Nonlinear Modular 3D Trajectory Control of a General Aviation Aircraft

Simon P. Schatz and Florian Holzapfel

1 Introduction

In the past decades, the use of fly-by-wire (FBW) became popular as a concept for flight controls. Nowadays, the FBW concept is becoming standard for large transportation aircraft, but research projects of Part-23 general aviation aircraft are also directed towards FBW concepts. In some cases, full FBW is of interest, in some cases the mechanical flight control system is altered by interconnected actuators of (sometimes) lower authority. Especially the emergence of unmanned aerial vehicles (UAV) and electrical flying further led to multiple applications of FBW systems, which incorporate multicopters of small MTOW as well as platforms of several tons.

Both manned and unmanned sensing platforms require waypoint based navigation to complete their missions and as a consequence, trajectory generation and control plays a significant role for these systems. In this paper, the trajectory controller of a fully integrated auto-flight control system is considered. At the institute, a modular control structure was developed in the context of adopting various components of the guidance and control structure to different platforms without the need of redesigning the controllers for the specific applications. Consequently, a range of modules were designed, which range from classical autopilot concepts to trajectory generation and automatic landing as well as to automatic aerodynamic trimming. All of these components may vary for different applications such as UAVs with unstable aircraft dynamics or classical autopilots for manned flight, but the rework can be decreased significantly due to the modular structure.

S.P. Schatz (✉) · F. Holzapfel
Institute of Flight System Dynamics, Technische Universität München,
Boltzmannstr. 15, 85748 Garching Bei München, Germany
e-mail: simon.p.schatz@tum.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_9

The trajectory control module is designed as a controller to drive any deviations in the lateral and vertical plane to zero and hence, is a candidate to be utilized by multiple higher-level functions such as waypoint navigation, automatic takeoff and landing, or altitude hold modes. From the controller's point of view, the deviations and their derivatives are used as control errors whereas other information about the trajectory such as trajectory angles and their rates are further utilized to improve the controller's performance, especially in cases of dynamical trajectories.

Previous work conducted in the field of trajectory control usually rely on multiple loops to address the problem of trajectory tracking. Commonly, there are two outer loops, which first compute a track / flight path angle command due to deviations and then compute inner loop commands to control the angular error. In this field, work has been conducted ranging from linear controllers to nonlinear methods such as backstepping or nonlinear dynamic inversion. Examples of such methods can be found in [1–10]. In this paper, a second-order approach for the outer loop is presented, which only consists of one control loop using both the position deviations in the lateral and vertical plane as well as their derivatives as control errors. This reduces the required time scale separation as there are just two second order loops used in the modular structure: trajectory controller and inner loop controller. The trajectory controller's design is based on the method of nonlinear dynamic inversion (NDI, see e.g. [11, 12]), which inherently leads to utilizing the feedforward terms contained in the dynamical trajectory, along with a PD controller.

In this paper, previous work [13, 14] is extended, specifically the second order error dynamics of the lateral and vertical position deviations, which are the basis for the derivation of the controller, are addressed in detail and the dynamic inversion applied to these error dynamics is derived for the 3D controller. Additionally, the command transformation to the inner loop controller is derived and further information is given regarding the trajectory control and the environment. Ultimately, specific simulation results are presented to show the controller's lateral tracking performance in comparison to a linear controller and in presence of significant wind as additional test data to the first principal flight test demonstration published in Ref. [14].

The remainder of this paper is arranged as follows: Section 1.1 states the nomenclature used in this paper. The environment of the trajectory controller is presented in Sect. 2, where the trajectory generation (Sect. 2.1), the command transformation (Sect. 2.2), and the inner loop controller (Sect. 2.3) are discussed in more detail. Section 3 deals with the trajectory control design, specifically the discussion of the kinematic equations of motion, the derivation of the nonlinear error dynamics, its dynamic inversion, and the control design. In Sect. 4 simulation results are given to show the controller's performance in presence of wind and also by comparison to a linear PD controller without feedforward terms. Finally, concluding remarks are given in Sect. 5.

