function applied at $t = T$ as follows:

$$r(t-T) = \begin{cases} 0, & t < T, \\ t - T, & t \geq T. \end{cases} \tag{6.63}$$

Since Δt is much smaller than the duration of the maneuver (and also because the torque magnitude by rocket thrusters briefly becomes very large), an impulsive thruster torque is commonly assumed. Equation (6.63) implies that the response to a single impulse is a linearly increasing displacement, but a step change in the speed. Therefore, if the maneuvering requirement is for a step change in the angular velocity, (a *spin maneuver*), then a single impulse is sufficient. However, if a given single-axis displacement is desired – called a *rest-to-rest maneuver* – one has to apply another impulse of opposite direction, $-\tau_z(0)\delta(t - T)$, in order to stop the rotation at time $t = T$, when the desired displacement has been reached. Since the governing differential equation, (6.61), is linear, its solution obeys the principle of linear superposition (Chap. 1), which allows a weighted addition of the responses to individual impulses, to yield the total displacement caused by multiple impulses. Hence, for two equal and opposite torque impulses spaced by an interval, T, with zero initial conditions, we have

$$\begin{aligned} \tau_z(t) &= \tau_z(0)[\delta(t) - \delta(t-T)] \\ R(t) &= \frac{\tau_z(0)}{J_z}[u_s(t) - u_s(t-T)] \\ \psi(t) &= \frac{\tau_z(0)}{J_z}[r(t) - r(t-T)]. \end{aligned} \tag{6.64}$$

Thus, the final angular velocity is zero, and a desired constant displacement, ψ_d, is reached at $t = T$. The magnitude of ψ_d can be controlled by varying the time T at which the second impulse is applied. The application of two equal and opposite

impulses of the maximum possible magnitude for achieving a desired displacement in the shortest possible time is called *bang-bang* control. We have seen such a control application in Example 3.5. A detailed discussion of open-loop, multi-axis attitude control of nonspinning as well as spin stabilized spacecraft by attitude thrusters can be found in [21].

6.6 Attitude Control of Spacecraft by Rotors

Frequent rocket firings (torque impulses) for attitude control can quickly deplete the propellant, thereby drastically curtailing the spacecraft life. A more efficient alternative is to use small rotors driven by solar arrays for applying the corrective control torques. Such internally generated torques can manage a spacecraft's angular momentum at all times without requiring human intervention. Attitude control rotors can be of two types: (a) with axis fixed relative to the spacecraft (reaction and momentum wheels), and (b) with a variable axis relative to the spacecraft (control moment gyroscopes).

6.6.1 Rotors with Fixed Axes

Consider three rotors, each spinning about a principal axis of the spacecraft (Fig. 6.20). The respective angular momenta of the rotors relative to the spacecraft are given by (h_x, h_y, h_z), with the following net relative angular momentum vector:

$$\mathbf{h} = h_x\mathbf{i} + h_y\mathbf{j} + h_z\mathbf{k}, \tag{6.65}$$

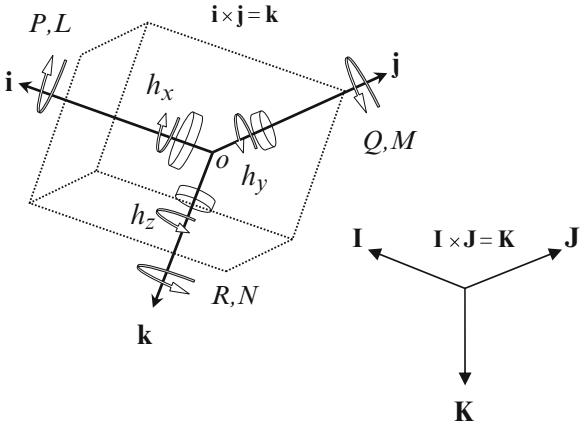


Fig. 6.20 Spacecraft with fixed axis rotors spinning parallel to the spacecraft's principal axes

where $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ is the principal body frame of the spacecraft. The total angular momentum of the vehicle is then given by

$$\mathbf{H} = \mathsf{J}\boldsymbol{\omega} + \mathbf{h} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} + \begin{Bmatrix} h_x \\ h_y \\ h_z \end{Bmatrix}, \tag{6.66}$$

where $\boldsymbol{\omega} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$ is the angular velocity of the spacecraft relative to an inertial reference frame and (J_x, J_y, J_z) are the principal moments of inertia of the spacecraft. If the following disturbing torque is applied to the spacecraft,

$$\boldsymbol{\tau} = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}, \tag{6.67}$$

then we have the following rotational kinetics equations (Chap. 2):

$$\begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = \mathsf{J} \begin{Bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{Bmatrix} + \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} \times \left(\mathsf{J} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} + \begin{Bmatrix} h_x \\ h_y \\ h_z \end{Bmatrix} \right) + \begin{Bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{h}_z \end{Bmatrix}, \tag{6.68}$$

where overdot represents time derivative. Written in the scalar form, (6.68) has the following components called the *roll*, *pitch*, and *yaw* equations, respectively:

$$L = J_x\dot{P} + (J_z - J_y)QR + Qh_z - Rh_y + \dot{h}_x, \tag{6.69}$$

$$M = J_y\dot{Q} + (J_x - J_z)PR + Rh_x - Ph_z + \dot{h}_y, \tag{6.70}$$

and

$$N = J_z\dot{R} + (J_y - J_x)PQ + Ph_y - Qh_x + \dot{h}_z. \tag{6.71}$$

6.6.1.1 Momentum Wheels

It is clear from (6.69)–(6.71) that the mere presence of a constant rotor angular momentum is capable of modifying the spacecraft's attitude dynamics. Such a rotor, spinning at a constant angular velocity with respect to the spacecraft is called a *momentum wheel*. We can have equilibrium solutions of single-axis, constant speed spacecraft rotation with a momentum wheel about the principal axis of rotation. For example, a spacecraft with a pitch momentum wheel ($h_y = \text{const.}$) can be rotating at a constant pitch rate, Q, but must have $P = R = 0$ and $h_x = h_z = 0$ for equilibrium ($L = M = N = 0$). It is also evident from these equations that a momentum wheel provides a coupling between rotations about the other two principal axes. Such a coupling is called *biasing* and is very useful, because it increases the controllability of the motion about all the principal axes.

Example 6.6. Consider a satellite with $J_x = 1{,}000\ \text{kg m}^2$, $J_y = 2{,}000\ \text{kg m}^2$, and $J_z = 3{,}000\ \text{kg m}^2$, equipped with a pitch momentum wheel of angular momentum, $h_y = 100\ \text{N m s}$. The satellite must maintain a Nadir pointing attitude in a circular orbit of orbital frequency, $n = \sqrt{\mu/a^3} = 0.0011$ rad/s (Chap. 2), such that the axis **k** always points vertically downward while the axis **i** is along the flight direction. Thus, the spacecraft must maintain an equilibrium angular velocity of

$$\boldsymbol{\omega} = -n\mathbf{j}. \tag{6.72}$$

The control torque vector is given here by $\mathbf{u} = (u_x, u_y, u_z)^{\mathrm{T}}$. Assuming small rate perturbations, (P, q, R), from equilibrium, examine the stability of the attitude dynamics about the given equilibrium. Is the system controllable if both a rolling moment, $u_x(t)$, and a pitching moment, $u_y(t)$, are applied together? Is it controllable with respect to a control torque, $u_y(t)$, about the pitch axis alone?

We express the net pitch-rate as $Q = -n + q$, and substitute it into (6.69)–(6.71), resulting in the following linearized dynamics:

$$\begin{aligned}
\dot{P} &= \frac{1}{J_x}\left[n(J_z - J_y) + h_y\right] R + u_x/J_x \\
\dot{q} &= u_y/J_y \\
\dot{R} &= \frac{1}{J_z}\left[n(J_y - J_x) - h_y\right] P + u_z/J_z.
\end{aligned} \tag{6.73}$$

Clearly, the decoupled pitch dynamics is not asymptotically stable, and the coupled roll-yaw motion has the following dynamics matrix:

$$\mathsf{A} = \begin{bmatrix} 0 & \dfrac{n(J_z - J_y) + h_y}{J_x} \\ \dfrac{n(J_y - J_x) - h_y}{J_z} & 0 \end{bmatrix},$$

resulting in the characteristic equation

$$|s\mathsf{I} - \mathsf{A}| = s^2 - \frac{[n(J_z - J_y) + h_y][n(J_y - J_x) - h_y]}{J_x J_z} = 0.$$

Putting the numerical values, we have

$$s^2 + 0.0033 = 0,$$

or $s_{1,2} = \pm 0.0577\mathrm{i}$ rad/s. Thus, the roll-yaw dynamics is stable but not asymptotically stable. For controllability with respect to a rolling and a pitching moment, we write

$$\mathsf{A} = \begin{pmatrix} 0 & 0 & a_1 \\ 0 & 0 & 0 \\ a_2 & 0 & 0 \end{pmatrix}; \qquad \mathsf{B} = \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{pmatrix},$$

where

$$a_1 = \frac{n(J_z - J_y) + h_y}{J_x}; \;\; a_2 = \frac{n(J_y - J_x) - h_y}{J_z}; \;\; b_1 = 1/J_x; \;\; b_2 = 1/J_y.$$

The controllability test matrix (Chap. 1) is the following:

$$\mathsf{P} = \left(\mathsf{B}, \mathsf{AB}, \mathsf{A}^2\mathsf{B}\right) = \begin{pmatrix} b_1 & 0 & 0 & 0 & a_1 a_2 b_1 & 0 \\ 0 & b_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 b_1 & 0 & 0 & 0 \end{pmatrix},$$

which is clearly of Rank 3 for $a_1 \neq 0$ and $a_2 \neq 0$. Hence, the vehicle is controllable by roll and pitch inputs. Even in an axisymmetric spacecraft where two moments of inertia are identical, the presence of the momentum wheel prevents a_1 and a_2 from vanishing, thereby maintaining controllability. However, we do not expect the vehicle to be controllable with respect to a pitch control moment (verify), since roll-yaw dynamics is decoupled from the pitching motion.

6.6.1.2 Reaction Wheels

A rotor spinning about axis fixed relative to the spacecraft can change its angular momentum by varying the spin rate. Such a rotor is called a *reaction wheel*, because it exerts a control torque on the spacecraft equal and opposite to the rate of change of its angular momentum. Since the spacecraft's rate of rotation is thousands of times slower than the spin speed of a reaction wheel, it is reasonable to assume a step change of angular momentum applied by a reaction wheel. For example, a yaw reaction wheel can create a torque of magnitude

$$\tau_z(t) = \dot{h}_z u_s(t), \tag{6.74}$$

about **k** by changing its speed in an infinitesimal duration relative to the spacecraft. Due to its small size, the torque exerted by a reaction wheel is typically much smaller in magnitude than that of a rocket thruster. When a maneuver requires a continuous variation of the torque, it can be produced by a smooth variation in the speed of the reaction wheel.

