15

Control Techniques

Using inverse kinematics, we can calculate the joint kinematics for a desired geometric path of the end-effector of a robot. Substitution of the joint kinematics in equations of motion provides the actuator commands. Applying the commands will move the end-effector of the robot on the desired path ideally. However, because of perturbations and non-modeled phenomena, the robot will not follow the desired path. The techniques that minimize or remove the difference are called the *control techniques*.

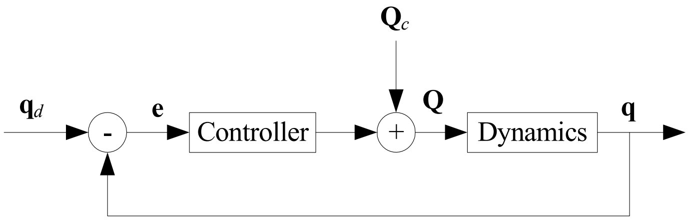

FIGURE 15.1. Illustration of feedback control algorithm.

15.1 Open and Closed-Loop Control

A robot is a mechanism with an actuator at each joint i to apply a force or torque to derive the link (i). The robot is instrumented with position, velocity, and possibly acceleration sensors to measure the joint variables' kinematics. The measured values are usually kinematics information of the frame B_i, attached to the link (i), relative to the frame B_{i-1} or B_0.

To cause each joint of the robot to follow a desired motion, we must provide the required torque command. Assume that the desired path of joint variables $\mathbf{q}_d = \mathbf{q}(t)$ are given as functions of time. Then, the required torques that cause the robot to follow the desired motion are calculated by the equations of motion and are equal to

$$\mathbf{Q}_c = \mathbf{D}(\mathbf{q}_d)\,\ddot{\mathbf{q}}_d + \mathbf{H}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \mathbf{G}(\mathbf{q}_d) \tag{15.1}$$

where the subscripts d and c stand for *desired* and *controlled*, respectively.

In an ideal world, the variables can be measured exactly and the robot can perfectly work based on the equations of motion (15.1). Then, the actuators' *control command* $\mathbf{Q}_c$ can cause the desired path $\mathbf{q}_d$ to happen.

R.N. Jazar, *Theory of Applied Robotics,* 2nd ed., DOI 10.1007/978-1-4419-1750-8_15,

This is an *open-loop control algorithm*, that the control commands are calculated based on a known desired path and the equations of motion. Then, the control commands are fed to the system to generate the desired path. Therefore, in an open-loop control algorithm, we expect the robot to follow the designed path, however, there is no mechanism to compensate any possible error.

Now assume that we are watching the robot during its motion by measuring the joints' kinematics. At any instant there can be a difference between the actual joint variables and the desired values. The difference is called *error* and is measured by

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \tag{15.2}$$

$$\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d. \tag{15.3}$$

Let's define a control law and calculate a new control command vector by

$$\mathbf{Q} = \mathbf{Q}_c + \mathbf{k}_D \dot{\mathbf{e}} + \mathbf{k}_P \mathbf{e} \tag{15.4}$$

where $\mathbf{k}_P$ and $\mathbf{k}_D$ are *constant control gains*. The control law compares the actual joint variables $(\mathbf{q}, \dot{\mathbf{q}})$ with the desired values $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$, and generates a command proportionally. Applying the new control command changes the dynamic equations of the robot to produce the actual joint variables $\mathbf{q}$.

$$\mathbf{Q}_c + \mathbf{k}_D \dot{\mathbf{e}} + \mathbf{k}_P \mathbf{e} = \mathbf{D}(\mathbf{q})\, \ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \tag{15.5}$$

Figure 15.1 illustrates the idea of this control method in a *block diagram*. This is a *closed-loop control algorithm*, in which the control commands are calculated based on the difference between actual and desired variables. Reading the actual variables and comparing with the desired values is called *feedback*, and because of that, the closed-loop control algorithm is also called a *feedback control algorithm*.

The controller provides a signal proportional to the error and its time rate. This signal is added to the predicted command $\mathbf{Q}_c$ to compensate the error.

The principle of feedback control can be expressed as: *Increase the control command when the actual variable is smaller than the desired value and decrease the control command when the actual variable is larger than the desired value.*

Example 387 *Mass-spring-damper oscillator.*

Consider a linear oscillator made by a mass-spring-damper system shown in Figure 15.2. The equation of motion for the oscillator under the effect of an external force f is

$$m\ddot{x} + c\dot{x} + kx = f \tag{15.6}$$

where, f is the control command, m is the mass of the oscillating object, c is the viscous damping, and k is the stiffness of the spring. The required

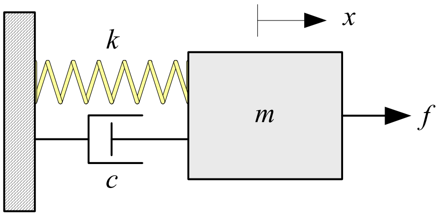

FIGURE 15.2. A linear mass-spring-damper oscillator.

force to achieve a desired displacement $x_d = x(t)$ *is calculated from the equation of motion.*

$$f_c = m\ddot{x}_d + c\dot{x}_d + kx_d \tag{15.7}$$

The open-loop control algorithm is shown in Figure 15.3(a).

To remove any possible error, we may use the difference between the desired and actual outputs $e = x - x_d$*, and define a control law.*

$$f = f_c + k_D\dot{e} + k_P e \qquad k_D > 0 \qquad k_P > 0 \tag{15.8}$$

The new control law uses a feedback command as shown in Figure 15.3(b).

It is also possible to define a new control law only based on the error signal such as

$$f = -k_D\dot{e} - k_P e. \tag{15.9}$$

Employing this law, we can define a more compact feedback control algorithm and change the equation of motion to

$$m\ddot{x} + (c + k_D)\,\dot{x} + (k + k_P)\,x = k_D\dot{x}_d + k_P x_d. \tag{15.10}$$

The equation of the system can be summarized in a block diagram as shown in Figure 15.3(c).