1.1 Nomenclature

In this paper, mainly vectors, which are noted bold, and scalars, noted in italic letters, are used. Furthermore, $\overrightarrow{(\cdot)}$ indicates that a vector is an element of the Euclidean space $\mathbb{R}^3$. In addition, Euclidean vectors considered in this paper are the following:

- $\left(\overrightarrow{\mathbf{r}}^{P}\right)_B$ denotes a position vector at the point P given in the body frame.
- $\left(\overrightarrow{\mathbf{V}}_K^{P}\right)^E$ denotes a kinematic velocity vector at the point P differentiated with respect to the E frame.
- $\left(\overrightarrow{\boldsymbol{\omega}}^{OK}\right)$ denotes an angular rate of the K frame with respect to the O frame.

All vector variables used in this paper can be compounded based on the concept introduced by these three examples.

2 Control Environment

The Institute of Flight System Dynamics owns a Diamond Aircraft Industries DA42 NG aircraft, which is adapted to serve as flying testbed. For this purpose, the aircraft has been modified to be controlled using a fly-by-wire system, which gears into the existing mechanical flight control system utilizing both an electrical and an overload clutch. The testbed is equipped with a safety system, which enables the safety pilot to either open the clutches by pushing a disengage button or by application of a sufficiently large stick force.

Figure 1 (taken from [13, 14] for clarity) shows the overall modular structure of the integrated autoflight system. The modules of the integrated auto-flight system range from the input monitoring and system automation part [15], where the incoming signals of sensors and ground station are evaluated and the overall operating modes are computed based on the result, to the actuator control electronics (ACEs) and the engine control unit (ECU).

The modular concept was chosen to keep certain parts of identical structure throughout different platforms. For this, certain interfaces are maintained throughout various projects. As an example, the control structure of the autopilot persists over several platforms and just requires adaptation of certain parameters such as gains and limits to the dynamics of the aircraft. The inner loop module, however, may differ significantly over different applications as it depends fully on aircraft dynamics and control allocation.

The trajectory controller is a part of the auto-flight-system, which in this context denotes the autopilot functionalities such as heading hold or flight level change mode and the speed controller / auto-thrust. The auto-flight-system also incorporates energy protection and prioritization, limiting the speed and the flight path angle such that no adverse conditions occur. In context of vertical trajectory control or upon application of the altitude hold mode, along with deviations also the permissible flight path angles are forwarded to the trajectory controller, which are then used to additionally

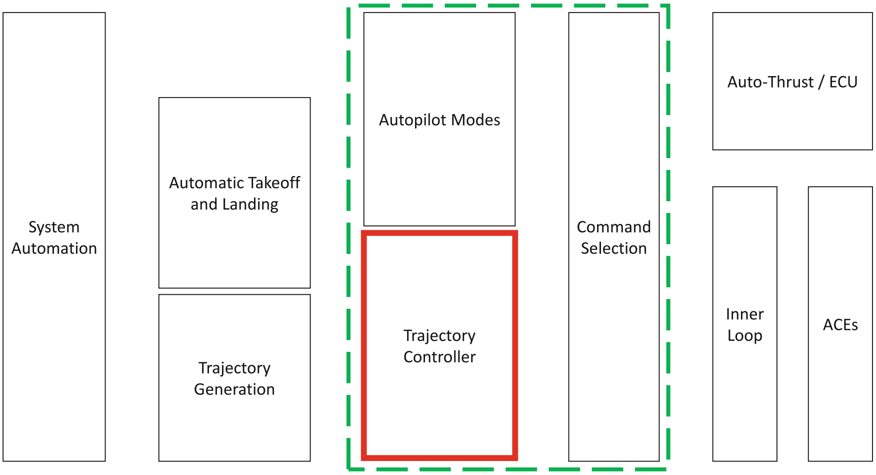

Fig. 1 Integrated autoflight system

limit the trajectory controller's specific force commands. For more details on the auto-flight-system refer to [16, 17].