By having reaction wheels about two principal axes and also a momentum wheel about any one of the two axes, we can render a spacecraft fully controllable about all axes. This is seen in the following example.

Example 6.7. Re-consider the spacecraft of Example 6.6, which is now equipped with roll and pitch reaction wheels, in addition to the pitch momentum wheel. Design a control system for damping out rate perturbations by using feedback from pitch and yaw rate gyros. The reaction wheel torques should not exceed $|\dot{h}_x| \leq 0.04$ N m and $|\dot{h}_y| \leq 0.05$ N m for an initial error of $q(0) = -0.1n$ and $R(0) = 0.1n$, while achieving a settling-time of about 100 s. Also, include first-order reaction wheel servomotors of time constant 1 s in the simulated closed-loop response.

We carry out the design by linear optimal control (LQR) for both regulator and observer. The design steps are given by the following MATLAB statements:

```
>> Jx=1000; Jy=2000; Jz=3000; hy=100; n=0.0011; %Satellite plant parameters
>> a1=(n*(Jz-Jy)+hy)/Jx; a2=(n*(Jy-Jx)-hy)/Jz;
>> A=[0 0 a1;0 0 0;a2 0 0],B=[1/Jx 0;0 1/Jy;0 0],C=[0 1 0;0 0 1];
       %State-space model

A =
            0            0      0.10113
            0            0            0
    -0.032958            0            0

B =
        0.001            0
            0       0.0005
            0            0

>> [k,S,E]=lqr(A,B,100*eye(3),[1/Jx 0;0 1/Jy]) %Regulator design by LQR

k =
       339.44  2.6366e-014      -230.88        %Regulator gain matrix
  2.6366e-014       447.21 -2.4247e-013

S =
       339.44  2.6366e-014      -230.88      %Algebraic Riccati solution
  2.6366e-014       447.21 -2.4247e-013
      -230.88 -2.4247e-013       3,419.3

E =                                          %Regulated system eigenvalues
     -0.30337
    -0.036069
     -0.22361

>> Lp=lqr(A',C',1000*eye(3),eye(2));L=Lp'  %Observer design by LQR

L =                                          %Observer gain matrix
  2.0865e-015      -31.522
       31.623  4.3637e-015
  4.3637e-015       31.656

% Augmented plant with servomotors of time-constant 1 s:
>> Abar=[A B;zeros(1,3) -1 0;zeros(1,4) -1]; Bbar=[0 0 0 1 0;0 0 0 0 1]';
>> Cbar=[C zeros(2,2)];

%Closed-loop system (plant+regulator+observer+servos):
>> Ac=[Abar  -Bbar*k;L*Cbar A-L*C-B*k]; Bc=[Bbar*k; B*k];
>> sys=ss(Ac,Bc,[eye(5) zeros(5,3)],zeros(5,3))

a =
                x1           x2           x3           x4           x5
   x1            0            0       0.1011        0.001            0
   x2            0            0            0            0       0.0005
   x3     -0.03296            0            0            0            0
```

```
    x4            0            0            0           -1            0
    x5            0            0            0            0           -1
    x6            0   2.087e-015       -31.52            0            0
    x7            0        31.62   4.364e-015            0            0
    x8            0   4.364e-015        31.66            0            0

                x6           x7           x8
                 0            0            0
                 0            0            0
                 0            0            0
            -339.4  -2.637e-014        230.9
       -2.637e-014       -447.2   2.425e-013
           -0.3394  -2.113e-015        31.85
       -1.318e-017       -31.85  -4.242e-015
          -0.03296  -4.364e-015       -31.66

b =
                u1           u2           u3
    x1           0            0            0
    x2           0            0            0
    x3           0            0            0
    x4       339.4   2.637e-014       -230.9
    x5   2.637e-014        447.2  -2.425e-013
    x6      0.3394   2.637e-017      -0.2309
    x7   1.318e-017       0.2236  -1.212e-016
    x8           0            0            0

c =
        x1  x2  x3  x4  x5  x6  x7  x8
    y1   1   0   0   0   0   0   0   0
    y2   0   1   0   0   0   0   0   0
    y3   0   0   1   0   0   0   0   0
    y4   0   0   0   1   0   0   0   0
    y5   0   0   0   0   1   0   0   0

d =
        u1  u2  u3
    y1   0   0   0
    y2   0   0   0
    y3   0   0   0
    y4   0   0   0
    y5   0   0   0

Continuous-time model.

>> [y,t,x]=initial(sys,[0 -0.1*n 0.1*n 0 0 0 0 0]',150);
    %Closed-loop initial response

%Closed-loop frequency response from pitch reaction wheel to yaw rate gyro:
>> sys1=ss(A-B*k,B(:,2),C(2,:),0)

a =
                x1           x2           x3
    x1      -0.3394  -2.637e-017        0.332
    x2  -1.318e-017      -0.2236   1.212e-016
    x3     -0.03296            0            0

b =
           u1
    x1      0
    x2  0.0005
    x3      0

c =
        x1  x2  x3
    y1   0   0   1
```

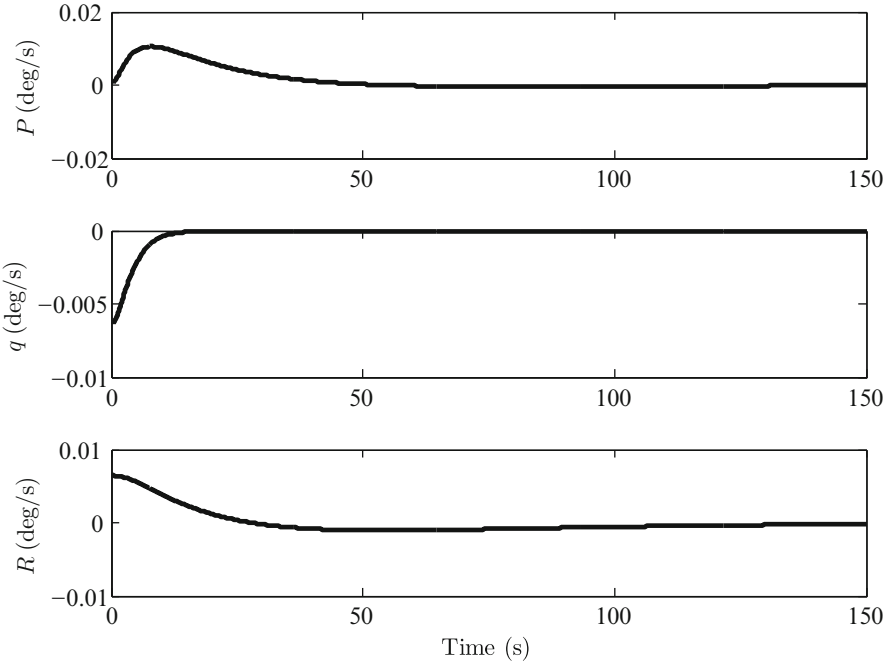


Fig. 6.21 Closed-loop initial response of a spacecraft equipped with roll and pitch reaction wheels and a pitch momentum wheel to an initial pitch and yaw rate perturbation

```
d =
       u1
   y1   0

Continuous-time model.
>> margin(sys1)
```

The simulated initial response is plotted in Figs. 6.21 and 6.22, while the closed-loop frequency response from pitch reaction wheel to yaw rate gyro is plotted in Fig. 6.23. All of the perturbations are seen to settle to zero in about 100 s, with the maximum reaction wheel torques limited to 0.04 N m for both the wheels. The system is also quite robust with respect to modeling uncertainties, as revealed by a gain margin of 418 dB and an infinite phase margin (Fig. 6.23).

6.6.2 Rotors with Variable Axes

A rotor can produce a control torque normal to the axis of rotation merely by deflecting the spin axis (to be contrasted with reaction wheels that can only create moments along the spin axis). When the spin speed of the rotor is constant, it

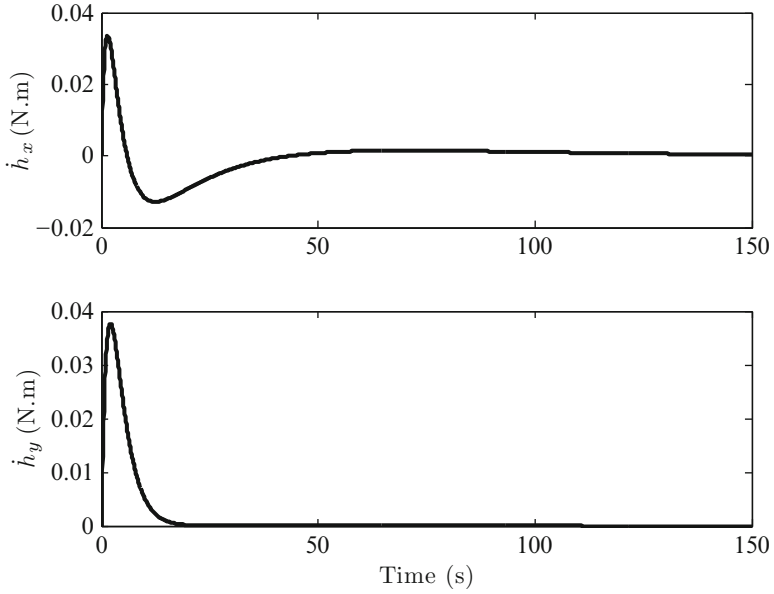


Fig. 6.22 Closed-loop reaction wheel torques of a spacecraft equipped with roll and pitch reaction wheels and a pitch momentum wheel to an initial pitch and yaw rate perturbation

is called a *control moment gyroscope* (CMG). However, if the spin speed can be changed, we have a *variable speed control moment gyroscope* (VSCMG), which can be regarded as a combination of a reaction wheel and a CMG. The advantage of a CMG (or a VSCMG) lies in the fact that a single rotor can be used to apply a multi-axis control torque. However, both CMG and VSCMG systems are inherently nonlinear [21], and thus beyond our present scope of controller design. Here, we will consider a simple but practical approach of controlling a spacecraft through a CMG rotor.