A general scheme of a feedback control system may be explained so that a signal from the output feeds back to be compared to the input. This feedback signal closes a loop and makes it reasonable to use the words ***feedback*** *and* ***close-loop****. The principle of a closed loop control is to detect any error between the actual output and the desired. As long as the error signal is not zero, the controller keeps changing the control command so that the error signal converges to zero.*

Example 388 *Stability of a controlled system.*

Consider a linear mass-spring-damper oscillator as shown in Figure 15.2 with the equation of motion given by

$$m\ddot{x} + c\dot{x} + kx = f. \tag{15.11}$$

We define a control law based on the actual output

$$f = -k_D\dot{x} - k_P x \qquad k_D > 0 \qquad k_P > 0 \tag{15.12}$$

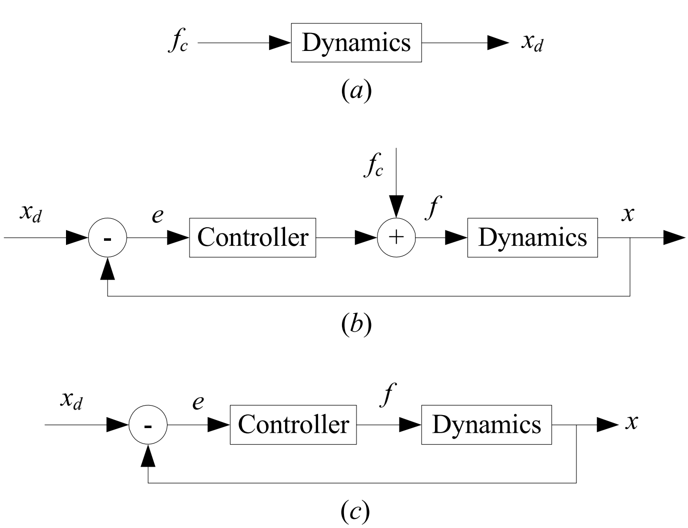

FIGURE 15.3. Open-loop and closed-loop control algorithms for a linear oscillator.

and transform the equation of motion to

$$m\ddot{x} + (c + k_D)\,\dot{x} + (k + k_P)\,x = 0. \tag{15.13}$$

By comparison with the open-loop equation (15.11), the closed loop equation shows that the oscillator acts as a free vibrating system under the action of new stiffness $k + k_P$ *and damping* $c + k_D$*. Hence, the control law has changed the apparent stiffness and damping of the actual system. This example introduces the most basic application of control theory to improve the characteristics of a system and run the system to behave in a desired manner.*

A control system must be stable when the desired output of the system changes, and also be able to eliminate the effect of a disturbance. Stability of a control system is defined as: The output must remain bounded for a given input or a bounded disturbance function.

To investigate the stability of the system, we must solve the closed loop differential equation (15.13). The equation is linear and therefore, it has an exponential solution.

$$x = e^{\lambda t} \tag{15.14}$$

Substituting the solution into the equation (15.13) provides the characteristic equation

$$m\lambda^2 + (c + k_D)\,\lambda + (k + k_P) = 0 \tag{15.15}$$

with two solutions

$$\lambda_{1,2} = -\frac{c + k_D}{2m} \pm \frac{\sqrt{(c + k_D)^2 - 4m(k + k_P)}}{2m}. \tag{15.16}$$

The nature of the solution (15.14) depends on λ_1 *and* λ_2*, and therefore on* k_D *and* k_P*. The roots of the characteristic equation are complex*

$$\lambda = -a \pm bi \tag{15.17}$$

$$a = \frac{c + k_D}{2m} \tag{15.18}$$

$$b = \frac{\sqrt{4m(k + k_P) - (c + k_D)^2}}{2m} \tag{15.19}$$

provided the gains are such that

$$(c + k_D)^2 < 4m(k + k_P). \tag{15.20}$$

In this case, the solution of the equation of motion is

$$x = Ce^{-\xi\omega_n t} \sin\left(\omega_n \sqrt{1 - \xi^2} t + \varphi\right) \tag{15.21}$$

where,

$$\omega_n = \sqrt{\frac{k + k_P}{m}} = \sqrt{a^2 + b^2} \tag{15.22}$$

$$\xi = \frac{c + k_D}{2\sqrt{m(k + k_P)}} = \frac{a}{\sqrt{a^2 + b^2}}. \tag{15.23}$$

The parameter ω_n *is called* ***natural frequency****, and* ξ *is the* ***damping ratio*** *of the system. The damping ratio controls the behavior of the system according to the following categories:*

1. *If* $\xi = 0$*, then the characteristic values are purely imaginary,* $\lambda_{1,2} = \pm bi = \pm i\frac{1}{2m}\sqrt{4m(k + k_P)}$*. In this case, the system has no damping, and therefore, it oscillates with a constant amplitude around the equilibrium,* $x = 0$*, forever.*

2. *If* $0 < \xi < 1$*, then the system is* ***under-damped*** *and it oscillates around the equilibrium with a decaying amplitude. The system is asymptotically stable in this case.*

3. *If* $\xi = 1$*, then the system is* ***critically-damped****. A critically damped oscillator has the fastest return to the equilibrium in an unoscillatory manner.*

4. *If $\xi > 1$, then the system is* ***over-damped*** *and it slowly returns to the equilibrium in an unoscillatory manner. The characteristic values are real and the solution of an over-damped oscillator is*

$$\begin{aligned} x &= Ae^{\lambda_1 t} + Be^{\lambda_2 t} \\ (\lambda_{1,2}) &\in \mathbb{R}. \end{aligned} \tag{15.24}$$

5. *If $\xi < 0$, then the system is unstable because the solution is*

$$\begin{aligned} x &= Ae^{\lambda_1 t} + Be^{\lambda_2 t} \\ \mathrm{Re}(\lambda_{1,2}) &> 0 \end{aligned} \tag{15.25}$$

and shows a motion with an increasing amplitude.