The commands of the auto-flight-system and the trajectory controller, which are further dedicated as outer loop, are selected and forwarded to the inner loop controller. While the outer loops provide normalized specific force commands in the kinematic frame with respect to unaccelerated flight, these commands are transformed by a configuration-specific module, which is further depicted in Sect. 2.2, to the inputs required by the corresponding inner loop, which then computes the final commands to the ACEs and ECUs. The inner loop of the DA42 is briefly discussed in Sect. 2.3.

The trajectory controller deals with any deviations as long as they comply with the interface described in Sect. 2.1. In the current implementation for the DA42, trajectories and the corresponding deviations may either be provided by the automatic takeoff and landing module or a trajectory generation module [18, 19], which is based on waypoints. The latter is briefly discussed in Sect. 2.1. Furthermore, the trajectory controller is used to drive the altitude deviation to zero when the altitude hold mode is applied.

2.1 Trajectory Generation

The paper deals with a modular trajectory controller, which is designed to drive the deviations in a trajectory frame to zero. From the controller's point of view, any source of deviations and additional trajectory information is permitted. In this paper, fly-by trajectories generated by the trajectory generation module are considered. Here, the interface to this module is briefly introduced.

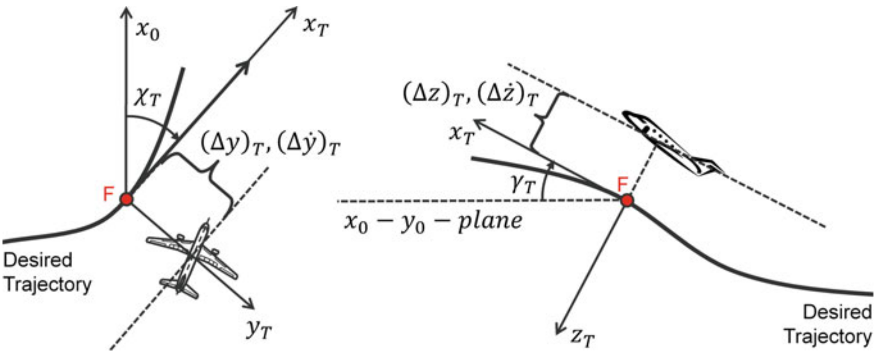

Fig. 2 Trajectory frame with deviations and their corresponding time derivatives

For this purpose, the trajectory frame T, which is related to the kinematic frame K as commonly used in aerospace applications, is introduced as follows:

- The x-axis is aligned with and in the direction of the desired velocity at the footpoint on the trajectory.
- The z-axis points downwards parallel to the projection of the local normal to the WGS84 ellipsoid into a plane perpendicular to the x-axis.
- The y-axis points parallel to the earth's surface and yields a right hand system with the previous two axes.

This frame is further illustrated in Fig. 2 (taken from [13, 14]), where the deviations $(\Delta y)_T$, $(\Delta z)_T \in \mathbb{R}$ and its time derivatives $(\Delta \dot{y})_T$, $(\Delta \dot{z})_T \in \mathbb{R}$ are also displayed.

The trajectory controller is driven by the following inputs:

- The deviations between aircraft and desired trajectory

$$\left(\vec{\mathbf{r}}^{FR}\right)_T = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_T \triangleq \begin{pmatrix} x^R - x^F \\ y^R - y^F \\ z^R - z^F \end{pmatrix}_T , \tag{1}$$

- the corresponding time derivatives of these deviations

$$\left(\vec{\mathbf{V}}_K^{FR}\right)_T^T = \begin{pmatrix} \Delta \dot{x} \\ \Delta \dot{y} \\ \Delta \dot{z} \end{pmatrix}_T^T \triangleq \begin{pmatrix} \dot{x}^R - \dot{x}^F \\ \dot{y}^R - \dot{y}^F \\ \dot{z}^R - \dot{z}^F \end{pmatrix}_T^T , \tag{2}$$

- the desired trajectory angles χ_T, γ_T,
- the desired trajectory angular rates $\dot{\chi}_T$, $\dot{\gamma}_T$,
- the desired trajectory angular accelerations $\ddot{\chi}_T$, $\ddot{\gamma}_T$,
- the desired kinematic acceleration at the trajectory footpoint $\dot{V}_K^F$.