Consider a CMG that can tilt its spin axis about a gimbal axis, $\mathbf{n}$, which is inclined at a constant angle, σ, with respect to the spacecraft principal axis, $\mathbf{k}$, as shown in Fig. 6.24. If the angle $\alpha(t)$ is the deflection angle of the rotor axis measured from the spacecraft axis, $\mathbf{i}$ (Fig. 6.24), then the orientation of the gimbal axis is given by the Euler angle sequence, $(\sigma)_1, (\alpha)_3$ (Chap. 2) as follows:

$$\mathbf{n} = -\sin\sigma\mathbf{j} + \cos\sigma\mathbf{k}. \tag{6.75}$$

The CMG can be gimbaled at a rate, $\dot{\alpha}$, relative to the spacecraft, thereby generating a control torque given by

$$\boldsymbol{\tau} = \dot{\alpha}\mathbf{n} \times h(\mathbf{i}\cos\alpha + \mathbf{j}'\sin\alpha), \tag{6.76}$$

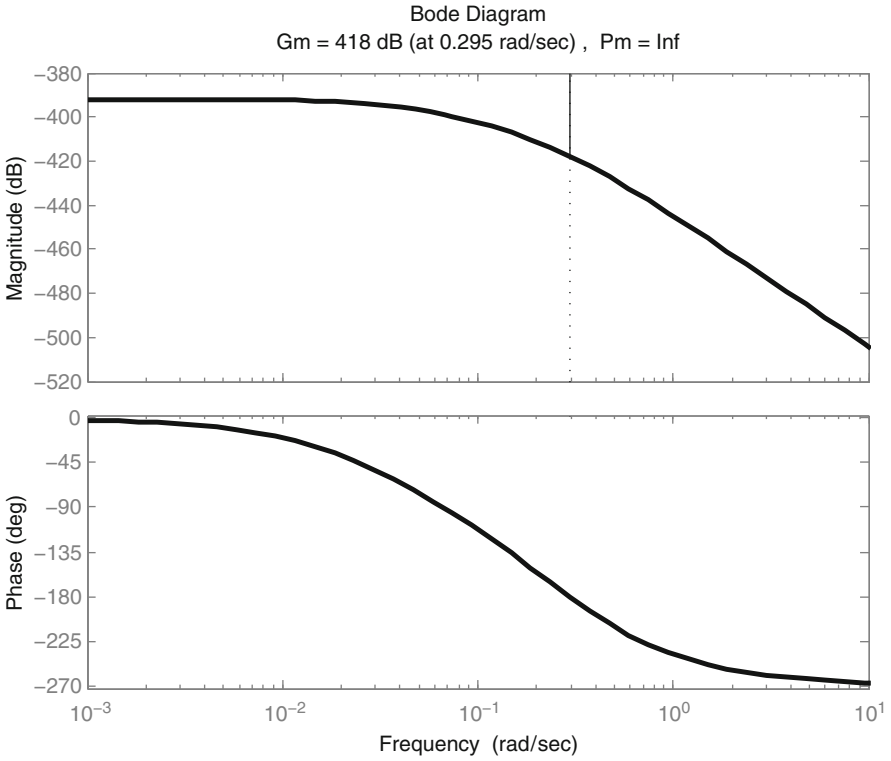


Fig. 6.23 Closed-loop frequency response from pitch reaction wheel to yaw rate gyro

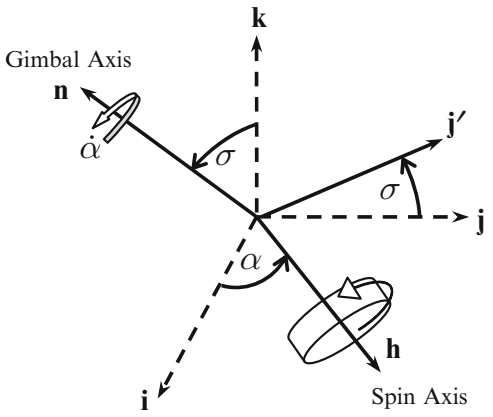


Fig. 6.24 A control moment gyroscope (CMG) with spin angular momentum, **h**, and gimbal axis, **n**, inclined at a constant angle, σ, with respect to the spacecraft's body axis, **k**

where

$$\mathbf{j}' = \cos\sigma\mathbf{j} + \sin\sigma\mathbf{k}. \tag{6.77}$$

The spin speed (hence h) of the CMG is usually chosen to be large enough in order to require only small deflections for a given control torque. Therefore, with the small α assumption, we have

$$\boldsymbol{\tau} \simeq \dot{\alpha}\mathbf{n} \times h(\mathbf{i} + \alpha\cos\sigma\mathbf{j} + \alpha\sin\sigma\mathbf{k}) = h\dot{\alpha}\,(-\alpha\mathbf{i} + \cos\sigma\mathbf{j} + \sin\sigma\mathbf{k})\,. \tag{6.78}$$

Hence, a single CMG is capable of affecting rotations about all spacecraft axes, but in a nonlinear fashion. By appropriately choosing σ, one can select the desired control effectiveness of the CMG rotor about the various axes. The gimbal speed, $\dot{\alpha}$, and the deflection angle, α, are the variables of the CMG actuator. A desired magnitude of roll, pitch, and yaw control moments can thus be generated by the following nonlinear relationships:

$$\begin{aligned} \dot{h}_x &= -h\alpha\dot{\alpha} \\ \dot{h}_y &= h\dot{\alpha}\cos\sigma \\ \dot{h}_z &= h\dot{\alpha}\sin\sigma. \end{aligned} \tag{6.79}$$

For a linear servomotor driving the CMG gimbal, the gimbal rate can be treated as a step input, $\dot{\alpha} = u\ u_s(t)$, which results in a ramp deflection $\alpha(t) = ut\ u_s(t)$. Since the gimbal rate is an order of magnitude larger than the spacecraft's angular velocity, such an approximation is usually valid. However, one still has to deal with the nonlinear actuator system.

Example 6.8. Let us revisit Example 6.7 and replace the roll and pitch reaction wheels by a single CMG of constant angular momentum magnitude $h = 10\ \mathrm{N\,m\,s}$, and a gimbal axis inclination, $\sigma = 30°$. With the axis deflection, $\alpha(t)$, as the sole control input, the control torque vector is given by

$$\boldsymbol{\tau} = h\dot{\alpha}\begin{pmatrix} -\alpha \\ \cos\sigma \\ \sin\sigma \end{pmatrix}.$$

The state equation of the nonlinear plant is given by

$$\dot{\mathbf{x}} = \mathsf{A}\mathbf{x} + \boldsymbol{\tau},$$

where the dynamics matrix, A, is the same as before, and $\mathbf{x} = (P, q, R)^{\mathrm{T}}$. With pitch and yaw rate gyro feedback, we have a single input, two-output, nonlinear, time-varying plant. Consider the following simple, linear feedback control law in order to stabilize the spacecraft:

$$\dot{\alpha} = -k_1 q - k_2 R,$$

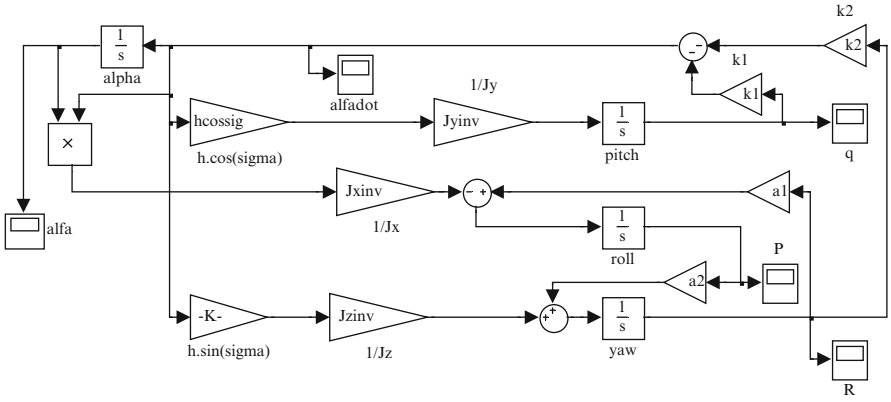


Fig. 6.25 Simulink block diagram for simulating spacecraft attitude dynamics with a CMG rotor in closed-loop with pitch and yaw rate gyros

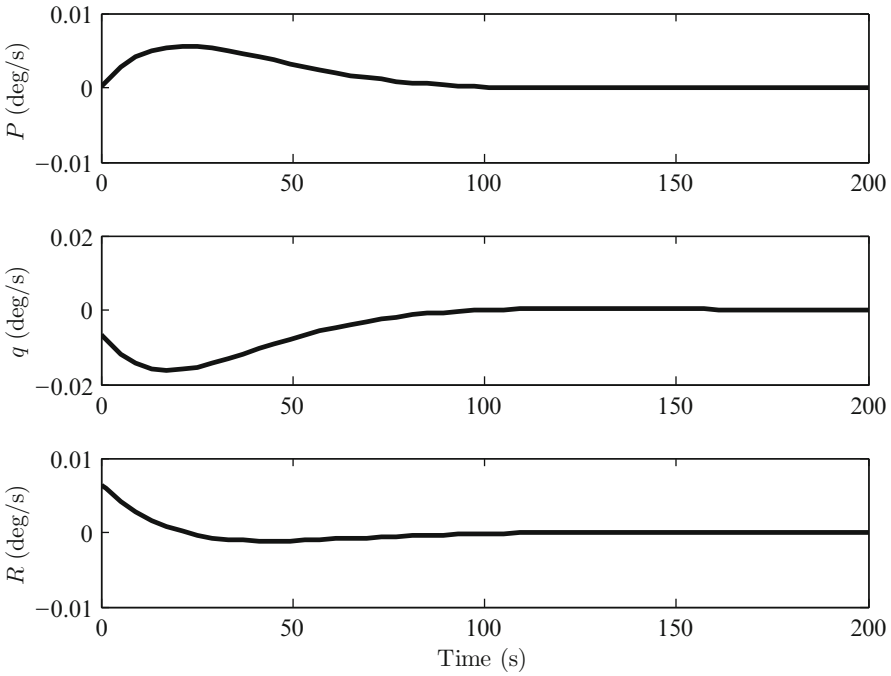


Fig. 6.26 Closed-loop initial response of a spacecraft equipped with a CMG rotor with a linear pitch and yaw rate feedback control