Example 389 ★ *Solution of a characteristic equation.*
Consider a system with the following characteristic equation:

$$\lambda^2 + 6\lambda + 10 = 0. \tag{15.26}$$

Solutions of this equation are

$$\lambda_{1,2} = -3 \pm i \tag{15.27}$$

showing a stable system because $\mathrm{Re}\,(\lambda_{1,2}) = -3 < 0$.
Characteristic equations are linear polynomials. Hence, it is possible to use numerical methods, such as Newton-Raphson, to find the solution and determine the stability of the system.

Example 390 ★ *Complex roots.*
In case the characteristic equation has complex roots

$$\lambda_{1,2} = a \pm bi \tag{15.28}$$

we may employ the Euler formula

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{15.29}$$

and show that the solution can be written in the form

$$\begin{aligned} x &= C_1 e^{at}\left(\cos bt + i\sin bt\right) + C_2 e^{at}\left(\cos bt - i\sin bt\right) \\ &= e^{at}\left(A\cos bt + B\sin bt\right) \end{aligned} \tag{15.30}$$

where, C_1 and C_2 are complex, and A and B are real numbers according to

$$A = C_1 + C_2 \tag{15.31}$$

$$B = \left(C_1 - C_2\right) i. \tag{15.32}$$

Example 391 *Robot Control Algorithms.*

Robots are nonlinear dynamical systems, and there is no general method for designing a nonlinear controller to be suitable for every robot in every mission. However, there are a variety of alternative and complementary methods, each best applicable to particular class of robots in a particular mission. The most important control methods are as follows:

***Feedback Linearization** or **Computed Torque Control Technique**. In feedback linearization technique, we define a control law to obtain a linear differential equation for error command, and then use the linear control design techniques. The feedback linearization technique can be applied to robots successfully, however, it does not guarantee robustness according to parameter uncertainty or disturbances.*

This technique is a model-based control method, because the control law is designed based on a nominal model of the robot.

***Linear Control Technique**. The simplest technique for controlling robots is to design a linear controller based on the linearization of the equations of motion about an operating point. The linearization technique locally determines the stability of the robot. Proportional, integral, and derivative, or any combination of them, are the most practical linear control techniques.*

***Adaptive Control Technique**. Adaptive control is a technique for controlling uncertain or time-varying robots. Adaptive control technique is more effective for low DOF robots.*

***Robust and Adaptive Control Technique.** In the robust control method, the controller is designed based on the nominal model plus some uncertainty. Uncertainty can be in any parameter, such as the load carrying by the end-effector. For example, we develop a control technique to be effective for loads in a range of* $1 - 10\,\text{kg}$.

***Gain-Scheduling Control Technique**. Gain-scheduling is a technique that tries to apply the linear control techniques to the nonlinear dynamics of robots. In gain-scheduling, we select a number of control points to cover the range of robot operation. Then at each control point, we make a linear time-varying approximation to the robot dynamics and design a linear controller. The parameters of the controller are then interpolated or scheduled between control points.*

15.2 Computed Torque Control

Dynamics of a robot can be expressed in the form

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \tag{15.33}$$

where $\mathbf{q}$ is the vector of joint variables, and $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the torques applied at joints. Assume a desired path in joint space is given by a twice differentiable function $\mathbf{q} = \mathbf{q}_d(t) \in C^2$. Hence, the desired time history of joints' position, velocity, and acceleration are known. We can control the robot to follow the desired path, by introducing a *computed torque control law* as below

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\ (\ddot{\mathbf{q}}_d - \mathbf{k}_D \dot{\mathbf{e}} - \mathbf{k}_P \mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \tag{15.34}$$

where $\mathbf{e}$ is the error vector

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \tag{15.35}$$

and $\mathbf{k}_D$ and $\mathbf{k}_P$ are constant gain diagonal matrices. The control law is stable and applied as long as all the eigenvalues of the following matrix have negative real part.

$$[A] = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{k}_P & -\mathbf{k}_D \end{bmatrix} \tag{15.36}$$

Proof. The required $\mathbf{Q}_c$ to track $\mathbf{q}_d(t)$ can directly be found by substituting the path function into the equations of motion.

$$\mathbf{Q}_c = \mathbf{D}(\mathbf{q}_d)\, \ddot{\mathbf{q}}_d + \mathbf{H}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \mathbf{G}(\mathbf{q}_d) \tag{15.37}$$

The calculated torques are called *control inputs*, and the control is based on the *open-loop control law.* In an open-loop control, we have the equations of motion for a robot and we need the required torques to move the robot on a given path. Open-loop control is a blind control method, since the current state of the robot is not used for calculating the inputs.

Due to non-modeled parameters and also errors in adjustment, there is always a difference between the desired and actual paths. To make the robot's actual path converge to the desired path, we must introduce a feedback control. Let us use the feedback signal of the actual path and apply the computed torque control law (15.34) to the robot. Substituting the control law in the equations of motion (15.33), gives us

$$\ddot{\mathbf{e}} + \mathbf{k}_D \dot{\mathbf{e}} + \mathbf{k}_P \mathbf{e} = 0. \tag{15.38}$$

This is a linear differential equation for the error variable between the actual and desired outputs. If the $n \times n$ gain matrices $\mathbf{k}_D$ and $\mathbf{k}_P$ are assumed to be diagonal, then we may rewrite the error equation in a matrix form.