Note that $\overrightarrow{\mathbf{r}}^{FR}$, $\overrightarrow{\mathbf{V}}_K^{FR} \in \mathbb{R}^3$ and all the other components are real valued.

The current state of the trajectory generation module, previously presented in [18, 19], which generates a trajectory online based on 3+1 given waypoints, employs waypoints of the type fly-by, fly-over, and loiter, where just fly-by trajectories and a loiter are considered in this paper. Note that the trajectory is only generated for three-dimensional trajectories not employing the x-position on the trajectory, i.e. the time component, as the airspeed is controlled independently by the auto-flight system to maintain the aerodynamic integrity of the system.

A fly-by trajectory is generated using the current "from" waypoint, the fly-by waypoint, and the next waypoint after the fly-by to generate a circle passing the fly-by waypoint. Note that the radii of the circle of the fly-by maneuvers and loiters are computed based on the current kinematic velocity, a wind estimation as well as the current and commanded airspeeds. The maximum expected kinematic velocity throughout the turn is used to calculate the radius by aiming for a turn rate of $5°/s$. In order to smoothly start the curve, the trajectory generation is currently based on clothoids [19–22] for fly-by maneuvers, which provide a linear devolution of the trajectory angular rate $\dot{\chi}_T$. As a consequence, however, the angular acceleration $\ddot{\chi}_T$ employs non-continuous characteristics and hence, the angular accelerations remain unused throughout this paper.

2.2 Outer - Inner Loop Command Interface

As mentioned earlier, the outer loops provide normalized specific force commands in the kinematic frame with respect to unaccelerated flight, i.e. curvature commands given by

$$\overrightarrow{\mathbf{u}}_{ol,nrm,K} = \begin{bmatrix} \frac{\Delta f_{x,cmd,K}}{g} & \frac{f_{y,cmd,K}}{g} & \frac{\Delta f_{z,cmd,K}}{g} \end{bmatrix}^{\mathrm{T}}, \qquad (3)$$

where g represents the acceleration due to gravity. In the following, the curvature commands are first transformed to the north-east-down (NED, O) frame, then converted to absolute normalized specific forces including gravity, and ultimately rotated about the heading Ψ as well as the pitch angle Θ, yielding

$$\overrightarrow{\mathbf{u}}_{ol,nrm,B'} = \mathbf{T}_{B'O}\left(\Psi, \Theta\right) \left(\mathbf{T}_{OK}\left(-\gamma_K, -\chi_K\right) \overrightarrow{\mathbf{u}}_{ol,nrm,K} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}_O \right), \qquad (4)$$

where $\mathbf{T} \in \mathbb{R}_i^{3\times 3}$ depict transformation matrices, γ_K is the kinematic flight path angle, and χ_K denotes the kinematic track / course angle. Finally, the command transformation to the body frame can be obtained by

$$\overrightarrow{\mathbf{u}}_{ol,nrm,B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix} \overrightarrow{\mathbf{u}}_{ol,nrm,B'}, \tag{5}$$

where Φ depicts the roll angle of the aircraft. Now, as no lateral acceleration in the body frame is desired for a coordinated turn, the second row of (5) is utilized to compute

$$0 = \frac{f_{y,cmd,B'}}{g}\cos(\Phi_{cmd}) + \frac{f_{z,,B'}}{g}\sin(\Phi_{cmd}), \tag{6}$$

ultimately yielding the roll angle command

$$\Phi_{cmd} = \text{atan}\left(-\frac{f_{y,cmd,B'}}{f_{z,cmd,B'}}\right). \tag{7}$$