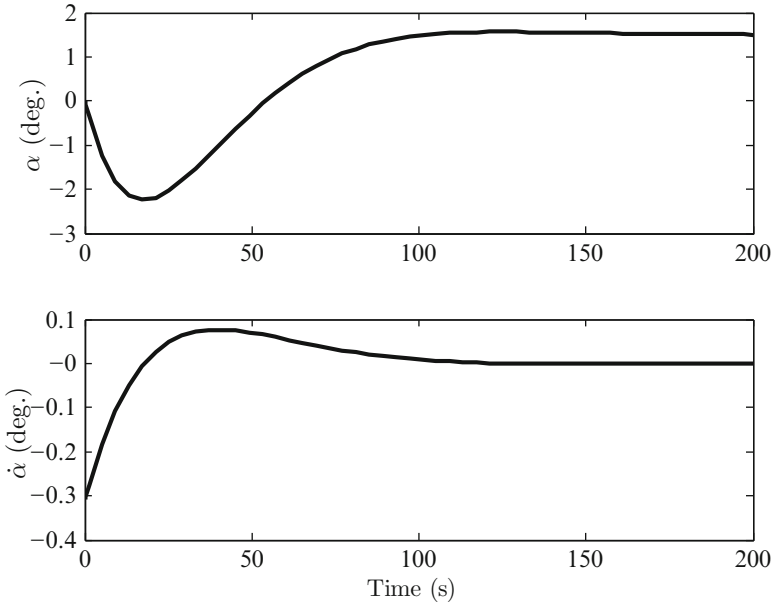


Fig. 6.27 Closed-loop CMG gimbal deflection and rate response of a spacecraft equipped with a CMG rotor with a linear pitch and yaw rate feedback control

where $k_1 > 0, k_2 > 0$ are controller constants. Thus, the gimbal rate is made proportional to a linear sum of the two rate errors, which is quite easy to implement. The controller design specifications are the same as in the previous example for the two reaction wheels, with a new requirement that the gimbal deflection must remain below $\pm 5^\circ$ for the given initial condition. By careful simulation, the controller constants are selected to be $k_1 = 2$ and $k_2 = 50$. A Simulink block-diagram shown in Fig. 6.25 is employed for the simulation of the closed-loop system, and the results are plotted in Figs. 6.26 and 6.27. Note the quick subsidence of all the spacecraft body rates to zero in about 100 s, without any oscillatory behavior. The gimbal deflection is within specified limits and the gimbal rate is limited to $\pm 0.3^\circ$/s. Therefore, a simple linear rate feedback scheme has been found to be adequate for controlling the spacecraft with a nonlinear CMG rotor.

6.7 Attitude Control of Spacecraft Under Solar Radiation Torque

Since most spacecraft are powered by large solar arrays facing the Sun, a small but constant solar radiation pressure torque acts on the spacecraft. If uncontrolled, the spacecraft will move away from the desired orientation under the integral effect of

solar radiation over a period of time, thereby rendering its mission useless. In such a case, attitude control consists of restoring the original equilibrium orientation of the spacecraft by canceling the effect of the disturbance. Rigid spacecraft rotation about a single principal axis is represented by the following second-order transfer function relating the angular displacement output, $\psi(s)$, and an external torque input, $\tau(s)$:

$$\frac{\psi(s)}{\tau(s)} = \frac{1}{Js^2}, \tag{6.80}$$

where J is the moment of inertia about the concerned axis. Let the disturbing torque,

$$\tau_D(s) = \frac{\tau_0}{s}, \tag{6.81}$$

be modeled as a step input of small, constant magnitude, τ_0, that starts acting at $t = 0$ when the spacecraft is at the desired orientation ($\psi(0) = \dot{\psi}(0) = 0$). Since the transfer function has a double pole at $s = 0$, the spacecraft is unstable with respect to external disturbances. Therefore, a feedback control system is necessary for returning the spacecraft to the desired orientation in the presence of the disturbance.

A common stabilization system is of proportional-derivative (PD) type (Chap. 3), based upon a reaction wheel torque input and angular displacement output, $\psi(t)$, that is obtained either by a rate-integrating gyro (Chap. 2), or a Sun/star scanner [23]. The control torque produced by the reaction wheel is thus given by

$$\tau_C(s) = \dot{h} = -(k_1 + k_2 s)\psi(s), \tag{6.82}$$

where $\dot{h}$ is the reaction wheel torque and $k_1 > 0, k_2 > 0$ are the PD controller constants. On substituting (6.82) into (6.80) with $\tau = \tau_D = \tau_0/s$, we have

$$\psi(s) = \frac{\tau_0/J}{s(s^2 + 2\zeta\omega s + \omega^2)}, \tag{6.83}$$

where

$$\omega = \sqrt{\frac{k_1}{J}} \tag{6.84}$$

and

$$\zeta = \frac{k_2}{2\sqrt{k_1 J}}. \tag{6.85}$$

Equation (6.83) is expanded partial fractions as follows:

$$\psi(s) = \frac{\tau_0}{J\omega^2}\left(\frac{1}{s} - \frac{s + 2\zeta\omega}{s^2 + 2\zeta\omega s + \omega^2}\right), \tag{6.86}$$

which by taking the inverse Laplace transform subject to zero initial conditions produces the following closed-loop angular displacement:

$$\psi(t) = \frac{\tau_0}{k_1}\left[1 - \mathrm{e}^{-\zeta\omega t}\left\{\cos(\omega_\mathrm{d} t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_\mathrm{d} t)\right\}\right] u_\mathrm{s}(t), \tag{6.87}$$

where $\omega_\mathrm{d} = \omega\sqrt{1-\zeta^2}$ is the damped natural frequency and $u_\mathrm{s}(t)$ the unit step function (Chap. 3). Clearly, the steady-state angular error is given by

$$\psi(\infty) = \lim_{s\to 0} s\psi(s) = \frac{\tau_0}{k_1}, \tag{6.88}$$

which can be minimized by selecting k_1 as large as physically possible.

6.8 Attitude Control of Spacecraft Under Gravity-Gradient Torque

All nonspherical objects experience a torque due to gravity, but such a torque is negligible on the ground compared with other environmental effects. However, the gravity-gradient torque can be significant for an elongated spacecraft in a low orbit, next only to aerodynamic drag. If not properly compensated for, gravity-gradient torque can overwhelm a large spacecraft, causing it to tumble, or to achieve an abnormal attitude. The *Skylab* mission of NASA came to an abrupt end in 1977 when the gravity-gradient torque turned the spacecraft in a way that caused its orbit to decay rapidly due to excessive drag, ultimately making a destructive re-entry in 1979. Thus, it is important to analyze the effects of gravity-gradient on low-orbiting satellites.

Consider a spacecraft in a circular orbit of frequency, n, with the principal axes in a Nadir pointing equilibrium attitude, $(\mathbf{i}_\mathbf{e}, \mathbf{j}_\mathbf{e}, \mathbf{k}_\mathbf{e})$ (Fig. 6.28). After a perturbation from the equilibrium, the spacecraft assumes a disturbed attitude, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, represented by the Euler angles, $(\psi)_3, (\theta)_2, (\phi)_1$ (Chap. 2), termed roll, pitch, and yaw, respectively, with respect to the equilibrium orientation. The gravity-gradient torque experienced in such a situation can be expressed as follows [21]:

$$\boldsymbol{\tau}_\mathbf{g} = \int \boldsymbol{\rho} \times \mathbf{g}\mathrm{d}m = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}, \tag{6.89}$$

where $\boldsymbol{\rho}$ locates an elemental mass from the center of mass, o, $\mathbf{g}$ is the acceleration due to gravity experienced at that location, and

$$L = \frac{3}{2}n^2(J_z - J_y)\cos^2\theta\sin(2\phi)$$

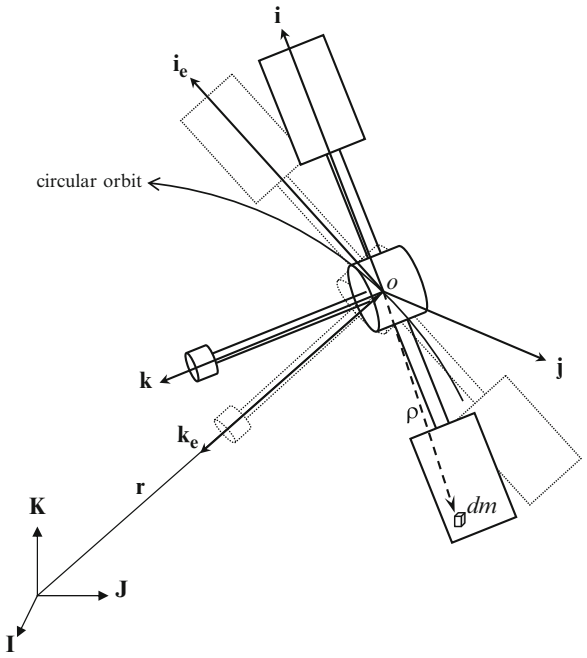


Fig. 6.28 Gravity-gradient satellite in a circular orbit

$$M = \frac{3}{2}n^2(J_z - J_x)\cos\phi\sin(2\theta)$$
$$N = \frac{3}{2}n^2(J_x - J_y)\sin\phi\sin(2\theta), \tag{6.90}$$

with J_x, J_y, J_z being the principal moments of inertia of the spacecraft. For small perturbations, ϕ, θ, ψ, the gravity-gradient torque components become linearized,

$$L \simeq 3n^2(J_z - J_y)\phi$$
$$M \simeq 3n^2(J_z - J_x)\theta$$
$$N \simeq 0, \tag{6.91}$$

thereby enabling a stability and control analysis. When the gravity-gradient torque is substituted into the linearized rotational kinetics equations, (6.69)–(6.71), we have

$$3n^2(J_z - J_y)\phi = J_x\dot{P} - (J_z - J_y)nR + (-n + q)h_z - Rh_y + \dot{h}_x$$
$$3n^2(J_z - J_x)\theta = J_y\dot{q} + Rh_x - Ph_z + \dot{h}_y$$

$$0 = J_z\dot{R} - (J_y - J_x)nP + Ph_y - (-n + q)h_x + \dot{h}_z. \tag{6.92}$$

Furthermore, by using rotational kinematics equations for the 3-2-1 Euler angles (Chap. 4) and assuming small angles and rates about the $\boldsymbol{\omega}_{\mathbf{e}} = -n\mathbf{j}$ equilibrium, we have the following kinematical relationships:

$$\left\{ \begin{array}{c} P \\ q \\ R \end{array} \right\} \simeq \left\{ \begin{array}{c} \dot{\phi} - n\psi \\ \dot{\theta} \\ \dot{\psi} + n\phi \end{array} \right\}. \tag{6.93}$$

It is evident from (6.92) and (6.93) that an absence of roll and yaw momentum wheels ($h_x = h_z = 0$) causes a decoupling of the motion into pitch (θ) and roll-yaw (ϕ, ψ) dynamics. The stability of the resulting equilibrium requires that the axis pointing toward (or away) from the planetary center should be the minor axis, while the major axis should be normal to the orbital plane [21]. Thus, a stable gravity-gradient orientation is possible only if we have $J_y > J_x > J_z$ for the spacecraft shown in Fig. 6.28.