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{k}_P & -\mathbf{k}_D \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = [A] \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} \tag{15.39}$$

The linear differential equation (15.39) is asymptotically stable when all the eigenvalues of $[A]$ have negative real part. The matrix $\mathbf{k}_P$ has the role of natural frequency, and $\mathbf{k}_D$ acts as damping.

$$\mathbf{k}_P = \begin{bmatrix} \omega_1^2 & 0 & 0 & 0 \\ 0 & \omega_2^2 & 0 & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \omega_n^2 \end{bmatrix} \tag{15.40}$$

$$\mathbf{k}_D = \begin{bmatrix} 2\xi_1\omega_1 & 0 & 0 & 0 \\ 0 & 2\xi_2\omega_2 & 0 & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 2\xi_n\omega_n \end{bmatrix} \tag{15.41}$$

Since $\mathbf{k}_D$ and $\mathbf{k}_P$ are diagonal, we can adjust the gain matrices $\mathbf{k}_D$ and $\mathbf{k}_P$ to control the response speed of the robot at each joint independently. A simple choice for the matrices is to set $\xi_i = 0, i = 1, 2, \cdots, n$, and make each joint response equal to the response of a critically damped linear second order system with natural frequency ω_i.

The computed torque control law (15.34) has two components as shown below.

$$\mathbf{Q} = \underbrace{\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})}_{\mathbf{Q}_{ff}} + \underbrace{\mathbf{D}(\mathbf{q})\left(-\mathbf{k}_D\dot{\mathbf{e}} - \mathbf{k}_P\mathbf{e}\right)}_{\mathbf{Q}_{fb}} \tag{15.42}$$

The first term, $\mathbf{Q}_{ff}$, is the *feedforward* command, which is the required torques based on open-loop control law. When there is no error, the control input $\mathbf{Q}_{ff}$ makes the robot follow the desired path $\mathbf{q}_d$. The second term, $\mathbf{Q}_{fb}$, is the *feedback* command, which is the correction torques to reduce the errors in the path of the robot.

Computed torque control is also called *feedback linearization*, which is an applied technique for robots' nonlinear control design. To apply the feedback linearization technique, we develop a control law to eliminate all nonlinearities and reduce the problem to the linear second-order equation of error signal (15.38) ■

Example 392 *Computed force control for an oscillator.*

Figure 15.2 depicts a linear mass-spring-damper oscillator under the action of a control force. The equation of motion for the oscillator is

$$m\ddot{x} + c\dot{x} + kx = f. \tag{15.43}$$

Applying a computed force control law

$$f = m\left(\ddot{x}_d - k_D\dot{e} - k_P e\right) + c\dot{x} + kx \tag{15.44}$$

$$e = x - x_d \tag{15.45}$$

reduces the error differential equation to

$$\ddot{e} + k_D\dot{e} + k_P e = 0. \tag{15.46}$$

The solution of the error equation is

$$e = Ae^{\lambda_1 t} + Be^{\lambda_2 t} \tag{15.47}$$

$$\lambda_{1,2} = -k_D \pm \sqrt{k_D^2 - 4k_P} \tag{15.48}$$

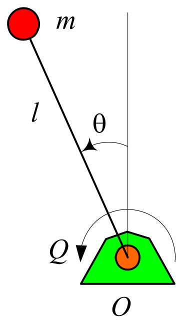

FIGURE 15.4. A controlled inverted pendulum.

where A and B are functions of initial conditions, and $\lambda_{1,2}$ are solutions of the characteristic equation

$$m\lambda^2 + k_D\lambda + k_P = 0. \tag{15.49}$$

The solution (15.47) is stable and $e \to 0$ exponentially as $t \to \infty$ if $k_D > 0$.

Example 393 *Inverted pendulum.*

Consider an inverted pendulum shown in Figure 15.4. Its equation of motion is

$$ml^2\ddot{\theta} - mgl\sin\theta = Q. \tag{15.50}$$

To control the pendulum and bring it from an initial angle $\theta = \theta_0$ to the vertical-up position, we may employ a feedback control law as

$$Q = -k_D\dot{\theta} - k_P\theta - mgl\sin\theta. \tag{15.51}$$

The parameters k_D and k_P are positive gains and are assumed constants. The control law (15.51) transforms the dynamics of the system to

$$ml^2\ddot{\theta} + k_D\dot{\theta} + k_P\theta = 0 \tag{15.52}$$

showing that the system behaves as a stable mass-spring-damper.

In case the desired position of the pendulum is at a nonzero angle, $\theta = \theta_d$, we may employ a feedback control law based on the error $e = \theta - \theta_d$ as below,

$$Q = ml^2\ddot{\theta}_d - k_D\dot{e} - k_Pe - mgl\sin\theta. \tag{15.53}$$

Substituting this control law in the equation of motion (15.50) shows that the dynamic of the controlled system is governed by

$$ml^2\ddot{e} + k_D\dot{e} + k_Pe = 0. \tag{15.54}$$

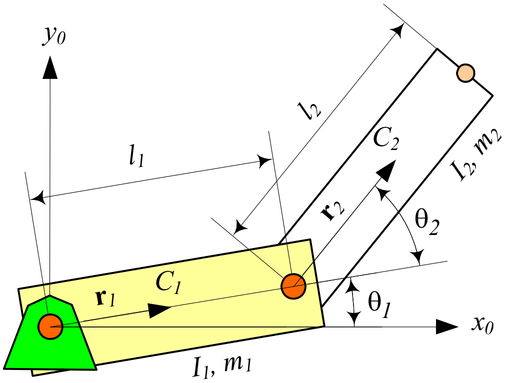

FIGURE 15.5. A 2R planar manipulator with massive links.