Taking into account the second and third row of (5) and utilizing Φ_{curr}, we have

$$f_{y,cmd,B'} = -f_{z,cmd,B'}\tan(\Phi_{curr}) \tag{8}$$

$$\frac{f_{z,cmd,B}}{g} = -\frac{f_{y,cmd,B'}}{g}\sin(\Phi_{curr}) + \frac{f_{z,cmd,B'}}{g}\cos(\Phi_{curr}), \tag{9}$$

and the normal body load factor command in the longitudinal plane as commanded to the inner loop can be computed as

$$\frac{f_{z,cmd,B}}{g} = \frac{f_{z,cmd,B'}}{g\cos(\Phi_{curr})}, \tag{10}$$

where Φ_{curr} is currently chosen as the measurement of the roll angle, but can be altered by a small feedforward term based on the roll rate and the computational rate. The command transformation as given by Eqs. (8) and (10) is designed in this form to cope with the different time scales regarding longitudinal and lateral motion. Therefore, the given longitudinal command is computed using the current measurement of the roll angle as it takes more time to built up the angle rather than the longitudinal acceleration to maintain the appropriate curvature in both axes.

2.3 Inner Loop

As mentioned in Sect. 2.2, the inner loop comprises a normal body load factor command system in the longitudinal plane and a roll angle command system in the lateral plane. In case of the longitudinal plane, the normalized normal acceleration in the body frame and the pitch rate q are used for a PI controller. The lateral inner loop comprises feedback of roll and yaw rates, p and r, respectively, along with the normalized body lateral acceleration and the roll angle for a MIMO control architecture.

Since the control design of the inner loop is based on linear approaches such as LQR [23] for the longitudinal plane and eigenstructure assignment [24] for the lateral plane, gain scheduling is required to compose the controller over the whole envelope. In order to increase the validity of the linear controller and incorporate turn compensation, the feedback signals are treated to achieve smaller errors as inputs to the inner loop controller. As a consequence, the feedback signals as seen from the inner loop are given by

$$\frac{\delta f_{z,B}}{g} = \frac{f_{z,B}}{g} + \frac{\cos\left(\Theta\right)}{\cos\left(\Phi\right)}, \tag{11}$$

$$\delta p = p - \frac{g}{V_K^R}\tan\left(\Phi\right), \tag{12}$$

$$\delta q = q - \frac{g}{V_K^R}\tan\left(\Phi\right)\cos\left(\Theta\right)\sin\left(\Phi\right), \tag{13}$$

$$\delta r = r - \frac{g}{V_K^R}\sin\left(\Phi\right)\cos\left(\Theta\right), \tag{14}$$

where V_K^R denotes the absolute kinematic velocity of the reference point and $\tan\left(\mu_K\right)$ is approximated by $\tan\left(\Phi\right)$ for turn compensation as the kinematic bank angle μ_K is not available for measurement. In order to remain in the same magnitude, also the normal body load factor command is treated by means of

$$\frac{\delta f_{z,cmd,B}}{g} = \frac{f_{z,cmd,B}}{g} + \frac{\cos\left(\Theta\right)}{\cos\left(\Phi\right)}. \tag{15}$$

The current gains are chosen in such a way that the inherent dynamics of the DA42 are not changed significantly. In the lateral plane, the roll dynamics and the natural frequency of the dutch roll are maintained while the yaw damping is increased. Furthermore, the spiral pole is set to -1 for the whole envelope. In the longitudinal plane, the pitch damping is increased and the normal body load factor command system is implemented. The tracking performance is achieved by the feedforward part, while the feedback is designed primarily for disturbance rejection.

3 Trajectory Controller

In this section, the proposed trajectory controller is presented. Section 3.1 gives a short derivation of the kinematic equations of motion for the reference point of a rigid body aircraft. The error dynamics used for the trajectory controller are derived in Sect. 3.2. The nonlinear dynamic inversion controller, which is applied based on these error dynamics, is presented and analyzed in Sect. 3.3.

3.1 Kinematic Equations of Motion

This section deals with the kinematic equations of motion used for the path dynamics of an aircraft. For this purpose, the results known from e.g. [25, 26] are computed based on Newton's Second Axiom. For the usage addressed in this paper, the reference point R of a rigid body aircraft is considered.