Example 6.9. Consider a spacecraft in a 90-min circular orbit around the Earth, having the following principal moments of inertia in a Nadir pointing equilibrium attitude (Fig. 6.28):

$$J_x = 2 \times 10^5 \text{ kg m}^2; \quad J_y = 6 \times 10^4 \text{ kg m}^2$$

$$J_z = 1 \times 10^5 \text{ kg m}^2.$$

The spacecraft is not equipped with any momentum wheels. Without making the small perturbation assumption, simulate the spacecraft's uncontrolled response to initial pitch and roll angle errors of $\phi(0) = 0.1°$ and $\theta(0) = -0.1°$, respectively, for 100 min.

We begin by writing a MATLAB code called *ggspin.m* (Table 6.12) for representing the state equations of a rigid spacecraft under the influence of nonlinear gravity-gradient torque. The equations are integrated in time by a Runge–Kutta solver, and the trajectory post-processed to yield gravity-gradient torqe with the following MATLAB statements:

```
global mu; mu=3.986004e14; % Earth's gravitational constant (m^3/s^2)
global n; n=2*pi/(60*90); % Orbital period (rad/s)
global Jx; Jx=2e5; % (kg\,m^2)
global Jy; Jy=6e4; % (kg\,m^2)
global Jz; Jz=1e5; % (kg\,m^2)

% Computation of rotational trajectory by Runge-Kutta integration:
dtr=pi/180;
[t,x]=ode45('ggspin',[0 12000],[0.1*dtr -0.1*dtr 0 0 0 0]');

% Computation of gravity-gradient torque by post-processing:
```

Table 6.12 Program *ggspin.m* for generating the state equations of nonlinear gravity-gradient rotational kinetics and Euler 3-2-1 kinematics of a spacecraft in a circular orbit

```
function xdot=ggspin(t,x)
% Program for gravity-gradient rotational kinetics and Euler 3-2-1 kinematics
% of a spacecraft in a circular orbit.
% x(1)=P, x(2)=q, x(3)=R (angular velocity in rad/s)
% x(4)=phi, x(5)=theta, x(6)=psi (rad)
global mu;
global n;
global Jx;
global Jy;
global Jz;
r=(mu/n^2)^(1/3);
X=r*sin(x(5));
Y=-r*sin(x(4))*cos(x(5));
Z=-r*cos(x(4))*cos(x(5));

% Nonlinear rotational kinetics with GG torque:
pdot=-(Jz-Jy)*(x(2)*x(3)-3*mu*Y*Z/r^5)/Jx;
qdot=-(Jx-Jz)*(x(1)*x(3)-3*mu*X*Z/r^5)/Jy;
rdot=-(Jy-Jx)*(x(1)*x(2)-3*mu*X*Y/r^5)/Jz;

% Nonlinear Euler 3-2-1 kinematics:
phidot=x(1)+(x(2)*sin(x(4))+x(3)*cos(x(4)))/cos(x(5));
thetadot=x(2)*cos(x(4))-x(3)*sin(x(4));
psidot=(x(2)*sin(x(4))+x(3)*cos(x(4)))/cos(x(5));
xdot=[pdot;qdot;rdot;phidot;thetadot;psidot];
```

```
r=(mu/n^2)^(1/3);
X=sin(x(:,5));
Y=-sin(x(:,4)).*cos(x(:,5));
Z=-cos(x(:,4)).*cos(x(:,5));
L=3*n^2*(Jz-Jy)*Y.*Z; % (N\,m)
M=3*n^2*(Jx-Jz)*X.*Z; % (N\,m)
N=3*n^2*(Jy-Jx)*X.*Y; % (N\,m)
```

The angular deviations and the gravity-gradient torque components are plotted in Figs. 6.29 and 6.30, respectively. There is an oscillatory attitude response – called *nutation* – which is also seen to be unstable. The unstable gravity-gradient torque components constantly feed energy, leading to angular displacements quickly building up to large magnitudes. The spacecraft can be stabilized by merely re-orienting it such that the axis **j** points toward (or away) from Earth's center and the axis **i** is normal to the plane of orbit.

6.8.1 Active Libration Damping of Gravity-Gradient Spacecraft

When a spacecraft is in a stable gravity-gradient equilibrium, it has poles on the imaginary axis causing it to have undamped oscillations if perturbed from equilibrium. Such an oscillation is called *libration*. We shall attempt to damp-out libration using a combination of reaction and momentum wheels.

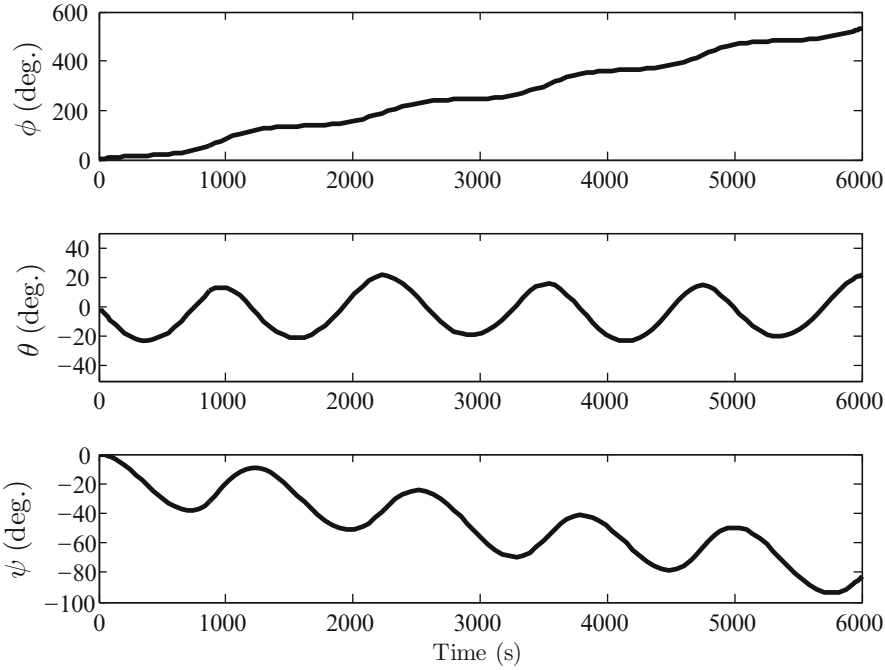


Fig. 6.29 Angular response of a spacecraft with $J_x > J_z > J_y$ with equilibrium configuration shown in Fig. 6.28 in a circular Earth orbit of 90-min period

Example 6.10. Consider an Earth satellite in a nearly polar, circular orbit of period 90 min equipped with roll and pitch reaction wheels and a pitch momentum wheel with the following parameters:

$$\begin{aligned} J_x &= J_y = 1{,}000 \text{ kg m}^2 \\ J_z &= 200 \text{ kg m}^2 \\ n &= 0.0012 \text{ rad/s} \\ h_y &= 10 \text{ N m s}. \end{aligned}$$

The two reaction wheels are driven by first-order servos of time-constant 0.1 s and have maximum available torque magnitude of ± 0.05 N m per wheel. Design an active libration damper system based on roll and pitch rate gyro feedback for damping-out the spacecraft's rate response about all axes to an initial error of $q(0) = -0.3n$ and $R(0) = 0.3n$ within 500 s.

Incorporating the gravity-gradient torques into the plant's kinetic equations with the state vector, $\mathbf{x} = (\phi, \theta, \psi, P, q, R)^{\mathrm{T}}$, and reaction wheel input vector, $\mathbf{u} = (\dot{h}_x, \dot{h}_y)^{\mathrm{T}}$, we have the following state-space coefficient matrices:

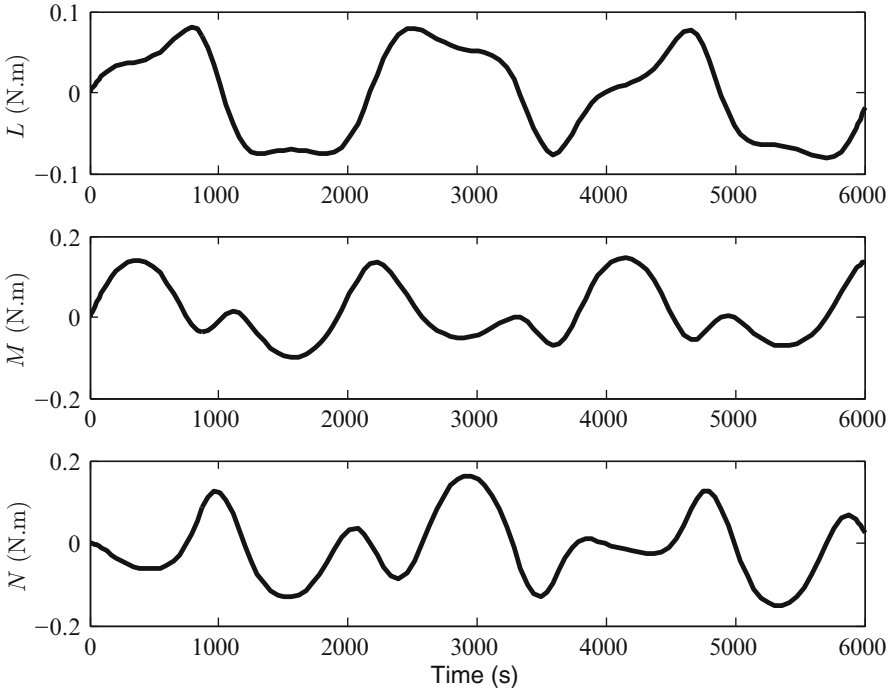


Fig. 6.30 Gravity-gradient torque of a spacecraft with $J_x > J_z > J_y$ with equilibrium configuration shown in Fig. 6.28 in a circular Earth orbit of 90 min period

$$\mathsf{A} = \begin{pmatrix} 0 & 0 & n & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -n & 0 & 0 & 0 & 0 & 1 \\ 3n^2\dfrac{J_z - J_y}{J_x} & 0 & 0 & 0 & 0 & \dfrac{h_y + n(J_z - J_y)}{J_x} \\ 0 & 3n^2\dfrac{J_z - J_x}{J_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{(J_y - J_x)n - h_y}{J_z} & 0 & 0 \end{pmatrix}$$

and

$$\mathsf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/J_x & 0 \\ 0 & 1/J_y \\ 0 & 0 \end{pmatrix}.$$