Example 394 *Control of a 2R planar manipulator.*

A 2R planar manipulator is shown in Figure 15.5 with dynamic equations given below.

$$\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{15.55}$$

where

$$D_{11} = m_1 r_1^2 + I_1 + m_2 \left(l_1^2 + l_1 r_2 \cos\theta_2 + r_2^2 \right) + I_2 \tag{15.56}$$
$$D_{21} = D_{12} = m_2 l_1 r_2 \cos\theta_2 + m_2 r_2^2 + I_2 \tag{15.57}$$
$$D_{22} = m^2 r_2^2 + I_2 \tag{15.58}$$

$$C_{11} = -m_2 l_1 r_2 \dot{\theta}_2 \sin\theta_2 \tag{15.59}$$
$$C_{21} = -m_2 l_1 r_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin\theta_2 \tag{15.60}$$
$$C_{12} = m_2 l_1 r_2 \dot{\theta}_1 \sin\theta_2 \tag{15.61}$$
$$C_{22} = 0 \tag{15.62}$$

$$G_1 = m_1 g r_1 \cos\theta_1 + m_2 g \left(l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2) \right) \tag{15.63}$$
$$G_2 = m_2 g r_2 \cos(\theta_1 + \theta_2) . \tag{15.64}$$

Let's write the equations of motion in the following form:

$$\mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{Q} \tag{15.65}$$

and multiply both sides by $\mathbf{D}^{-1}$ *to transform the equations of motion to*

$$\ddot{\mathbf{q}} + \mathbf{D}^{-1}\mathbf{C}\dot{\mathbf{q}} + \mathbf{D}^{-1}\mathbf{G} = \mathbf{D}^{-1}\mathbf{Q}. \tag{15.66}$$

To control the manipulator to follow a desired path $\mathbf{q} = \mathbf{q}_d(t)$, *we apply the following control law:*

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\,\mathbf{U} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{15.67}$$

where

$$\mathbf{U} = \ddot{\mathbf{q}}_d - 2k\dot{\mathbf{e}} - k^2\mathbf{e} \tag{15.68}$$

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d. \tag{15.69}$$

The vector $\mathbf{U}$ *is the controller input,* $\mathbf{e}$ *is the position error, and* k *is a positive constant gain number. Substituting the control law into the equation of motion shows that the error vector satisfies a linear second-order ordinary differential equation*

$$\ddot{\mathbf{e}} + 2k\dot{\mathbf{e}} + k^2\mathbf{e} = 0 \tag{15.70}$$

and therefore, exponentially converges to zero.

15.3 Linear Control Technique

Linearization of a robot's equations of motion about an operating point while applying a linear control algorithm is an old practical robot control method. This technique works well in a vicinity of the operating point. Hence, it is only a locally stable method. The linear control techniques are proportional, integral, derivative, and any combination of them.

The idea is to linearize the nonlinear equations of motion about some reference operating points to make a linear system, design a controller for the linear system, and then, apply the control to the robot. This technique will always result in a stable controller in some neighborhood of the operating point. However, the stable neighborhood may be quite small and hard to be determined.

A proportional-integral-derivative (PID) control algorithm employs a position error, derivative error, and integral error to develop a control law. Hence, a PID control law has the following general form for the input command:

$$Q = k_P e + k_I \int_0^t e\,dt + k_D \dot{e} \tag{15.71}$$

where $e = q - q_d$ is the error signal, and k_P, k_I, and k_D are positive constant gains associated to the proportional, integral, and derivative controllers.

The control command Q is thus a sum of three terms: the P-term, which is proportional to error e, the I-term, which is proportional to the integral of the error, and the D-term, which is proportional to the derivative of the error.

15.3.1 Proportional Control

In the case of proportional control, the *PID* control law (15.71) reduces to

$$Q = k_P e + Q_d. \tag{15.72}$$

The variable Q_d is the desired control command, which is called a *bias* or *reset factor*. When the error signal is zero, the control command is equal to the desired value. The proportional control has a drawback that results in a constant error at steady state condition.

15.3.2 Integral Control

The main function of an integral control is to eliminate the steady state error and make the system follow the set point at steady state conditions. The integral controller leads to an increasing control command for a positive error, and a decreasing control command for a negative error. An integral controller is usually used with a proportional controller. The control law for a PI controller is

$$Q = k_P e + k_I \int_0^t e\, dt. \tag{15.73}$$

15.3.3 Derivative Control

The purpose of derivative control is to improve the closed-loop stability of a system. A derivative controller has a predicting action by extrapolating the error using a tangent to the error curve. A derivative controller is usually used with a proportional controller. The *PD* control law is

$$Q = k_P e + k_D \dot{e}. \tag{15.74}$$

Proof. Any linear system behaves linearly if it is sufficiently near a reference operating point. Consider a nonlinear system

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{Q}) \tag{15.75}$$

where $\mathbf{q}_d$ is a solution generated by a specific input $\mathbf{Q}_c$

$$\dot{\mathbf{q}}_d = \mathbf{f}(\mathbf{q}_d, \mathbf{Q}_c). \tag{15.76}$$

Assume $\delta\mathbf{q}$ is a small change from the reference point $\mathbf{q}_d$ because of a small change $\delta\mathbf{Q}$ from $\mathbf{Q}$.

$$\mathbf{q} = \mathbf{q}_d + \delta\mathbf{q} \tag{15.77}$$

$$\mathbf{Q} = \mathbf{Q}_c + \delta\mathbf{Q} \tag{15.78}$$

If the changes $\delta\mathbf{q}$ and $\delta\mathbf{Q}$ are assumed small for all times, then the equation (15.75) can be approximated by its Taylor expansion and $\mathbf{q}$ be the solution of

$$\dot{\mathbf{q}} = \frac{\partial\mathbf{f}}{\partial\mathbf{q}_d}\mathbf{q} + \frac{\partial\mathbf{f}}{\partial\mathbf{Q}_d}\mathbf{Q}. \tag{15.79}$$

The partial derivative matrices $\left[\frac{\partial\mathbf{f}}{\partial\mathbf{q}_d}\right]$ and $\left[\frac{\partial\mathbf{f}}{\partial\mathbf{Q}_d}\right]$ are evaluated at the reference point $(\mathbf{q}_d, \mathbf{Q}_d)$. ■