The design is carried out by LQR method by the following MATLAB statements:

```
>> n=0.0012;
>> hy=10; Jx=1000; Jy=1000; Jz=200;
>> k1=3*n*n*(Jz-Jy)/Jx;
>> k2=(hy+n*(Jz-Jy))/Jx;
>> k3=3*n*n*(Jz-Jx)/Jy;
>> k4=(-hy+n*(Jy-Jx))/Jz;
>> A=[0 0 n 1 0 0;  0 0 0 0 1  0;-n 0 0 0 0 1; k1 0 0 0 0 k2;0 k3 0 0 0 0;
        0 0 0 k4 0 0]

A =

         0         0    0.0012    1.0000         0         0
         0         0         0         0    1.0000         0
   -0.0012         0         0         0         0    1.0000
   -0.0000         0         0         0         0    0.0090
         0   -0.0000         0         0         0         0
         0         0         0   -0.0500         0         0

>> B=[zeros(3,2);1/Jx 0;0 1/Jy;0 0]

B =

            0            0
            0            0
            0            0
        0.001            0
            0        0.001
            0            0

>> C=[0 0 0 1 0 0;0 0 0 0 1 0], D=zeros(2,2);

C =
     0     0     0     1     0     0
     0     0     0     0     1     0

% Plant augmented with servos:
>> Abar=[A B;zeros(1,6) -10 0;zeros(1,6) 0 -10];
>> Bbar=[zeros(1,6) 10 0;zeros(1,7) 10]';  Cbar=[C zeros(2,2)];

% Open-loop plant's initial response:
>> sys0=ss(Abar,Bbar,Cbar,D);
>> [y0,t0,x0]=initial(sys0,[0 0 0 0 -0.3*n 0.3*n 0 0]',500);

% Regulator design by output weighted LQR:
>> [k,S,E]=lqry(A,B,C,D,20000*[0.1 0; 0 1],eye(2))

k =
 3.6437e-018 -3.1001e-017 1.3661e-016       44.721 -7.3087e-016 2.3529e-014
-5.6875e-018 -8.2803e-016 3.3966e-018 -7.3087e-016       141.42 2.1009e-015

S=
   0.0034037 -2.9995e-017-9.0764e-019  3.6437e-015 -5.6875e-015      -3.0231
-2.9995e-017      0.48875 1.4285e-016 -3.1001e-014 -8.2803e-013   2.182e-014
-9.0764e-019  1.4285e-016   0.0034037  1.3661e-013  3.3966e-015 -1.1991e-013
 3.6437e-015 -3.1001e-014 1.3661e-013        44721 -7.3087e-013  2.3529e-011
-5.6875e-015 -8.2803e-013 3.3966e-015 -7.3087e-013  1.4142e+005  2.1009e-012
      -3.0231   2.182e-014-1.1991e-013  2.3529e-011  2.1009e-012      8025.2

E =
    -0.029052
    -0.015626
  -2.142e-005 + 0.00091277i
  -2.142e-005 - 0.00091277i
      -0.1414
 -2.4442e-005
```

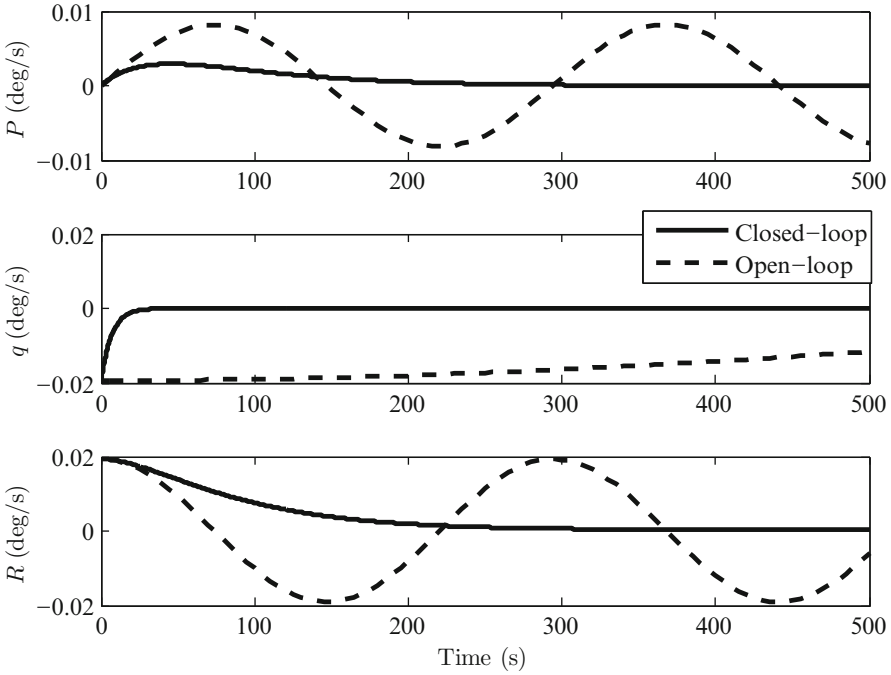


Fig. 6.31 Closed-loop initial response of a gravity-gradient spacecraft with an active libration damping system

```
% Observer design by state weighted LQR:
>> Lp=lqr(A',C',10000*eye(6),eye(2));L=Lp'

L =
      0.88688  1.1928e-007
 -6.5673e-009      -99.005
      0.24877 -1.8376e-008
       100.01 -4.1344e-016
 -4.1344e-016          100
        99.95  4.0112e-011

% Closed-loop system's initial response:
>> Ac=[Abar  -Bbar*k;L*Cbar A-L*C-B*k]; Bc=[Bbar*k; B*k];
>> sys=ss(Ac,Bc,[Cbar zeros(2,6)],zeros(2,6));
>> [y,t,x]=initial(sys,[0 0 0 0 -0.3*n 0.3*n 0 0 zeros(1,6)]',500);
>> u=-[k zeros(2,8)]*x'; % Reaction wheel torques
```

The results are plotted in Figs. 6.31 and 6.32. Observe the undamped roll-yaw libration and a long-period pitch oscillation of the open-loop spacecraft, which are quickly damped by the reaction wheels. The roll-rate and yaw-rate transients settle in about 450 s, while the pitch-rate is brought to zero in about 30 s (Fig. 6.31). The reaction wheel torques are seen in Fig. 6.32 to be limited to ± 0.05 N m, as specified. However, being based purely on rate feedback, the controller lacks integral action

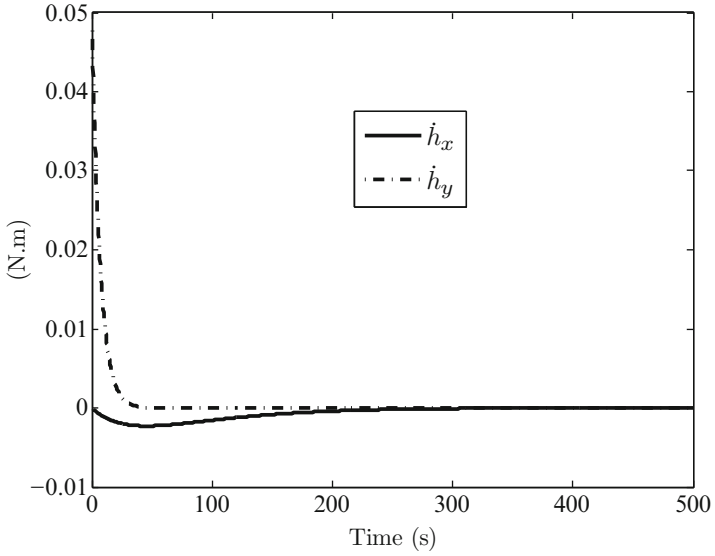


Fig. 6.32 Closed-loop reaction wheel torques of a gravity-gradient with an active libration damping system

and thus it is possible for the spacecraft to slowly drift away from the equilibrium. In order to avoid such an eventuality, a feedback loop from a rate-integrating gyro (or horizon scanner) is generally added.

6.9 Summary

For orbital and attitude control of spacecraft, the controller design is based upon tracking a reference orbit or attitude profile. An orbit control system comprises an actuator capable of applying an external force on the spacecraft (generally a rocket thruster), sensors for measuring instantaneous orbital position (or speed) relative to the central body, and a control law for driving the plant toward a zero steady-state positional error. While radial thrust offers the best scope of controlling orbital shape when circumferential perturbations are absent, a tangential thrust is necessary for correcting the angular momentum in the presence of such disturbances (such as atmospheric drag), or for decreasing the angular momentum for de-orbiting a satellite. Control of orbital plane requires that the thrust should be applied normal to the plane of orbit. With constant thrust, a switching of the thrust direction becomes necessary due to the nonlinear and underactuated plant. In order to avoid chatter caused by rapid switching, a dead-zone is generally built into the actuating system. When thrust modulating is possible, switching can be eliminated in favor of a smooth, linear regulator with a proportional feedback gain.

Attitude control system of a spacecraft can be designed separately from the orbit control system. Of the available means of applying a control torque, a rotor powered by solar arrays is the best as it does not involve fuel expenditure. The rotors can be either fixed axis (momentum and reaction wheels) or variable axis (CMG, VSCMG) type. A momentum wheel mounted along a principal axis is used for biasing (coupling) the rotation about the other two axes and thus increases the controllability of the attitude plant. While multiple reaction wheels are required for general attitude control, the same can be achieved using a single CMG rotor. However, a CMG suffers from inherent nonlinearity that is not present in a reaction wheel system. While solar radiation pressure exerts a nearly constant torque that can be easily compensated for using reaction wheel-based PD control, the gravity-gradient disturbance requires a multivariable control system. Active damping of nutation and libration is possible with a suitably designed control system.

Exercises

6.1. Using the polar coordinates shown in Fig. 6.1, derive the planar equations of translational motion of a vectored thrust spacecraft in a scalar form, (6.2) and (6.3).

6.2. Formulate the problem of guiding a spacecraft from an initial circular orbit to a final, coplanar circular orbit by the application of a small but continuous, constant thrust acceleration, $u = T/m$. The direction of the thrust vector from the local flight direction is given by angle, $\alpha(t)$, and can be regarded as the input variable. The total time of flight is a free variable. Is it possible to have a closed-form solution to the problem?

6.3. For the problem of Exercise 6.2, take the special case of tangential thrust ($\alpha = 0$) and try to derive a closed-form solution with the assumption of a small, constant acceleration magnitude, u, compared to the acceleration due to gravity.