Example 395 ★ *Linear control for a pendulum.*

Figure 11.12 illustrates a controlled pendulum as a one-arm manipulator. The equation of motion for the arm is

$$Q = I\,\ddot{\theta} + c\dot{\theta} + mgl\sin\theta \tag{15.80}$$

where I is the arm's moment of inertia about the pivot joint and m is the mass of the arm. The joint has a viscous damping c and kinematic length, the distance between the pivot and C, is l. Introducing a new set of variables

$$\theta = x_1 \tag{15.81}$$
$$\dot{\theta} = x_2 \tag{15.82}$$

converts the equation of motion to

$$\dot{x}_1 = x_2 \tag{15.83}$$
$$\dot{x}_2 = \frac{Q - c\,x_2 - mgl\sin x_1}{I}. \tag{15.84}$$

The linearized form of these equations is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -mgl/I & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1/I \end{bmatrix} \begin{bmatrix} 0 \\ Q \end{bmatrix}. \tag{15.85}$$

Assume that the reference point is

$$\mathbf{x}_d = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \pi/2 \\ 0 \end{bmatrix} \tag{15.86}$$
$$Q_c = mgl. \tag{15.87}$$

The coefficient matrices in Equation (15.85) must then be evaluated at the reference point. We use a set of sample data

$$\begin{aligned} m &= 1\,\mathrm{kg} \\ l &= 0.35\,\mathrm{m} \\ I &= 0.07\,\mathrm{kg.\,m^2} \\ c &= 0.01\,\mathrm{N\,s/\,m} \end{aligned} \tag{15.88}$$

and find

$$\frac{\partial \mathbf{f}}{\partial \mathbf{q}_d} = \begin{bmatrix} 0 & 1 \\ -49.05 & -0.01 \end{bmatrix} \tag{15.89}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{Q}_c} = \begin{bmatrix} 0 & 0 \\ 0 & 14.286 \end{bmatrix}. \tag{15.90}$$

Now we have a linear system and we may apply any control law that applies to linear systems. For instance, a PID control law

$$\mathbf{Q} = \mathbf{Q}_c - \mathbf{k}_D \dot{\mathbf{e}} - \mathbf{k}_P \mathbf{e} + \mathbf{k}_I \int_0^t \mathbf{e}\, dt \tag{15.91}$$

where

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d = \begin{bmatrix} x_1 - \pi/2 \\ x_2 \end{bmatrix} \tag{15.92}$$

can control the arm around the reference point.

Example 396 *PD control.*

Let us define a PD control law as

$$\mathbf{Q} = -\mathbf{k}_D \dot{\mathbf{e}} - \mathbf{k}_P \mathbf{e} \tag{15.93}$$

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \tag{15.94}$$

Applying the PD control to a robot with dynamic equations as

$$\begin{aligned} \mathbf{Q} &= \mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \\ &= \mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \end{aligned} \tag{15.95}$$

will produce the following control equation:

$$\mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{k}_D\,(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) - \mathbf{k}_P\,(\mathbf{q} - \mathbf{q}_d) = 0. \tag{15.96}$$

This control is ideal when $\mathbf{q}_d$ *is a constant vector associated with a specific configuration of a robot, and therefore* $\dot{\mathbf{q}}_d = 0$*. In this case the PD controller can make the configuration* $\mathbf{q}_d$ *globally stable.*

In case of a path given by $\mathbf{q} = \mathbf{q}_d(t)$*, we define a modified PD controller in the form*

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}) - \mathbf{k}_D \dot{\mathbf{e}} - \mathbf{k}_P \mathbf{e} \tag{15.97}$$

and reduce the closed-loop equation to

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{e}} + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{k}_D)\,\dot{\mathbf{e}} + \mathbf{k}_P \mathbf{e} = 0. \tag{15.98}$$

The linearization of this equation about a control point $\mathbf{q} = \mathbf{q}_d = const$ *provides a stable dynamics for the error signal*

$$\mathbf{D}(\mathbf{q}_d)\ddot{\mathbf{e}} + \mathbf{k}_D \dot{\mathbf{e}} + \mathbf{k}_P \mathbf{e} = 0. \tag{15.99}$$

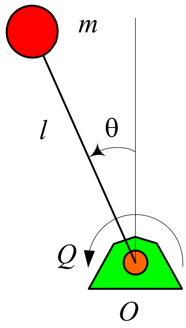

FIGURE 15.6. An inverted pendulum.

15.4 Sensing and Control

Position, velocity, acceleration, and force sensors are the most common sensors used in robotics. Consider the inverted pendulum shown in Figure 15.6 as a one DOF manipulator with the following equation of motion:

$$ml^2\ddot{\theta} - c\dot{\theta} - mgl\sin\theta = Q. \tag{15.100}$$

From an open-loop control viewpoint, we need to provide a moment $Q_c(t)$ to force the manipulator to follow a desired path of motion $\theta_d(t)$ where

$$Q_c = ml^2\ddot{\theta}_d - c\dot{\theta}_d - mgl\sin\theta_d. \tag{15.101}$$

In robotics, we usually calculate Q_c from the dynamics equation and dictate it to the actuator.

The manipulator will respond to the applied moment and will move. The equation of motion (15.100) is a model of the actual manipulator. In other words, we want the manipulator to work based on this equation. However, there are so many unmodeled phenomena that we cannot include them in our equation of motion, or we cannot model them. Some are temperature, air pressure, exact gravitational acceleration, or even the physical parameters such as m and l that we think have good accuracy. So, applying a control command Q_c will move the manipulator and provide a real value for θ, $\dot{\theta}$, and $\ddot{\theta}$, which are not necessarily equal to θ_d, $\dot{\theta}_d$, and $\ddot{\theta}_d$. Sensing is now important because we need to measure the actual angle θ, angular velocity $\dot{\theta}$, and angular acceleration $\ddot{\theta}$ to compare them with θ_d, $\dot{\theta}_d$, and $\ddot{\theta}_d$ and make sure that the manipulator is following the desired path. This is the reason why the feedback control systems and the error signal $e = \theta - \theta_d$ were introduced.