6.4. Consider a spacecraft that can only fire its rocket engine either on or off to produce a constant, positive radial acceleration. Design a suitable switching control system to raise the orbital radius as desired, beginning from an initial circular orbit. If the constant acceleration input is 1/2 of the initial gravitational acceleration, select the switching controller gain such that the radius is doubled within an error of $\pm$ 5%.

6.5. Formulate a problem for plane-change maneuver wherein a small but continuous, constant thrust is applied normal to the local orbital plane by suitably adjusting the thrust direction from the local vertical, $\alpha(t)$. Do we have a linear plant? How can such a problem be solved?

6.6. Can the spacecraft of Example 6.7 be stabilized if the pitch momentum wheel is replaced by a yaw momentum wheel? Explain your answer.

6.7. Can the spacecraft of Example 6.7 be stabilized if the pitch reaction wheel is replaced by a yaw reaction wheel? Explain your answer.

6.8. What happens to the design of Example 6.8 if the pitch momentum wheel is removed? Explain your answer.

6.9. For the spacecraft of Example 6.9, re-orient the vehicle such that the axes **i** and **j** are interchanged, and repeat the simulation. Is the spacecraft stable in the new equilibrium configuration?

6.10. For the spacecraft of Example 6.9, design an active nutation damping system using a roll reaction wheel of maximum torque magnitude 5 N m and a pitch reaction wheel of 1 N m torque magnitude. Test the design with the initial perturbation specified in the Example 6.9. What is the smallest achievable settling-time of all rate transients?

6.11. To the design of Exercise 6.10, add a pitch momentum wheel of angular momentum 100 N m s and re-design for the given specifications. Do the required wheel torques increase or decrease with the addition of the momentum wheel?

6.12. For the axisymmetric spacecraft of Example 6.9, replace the roll reaction wheel by a yaw reaction wheel and re-design the control system with the same specifications. What differences (if any) are observed, and why?

6.13. Re-design the active libration damper of Example 6.9 by adding a feedback loop from a roll rate-integrating gyro such that the roll angle drift is limited to within $\pm 0.1^\circ$ for the specified initial error. (*Hint:* Re-design the new observer with the three outputs, and see whether an adjustment in the regulator gains is required for meeting the specifications.)

Appendix A
Linear Optimal Control

Without entering into mathematical details, we introduce the basic results of linear optimal control theory, for which the reader can refer to [1, 4, 6, 18].

Consider a linear time-varying plant described by the following state equation and initial condition:

$$\dot{\mathbf{x}} = \mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x_0}. \tag{A.1}$$

It is desired that the following quadratic objective function of state and control variables is to be minimized:

$$J = \frac{1}{2}\mathbf{x}^{\mathrm{T}}(t_{\mathrm{f}})\mathsf{Q}_{\mathrm{f}}\mathbf{x}(t_{\mathrm{f}}) + \frac{1}{2}\int_{t_0}^{t_{\mathrm{f}}} \left[\mathbf{x}^{\mathrm{T}}(t)\mathsf{Q}(t)\mathbf{x}(t) + 2\mathbf{x}^{\mathrm{T}}(t)\mathsf{S}(t)\mathbf{u}(t) + \mathbf{u}^{\mathrm{T}}(t)\mathsf{R}(t)\mathbf{u}(t)\right] \mathrm{d}t, \tag{A.2}$$

where $\mathbf{x}(t_{\mathrm{f}})$ is the terminal state vector at final time, t_{f}, Q_{f}, $\mathsf{Q}(t)$, and $\mathsf{R}(t)$ are symmetric (but $\mathsf{S}(t)$ could be asymmetric) cost coefficient matrices. Since $J \geq 0$, we also require that Q_{f}, $\mathsf{Q}(t)$, and $\mathsf{R}(t)$ be at least positive semidefinite. The integrand is expressed as the following *Lagrangian* function:

$$L[\mathbf{x}(t), \mathbf{u}(t), t] = \frac{1}{2}\mathbf{x}^{\mathrm{T}}(t)\mathsf{Q}(t)\mathbf{x}(t) + \mathbf{x}^{\mathrm{T}}(t)\mathsf{S}(t)\mathbf{u}(t) + \frac{1}{2}\mathbf{u}^{\mathrm{T}}(t)\mathsf{R}(t)\mathbf{u}(t). \tag{A.3}$$

A.1 Derivation of Matrix Riccati Equation

Let the following quadratic *value function*,

$$V[\mathbf{x}(t), t] = \frac{1}{2}\hat{\mathbf{x}}^{\mathrm{T}}(t)\mathsf{P}(t)\hat{\mathbf{x}}(t), \tag{A.4}$$

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0, © Springer Science+Business Media, LLC 2011

with a positive definite, symmetric matrix, $\mathsf{P}(t)$, satisfy the *Hamilton-Jacobi-Bellman* (HJB) equation given by

$$-\frac{\partial \hat{V}}{\partial t} = L[\hat{\mathbf{x}}, \hat{\mathbf{u}}, t] + \frac{\partial \hat{V}}{\partial \mathbf{x}}(\mathsf{A}\hat{\mathbf{x}} + \mathsf{B}\hat{\mathbf{u}}). \tag{A.5}$$

with the terminal boundary condition,

$$\hat{V}[\hat{\mathbf{x}}(t_\mathrm{f}), t_\mathrm{f}] = \frac{1}{2}\hat{\mathbf{x}}^\mathrm{T}(t_\mathrm{f})\mathsf{Q}_\mathrm{f}\hat{\mathbf{x}}(t_\mathrm{f}). \tag{A.6}$$

Here, $\hat{\mathbf{x}}(t)$ is the optimal trajectory, $\hat{\mathbf{u}}(t)$ the corresponding control history, and $\hat{V}(\hat{\mathbf{x}}, t)$ the optimal value function.

For the minimization of J with respect to the control vector, $\mathbf{u}(t)$, subject to the state equation of the plant, (A.1), we adjoin the Lagrangian with the following *Lagrange multiplier vector*, $\boldsymbol{\lambda}(t)$, resulting in a *Hamiltonian* function, $H[\mathbf{x}, \mathbf{u}, t]$, as follows:

$$\boldsymbol{\lambda}^\mathrm{T} = \frac{\partial V}{\partial \mathbf{x}} = \mathbf{x}^\mathrm{T}(t)\mathsf{P}(t), \tag{A.7}$$

$$H[\mathbf{x}, \mathbf{u}, t] = L[\mathbf{x}(t), \mathbf{u}(t), t] + \boldsymbol{\lambda}^\mathrm{T}(t)[\mathsf{A}(t)\mathbf{x}(t) + \mathsf{B}(t)\mathbf{u}(t)]. \tag{A.8}$$

In order to derive the optimal control history, $\hat{\mathbf{u}}(t)$, we differentiate H with respect to $\mathbf{u}$ and equate the result to zero, $H_\mathbf{u} = \mathsf{0}$:

$$\hat{\mathbf{x}}^\mathrm{T}(t)\mathsf{S}(t) + \hat{\mathbf{u}}^\mathrm{T}(t)\mathsf{R}(t) + \hat{\mathbf{x}}^\mathrm{T}(t)\hat{\mathsf{P}}(t)\mathsf{B}(t) = \mathsf{0}, \tag{A.9}$$

or

$$\hat{\mathbf{u}}(t) = -\mathsf{R}^{-1}(t)[\mathsf{B}^\mathrm{T}(t)\hat{\mathsf{P}}(t) + \mathsf{S}^\mathrm{T}(t)]\hat{\mathbf{x}}(t). \tag{A.10}$$

Equation (A.10) is called the *linear optimal feedback control law*. Since the matrix $\mathsf{P}(t)$ is chosen to be positive definite, we have $V[\mathbf{x}(t), t] > 0$ for all $\mathbf{x}(t)$. By substituting the linear, optimal feedback control law into the optimal Hamiltonian,

$$\hat{H} = -\frac{\partial \hat{V}}{\partial t} = -\hat{\mathbf{x}}^\mathrm{T}\dot{\hat{\mathsf{P}}}\hat{\mathbf{x}}, \tag{A.11}$$

the *Legendre-Clebsch condition*,

$$H_{\mathbf{uu}} > \mathsf{0}, \tag{A.12}$$

is seen to be satisfied, thereby implying a negative definite matrix, $\dot{\hat{P}}(t)$ (i.e., $\dot{V}[\mathbf{x}(t), t] < 0$ for all $\mathbf{x}(t)$). Thus, we have a globally asymptotically stable control system by *Lyapunov's theorem*. We note that the Legendre-Clebsch condition requires that $\mathsf{R}(t)$ must be positive definite.

Substitution of the optimal value function into the HJB equation, (A.5), we have

$$\hat{\mathbf{x}}^{\mathrm{T}}\dot{\hat{\mathsf{P}}}\hat{\mathbf{x}} = -\hat{\mathbf{x}}^{\mathrm{T}}[(\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}})^{\mathrm{T}}\hat{\mathsf{P}} + \hat{\mathsf{P}}(\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}) - \hat{\mathsf{P}}\mathsf{B}\mathsf{R}^{-1}\mathsf{B}^{\mathrm{T}}\hat{\mathsf{P}} + \mathsf{Q} - \mathsf{S}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}]\hat{\mathbf{x}}, \tag{A.13}$$

which yields the following *matrix Riccati equation* to be satisfied by the optimal matrix, $\hat{\mathsf{P}}$:

$$-\dot{\hat{\mathsf{P}}} = \mathsf{Q} + (\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}})^{\mathrm{T}}\hat{\mathsf{P}} + \hat{\mathsf{P}}(\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}) - \hat{\mathsf{P}}\mathsf{B}\mathsf{R}^{-1}\mathsf{B}^{\mathrm{T}}\hat{\mathsf{P}} - \mathsf{S}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}, \tag{A.14}$$

which must be solved subject to the boundary condition,

$$\hat{\mathsf{P}}(t_{\mathrm{f}}) = \mathsf{Q}_{\mathrm{f}}. \tag{A.15}$$

Matrix Riccati equation (MRE) is fundamental to linear, optimal control and must be integrated backward in time by an appropriate numerical scheme, called *nonlinear* (or *dynamic*) *programming*.

A.2 Linear Time-Invariant System

The governing equation for a time-invariant linear, quadratic regulator (LQR) problem is derived simply by putting $\dot{\hat{\mathsf{P}}} = 0$ in the matrix Riccati equation, (A.14), resulting in the following *algebraic Riccati equation* (ARE):

$$0 = \mathsf{Q} - \mathsf{S}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}} + (\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}})^{\mathrm{T}}\hat{\mathsf{P}} + \hat{\mathsf{P}}(\mathsf{A}-\mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}) - \hat{\mathsf{P}}\mathsf{B}\mathsf{R}^{-1}\mathsf{B}^{\mathrm{T}}\hat{\mathsf{P}}. \tag{A.16}$$

The optimal feedback control law is obtained from the algebraic Riccati solution,

$$\hat{\mathbf{u}}(t) = -\mathsf{R}^{-1}[\mathsf{B}^{\mathrm{T}}\hat{\mathsf{P}} + \mathsf{S}^{\mathrm{T}}]\hat{\mathbf{x}}(t). \tag{A.17}$$

where the cost coefficient matrices, $\mathsf{Q}, \mathsf{R}, \mathsf{S}$, are constants. For asymptotic stability of the regulated system, all the eigenvalues of the closed-loop dynamics matrix,

$$\mathsf{A} - \mathsf{B}\mathsf{R}^{-1}\left(\mathsf{B}^{\mathrm{T}}\hat{\mathsf{P}} + \mathsf{S}^{\mathrm{T}}\right),$$

must be in the left-half s-plane (Chap. 1), which requires that the ARE solution, $\hat{\mathsf{P}}$ must be a symmetric, and positive semidefinite matrix. It can be proved [6] that

if the following sufficient conditions are satisfied, there exists a unique, symmetric, positive semidefinite solution to the ARE:

(a) The control cost coefficient matrix, R, is symmetric and positive definite, the matrix $(\mathsf{Q} - \mathsf{S}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}})$ is symmetric and positive semidefinite, and the pair $(\mathsf{A} - \mathsf{B}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}}, \mathsf{Q} - \mathsf{S}\mathsf{R}^{-1}\mathsf{S}^{\mathrm{T}})$ is detectable (Chap. 1).
(b) The pair (A, B) is either controllable, or at least stabilizable (Chap. 1).

Answers to Selected Exercises

Exercises of Chapter 1

1.1 (a) 2. (b) The only equilibrium point is given by $y = 0, \dot{y} = 0$.

1.2 (a) 4.

1.4 (a) 3.

1.5 (a) 8. (b) No. (c) 4.

1.10 (a) Unstable due to a pair of complex eigenvalues with positive real part. (b) Yes. (c) Yes. (d) Yes.

Exercises of Chapter 2

2.2 $r = 13,091$ km, $v = 5.8587$ km/s, $\phi = 8.9483°$

2.3 $r = 7251.6$ km, $v = 7.3864$ km/s, $\phi = 4.1216°$, and $\tau = 1689$ s.

2.4 $\mathbf{r} = 4710.1\mathbf{I} - 906.46\mathbf{J} - 5438.7\mathbf{K}$ km, $\mathbf{v} = 4.9494\mathbf{I} + 4.7188\mathbf{J} + 2.792\mathbf{K}$ km/s.

Exercises of Chapter 3

3.1 For $0 \leq \zeta < 1$:

$$g(t) = \frac{e^{-ct/(2m)}}{\sqrt{km\left(1 - \frac{c^2}{4km}\right)}} \sin\left(t\sqrt{\frac{k}{m} - \frac{c^2}{4m^2}}\right); \quad (t > 0)$$

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0, © Springer Science+Business Media, LLC 2011

3.6 (a) No. (b) $s = 1$, $s = -5.65$, and $s = -0.354$. (c) No, because the plant is both unstable (pole in the right-half s-plane) and non-minimum phase (zero in the right-half s-plane).

3.7 (a) $G(s) = \frac{90}{s^2+0.95s+9}$.
(b) Infinite gain margin; phase margin 6.01° with gain crossover frequency 9.93 rad/s.
(c) Steady-state error to unit step desired output is 9 units (900%).

3.9 (a) $s_{1,2} = 5.15 \times 10^{-5} \pm 0.101\text{i}$, $s_{3,4} = -0.5 \pm 0.86\text{i}$. No.
(b) $\omega_{n1} = 0.101$ rad/s, $\zeta_1 = -5.13 \times 10^{-4}$. $\omega_{n2} = 0.995$ rad/s, $\zeta_2 = 0.503$.

3.10 (a) No.
(b) With the following state-space representation of the plant,

$$\mathsf{A} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}; \quad \mathsf{B} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathsf{C} = \begin{pmatrix} 0 & 2.5 & 2.5 \end{pmatrix}; \quad \mathsf{D} = 0,$$

we require a state feedback gain matrix,

$$\mathsf{K} = \begin{pmatrix} 1 & 0.5 & 0.25 \end{pmatrix}.$$

3.13

$$u(t) = -0.2996x(t) - 7.7845\dot{x}(t).$$

Exercises of Chapter 4

4.6 Maximum overshoots: 0.56 m/s (airspeed), −3.15 m (altitude), −1.6° (angle-of-attack), −2.7° (pitch angle), −18.56° (elevator angle). Settling-time: 14.7 s.

4.8

$$\frac{\theta(s)}{\delta_{\text{Ec}}(s)} = \frac{1.865s^4 - 21.98s^3 - 69.98s^2 - 58.63s - 2.352}{s^7 + 22.6s^6 + 142s^5 + 478.4s^4 + 801s^3 + 802.3s^2 + 439.1s + 120}.$$

Gain margin: 12.3 dB, phase crossover frequency 0.544 rad/s. Phase margin: infinite.

4.11

$$\frac{\theta(s)}{u_d(s)} = \frac{0.003431s^2 + 0.0322s - 0.005521}{s^6 + 9.909s^5 + 37.85s^4 + 66.57s^3 + 8.542s^2 + 1.122s + 0.03894}.$$

Gain margin: 17 dB, phase crossover frequency 0 rad/s. Phase margin: infinite.

4.20

$$\mathsf{K} = (1.8186,\ 1,\ 2.2562,\ 0.28356,\ 0.66191),$$
$$\mathsf{L}^{\mathrm{T}} = (7.5928,\ 13.984,\ 1.67,\ -27.832,\ 41.374).$$

Gain margin: 3.72 dB, phase crossover frequency: 0.143 rad/s. Phase margin: infinite.

Exercises of Chapter 5

5.3

$$\tau_x(s) = (2,580 + 2,000s)\left[\Phi_d(s) - \Phi(s)\right].$$

5.4

$$\mu_{1d}(s) = -(75 + 23s)\left[\theta_d(s) - \theta(s)\right].$$

5.7

$$\mathsf{K} = (0.14356,\ 0.38653,\ 0.49552),$$
$$\mathsf{L}^{\mathrm{T}} = (7.4965,\ 7.0303,\ 11.405).$$

The design point settling-time is reduced to 50 s compared to about 70 s of Example 5.4. All overshoots in the transient response and the maximum gimbal deflection are reduced by about 5%. Gain margin: 36.8 dB, phase crossover frequency: 13.3 rad/s. Phase margin: 151 °, gain crossover frequency: 0.393 rad/s.

Exercises of Chapter 6

6.1

$$\ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} = \frac{T}{m}\sin\alpha,$$
$$r\ddot{\theta} + 2\dot{r}\dot{\theta} = \frac{T}{m}\cos\alpha.$$

6.3

$$r(t) = \frac{1}{\frac{1}{r_0} - 2\frac{u}{\sqrt{\mu r_0}}t + \frac{u^2t^2}{\mu}}.$$

6.6 No, because of a constant rolling moment, $-nh_z$, due to the yaw momentum wheel.

6.7 No, because the replacement of pitch reaction wheel by yaw reaction wheel results in an uncontrollable plant.

References

1. Athans, M., and Falb, P.L.: Optimal Control. Dover, New York (2007)
2. Blakelock, J.H.: Automatic Control of Aircraft and Missiles. Wiley-Interscience, New York (1991)
3. Britting, K.R.: Inertial Navigation Systems Analysis. Wiley Interscience, Somerset, NJ (1971)
4. Bryson, A.E., and Ho, Y.: Applied Optimal Control. Wiley, New York (1979)
5. Etkin, B., and Reid, L.D.: Dynamics of Flight: Stability and Control. Wiley, New York (1995)
6. Glad, T., and Ljung, L.: Control Theory–Multivariable and Nonlinear Methods. Taylor and Francis, New York (2000)
7. Hoak, D.E., *et al.*: USAF Stability and Control Datcom. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, O.H. (1978)
8. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
9. Katz, J., and Plotkin, A.: Low Speed Aerodynamics. McGraw-Hill, New York (1991)
10. Kautsky, J., Nichols, N.K., and Van Dooren, P.: Robust Pole Assignment in Linear State Feedback. International Journal of Control, **41**, 1129–1155 (1985)
11. Kreyszig, E.: Advanced Engineering Mathematics. Wiley, New York (2001)
12. Klawans, E.: The Vanguard Rocket Launching Vehicle–An Engineering Summary. Engr. Rept. No. 11022, The Martin Company (1960)
13. Mattheij, R.M.M., and Molenaar, J.: Ordinary Differential Equations in Theory and Practice. SIAM Classics in Applied Mathematics 43, SIAM, Philadelphia, PA (2002)
14. Pinkham, G.: Reference Solution for Low Thrust Trajectories. Journal of American Rocket Society, **32**, 775–776 (1962)
15. Shampine, L.F., Reichelt, M.W., and Kierzenka, J.: Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c. ftp://ftp.mathworks.com/pub/doc/papers/bvp/
16. Shampine, L.F., Gladwell, I., and Thompson, S.: Solving ODEs with MATLAB. Cambridge University Press, Cambridge, UK (2003)
17. Slotine, J.E., and Li, W.: Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, N.J. (1991)
18. Stengel, R.F.: Optimal Control and Estimation. Dover, New York (1994)
19. Stoer, J., and Bulirsch, R.: Introduction to Numerical Analysis. Springer-Verlag, New York (2002)
20. Tewari, A.: Modern Control Design with MATLAB and Simulink. Wiley, Chichester (2002)
21. Tewari, A.: Atmospheric and Space Flight Dynamics–Modeling and Simulation with MATLAB and Simulink. Birkhäuser, Boston (2006)

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0, © Springer Science+Business Media, LLC 2011

22. Wagner, H.: Uber die Enstehung des Dynamischen Autriebes von Tragflugeln. Zeitschrift fur Angewandte Mathematik und Mechanik, **5**, 17–35 (1925)
23. Wertz, J.R. (ed): Spacecraft Attitude Determination and Control. Kluwer Academic Publishers, Dordrecht (1978)

Index

A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*, Control Engineering,
DOI 10.1007/978-0-8176-4864-0, © Springer Science+Business Media, LLC 2011