Robots are supposed to do a job in an environment, so they can interact with the environment. Therefore, a robot needs two types of sensors: 1-sensing the robot's internal parameters, which are called *proprioceptors*, and 2-sensing the robot's environmental parameters, which are called *exteroceptors*. The most important interior parameters are position, velocity, acceleration, force, torque, and inertia.

15.4.1 Position Sensors

Rotary encoders. In robotics, almost all kinds of actuators provide a rotary motion. Then we may provide a rotation motion for a revolute joint, or a translation motion for a prismatic joint, by using gears. So, it is ideally possible to sense the relative position of the links connected by the joint based on the angular position of the actuator. The error in this position sensing is due to non-rigidity and backlash. The most common position sensor is a *rotary encoder* that can be *optical, magnetic*, or *electrical.* As an example, when the encoder shaft rotates, a disk counting a pattern of fine lines interrupts a light beam. A photodetector converts the light pulses into a countable binary waveform. The shaft angle is then determined by counting the number of pulses.

Resolvers. We may design an electronic device to provide a mathematical function of the joint variable. The mathematical function might be sine, cosine, exponential, or any combination of mathematical functions. The joint variable is then calculated indirectly by resolving the mathematical functions. Sine and cosine functions are more common.

Potentiometers. Using an electrical bridge, the potentiometers can provide an electric voltage proportional to the joint position.

LVDT and RVDT. LVDT/RVDT or a Linear/Rotary Variable Differential Transformer operates with two transformers sharing the same magnetic core. When the core moves, the output of one transformer increases while the other's output decreases. The difference of the current is a measure of the core position.

15.4.2 Speed Sensors

Tachometers. Generally speaking, a tachometer is a name for any velocity sensor. Tachometers usually provide an analog signal proportional to the angular velocity of a shaft. There are a vast amount of different designs for tachometers, using different physical characteristics such as magnetic field.

Rotary encoders. Any rotary sensor can be equipped with a time measuring system and become an angular velocity sensor. The encoder counts the light pulses of a rotating disk and the angular velocity is then determined by time between pulses.

Differentiating devices. Any kind of position sensor can be equipped with a digital differentiating device to become a speed sensor. The digital or numerical differentiating needs a simple processor. Numerical differentiating is generally an erroneous process.

Integrating devices. The output signal of an accelerometer can be numerically integrated to provide a velocity signal. The digital or numerical integrating also needs a simple processor. Numerical differentiating is generally a smooth and reliable process.

15.4.3 Acceleration Sensors

Acceleration sensors work based on Newton's second law of motion. They sense the force that causes an acceleration of a known mass. There are many types of accelerometers. Stress-strain gage, capacitive, inductive, piezoelectric, and micro-accelerometers are the most common. In any of these types, force causes a proportional displacement in an elastic material, such as deflection in a micro-cantilever beam, and the displacement is proportional to the acceleration.

Applications of accelerometers include measurement of acceleration, angular acceleration, velocity, position, angular velocity, frequency, impulse, force, tilt, and orientation.

Force and Torque Sensors. Any concept and method that we use in sensing acceleration may also be used in force and torque sensing. We equip the wrists of a robot with at least three force sensors to measure the contact forces and moments with the environment. The wrist's force sensors are important especially when the robot's job is involved with touching unknown surfaces and objects.

Proximity Sensors. Proximity sensors are utilized to detect the existence of an object, field, or special material before interacting with it. Inductive, capacitive, Hall effect, sonic, ultrasonic, and optical are the most common proximity sensors.

The inductive sensors can sense the existence of a metallic object due to a change in inductance. The capacitive sensors can sense the existence of gas, liquid, or metals that cause a change in capacitance. Hall effective sensors work based on the interaction between the voltage in a semiconductor material and magnetic fields. These sensors can detect the existence of magnetic fields and materials. Sonic, ultrasonic, and optical sensors work based on the reflection or modification in an emitted signal by objects.

15.5 Summary

In an open-loop control algorithm, we calculate the robot's required torque commands $\mathbf{Q}_c$ for a given joint path $\mathbf{q}_d = \mathbf{q}(t)$ based on the equations of motion

$$\mathbf{Q}_c = \mathbf{D}(\mathbf{q}_d)\,\ddot{\mathbf{q}}_d + \mathbf{H}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \mathbf{G}(\mathbf{q}_d). \tag{15.102}$$

However, there can be a difference between the actual joint variables and the desired values. The difference is called error $\mathbf{e}$

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \tag{15.103}$$
$$\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d. \tag{15.104}$$

By measuring the error command, we may define a control law and calculate a new control command vector

$$\mathbf{Q} = \mathbf{Q}_c + \mathbf{k}_D\dot{\mathbf{e}} + \mathbf{k}_P\mathbf{e} \tag{15.105}$$

to compensate for the error. The parameters $\mathbf{k}_P$ and $\mathbf{k}_D$ are constant gain diagonal matrices.

The control law compares the actual joint variables $(\mathbf{q}, \dot{\mathbf{q}})$ with the desired values $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$, and generates a command proportionally. Applying the new control command changes the dynamic equations of the robot to

$$\mathbf{Q}_c + \mathbf{k}_D\dot{\mathbf{e}} + \mathbf{k}_P\mathbf{e} = \mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}). \tag{15.106}$$

This is a closed-loop control algorithm, in which the control commands are calculated based on the difference between actual and desired variables.

Computed torque control

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\ (\ddot{\mathbf{q}}_d - \mathbf{k}_D\dot{\mathbf{e}} - \mathbf{k}_P\mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \tag{15.107}$$

is an applied closed-loop control law in robotics to make a robot follow a desired path.

15.6 Key Symbols

a, b	real and imaginary parts of a complex number
A	coefficient matrix
A, B	real coefficients
B	body coordinate frame
c	damping
C_i	complex coefficients
$\mathbf{e}$	error, exponential function
$f_c, \mathbf{f}_c$	actuator force control command
$f, \mathbf{f}$	actual force command
g	gravitational acceleration
G, B_0	global coordinate frame, Base coordinate frame
i	imaginary unit number
$\mathbf{I} = [I]$	identity matrix, moment of inertia
J	Jacobian
k	stiffness
$\mathbf{k}_P$	proportional constant control gain
$\mathbf{k}_D$	derivative constant control gain
l	length
m	mass
$\mathbf{q}$	actual vector of joint variables
$\mathbf{q}_d$	desired path of joint
$\mathbf{Q}$	actuators' actual command
$\mathbf{Q}_c$	actuators' control command
$\mathbf{Q}_{fb}$	feedback command
$\mathbf{Q}_{ff}$	feedforward command
$\mathbf{r}$	position vectors, homogeneous position vector
r_i	the element i of $\mathbf{r}$
t	time
x, y, z	local Cartesian coordinates
X, Y, Z	global Cartesian coordinates
Greek	
δ	small increment of a parameter
λ	characteristic value, eigenvalue
θ	rotary joint angle
ω_n	natural frequency
ξ	damping ratio
Symbol	
DOF	degree of freedom
$\mathbb{R}$	real numbers set
Re	real

Exercises

1. Response of second-order systems.

   Solve the characteristic equations and determine the response of the following second-order systems at $x(1)$, if they start from $x(0) = 1$, $\dot{x}(0) = 0$.

   (a)
   $$\ddot{x} + 2\dot{x} + 5x = 0$$

   (b)
   $$\ddot{x} + 2\dot{x} + x = 0$$

   (c)
   $$\ddot{x} + 4\dot{x} + x = 0$$

2. Modified PD control.

   Apply a modified PD control law

   $$\begin{aligned} f &= -k_P e - k_d \dot{x} \\ e &= x - x_d \end{aligned}$$

   to a second-order linear system

   $$m\ddot{x} + c\dot{x} + kx = f$$

   and reduce the system to a second-order equation in an error signal.

   $$m\ddot{e} + (c + k_D)\dot{e} + (k + k_P)e = kx_D$$

   Then, calculate the steady state error for a step input

   $$x = x_d = const.$$

3. Modified PID control.

   Apply a modified PD control law

   $$\begin{aligned} f &= -k_P e - k_d \dot{x} - k_I \int_0^t e dt \\ e &= x - x_d \end{aligned}$$

   to a second-order linear system

   $$m\ddot{x} + c\dot{x} + kx = f$$

and reduce the system to a third-order equation in an error signal.

$$m\dddot{e} + (c + k_D)\ddot{e} + (k + k_P)\dot{e} + k_I e = 0$$

Then, find the PID gains such that the characteristic equation of the system simplifies to

$$\left(\lambda^2 + 2\xi\omega_n\lambda + \omega_n^2\right)(\lambda + \beta) = 0.$$

4. Linearization.

   Linearize the given equations and determine the stability of the linearized set of equations.

$$\begin{aligned} \dot{x}_1 &= x_2^2 + x_1 \cos x_2 \\ \dot{x}_2 &= x_2 + (1 + x_1 + x_2)x_1 + x_1 \sin x_2 \end{aligned}$$

5. Expand the control equations for a $2R$ planar manipulator using the following control law:

$$\mathbf{Q} = \mathbf{D}(\mathbf{q})\,(\ddot{\mathbf{q}}_d - \mathbf{k}_D\dot{\mathbf{e}} - \mathbf{k}_P\mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})$$

6. One-link manipulator control.

   A one-link manipulator is shown in Figure 15.6.

   (a) Derive the equation of motion.

   (b) Determine a rest-to-rest joint path between $\theta(0) = 45\,\text{deg}$ and $\theta(0) = -45\,\text{deg}$.

   (c) Solve the time optimal control of the manipulator and determine the torque $Q_c(t)$ for

$$\begin{aligned} m &= 1\,\text{kg} \\ l &= 1\,\text{m} \\ |Q| &\le 120\,\text{N m}. \end{aligned}$$

   (d) Now assume the mass is $m = 1.01\,\text{kg}$ and solve the equation of motion numerically by feeding the calculated torques $Q_c(t)$. Determine the position and velocity errors at the end of the motion.

   (e) Design a computed torque control law to compensate the error during the motion.

7. ★ Mass-spring control.

   Solve Exercise 14.10 and calculate the optimal control input. Increase the stiffness %10, and design a computed torque control law to eliminate error during the motion.

8. ★ $2R$ manipulator control.

   (a) Solve Exercise 13.20 and calculate the optimal control inputs.

   (b) Increase the masses by 10%, and solve the dynamic equations numerically.

   (c) Determine the position and velocity error in Cartesian and joint spaces by applying the calculated optimal inputs.

   (d) Design a computed torque control law to eliminate error during the motion.

9. ★ PR planar manipulator control.

   (a) Solve Exercise 14.13 and calculate the optimal control inputs.

   (b) Increase the gravitational acceleration by 10%, and solve the dynamic equations numerically.

   (c) Determine the position and velocity error in Cartesian and joint spaces by applying the calculated optimal inputs.

   (d) Design a computed torque control law to eliminate error during the motion.

10. Sensing and measurement.

    Consider the one DOF manipulator in Figure (15.100). To control the manipulator, we need to sense the actual angle θ, angular velocity $\dot{\theta}$, and angular acceleration $\ddot{\theta}$ and compare them with θ_d, $\dot{\theta}_d$, and $\ddot{\theta}_d$ to make sure that the manipulator is following the desired path. Can we measure the actual moment Q, that the actuator is providing, and compare with the predicted value Q_c instead? Does making Q equal to Q_c guarantee that the manipulator does what it is supposed to do?