Assumption 1 The aircraft is assumed to be a rigid body. As a consequence, the relative position between points on the aircraft does not change over time, resulting in

$$\left(\vec{\mathbf{V}}_K^{RP}\right)^B = 0. \tag{16}$$

For computing the equations of motion, consider the position vector $\vec{\mathbf{r}}^P \in \mathbb{R}^3$

$$\left(\vec{\mathbf{r}}^P\right) = \left(\vec{\mathbf{r}}^R\right) + \left(\vec{\mathbf{r}}^{RP}\right), \tag{17}$$

and the corresponding velocities in the inertial frame given by

$$\left(\vec{\mathbf{V}}_K^R\right)^I = \left(\vec{\mathbf{V}}_K^R\right)^E + \left(\vec{\boldsymbol{\omega}}^{IE}\right) \times \left(\vec{\mathbf{r}}^R\right), \tag{18}$$

$$\left(\vec{\mathbf{V}}_K^{RP}\right)^I = \left(\vec{\boldsymbol{\omega}}^{IB}\right) \times \left(\vec{\mathbf{r}}^{RP}\right), \tag{19}$$

which are valid according to Assumption 1. In addition, the following assumption is made, which is fairly standard [25]:

Assumption 2 The influence of the mass flow onto the impulse of the aircraft can be neglected:

$$\dot{m}\left(\vec{\mathbf{V}}_K^P\right)^E \approx 0. \tag{20}$$

Differentiating the position vector (17) twice and incorporating the velocities (18), (19), and the Assumptions 1 and 2, Newton's Second Axiom can be given as

$$\sum\left(\vec{\mathbf{F}}^R\right) = m\left(\dot{\vec{\mathbf{V}}}_K^R\right)^{EO} + m\,\vec{\mathbf{a}}_{add}. \tag{21}$$

Remark 1 Note that $\vec{\mathbf{a}}_{add}$ is a term containing all additional accelerations such as the Coriolis term, the acceleration due to the transport rate, and the acceleration due to difference of reference point R and center of gravity G. When considering a flat, non-rotating earth and choosing the center of gravity as reference point, $\vec{\mathbf{a}}_{add} = \vec{\mathbf{0}}$. For reasons of brevity, $\vec{\mathbf{a}}_{add}$ is neglected in the subsequent sections.

Making use of the transformation into the kinematic frame

$$\left(\dot{\vec{\mathbf{V}}}_K^R\right)^{EO} = \left(\dot{\vec{\mathbf{V}}}_K^R\right)^{EK} + \left(\vec{\boldsymbol{\omega}}^{OK}\right) \times \left(\vec{\mathbf{V}}_K^R\right)^E, \tag{22}$$

the kinematic equations of motion can be computed as

$$\dot{V}_K^R = \Delta f_{x,K}, \tag{23}$$

$$\dot{\chi}_K = \frac{f_{y,K}}{V_K^R \cos\gamma_\mathrm{K}}, \tag{24}$$

$$\dot{\gamma}_K = \frac{-\Delta f_{z,K}}{V_K^R}, \tag{25}$$

where the specific forces are defined as

$$\Delta f_{x,K} \triangleq \frac{(X_A^R + X_P^R)_K}{m} - g\sin\gamma_\mathrm{K}, \tag{26}$$

$$f_{y,K} \triangleq \frac{(Y_A^R + Y_P^R)_K}{m}, \tag{27}$$

$$\Delta f_{z,K} \triangleq \frac{(Z_A^R + Z_P^R)_K}{m} + g\cos\gamma_\mathrm{K}, \tag{28}$$

where $m \in \mathbb{R}$ denotes the mass of the aircraft and $(X_A^R)_K$, $(Y_A^R)_K$, $(Z_A^R)_K \in \mathbb{R}$ as well as $(X_P^R)_K$, $(Y_P^R)_K$, $(Z_P^R)_K \in \mathbb{R}$ are the components of the aerodynamic and propulsion forces at the reference point, respectively, noted in the kinematic frame.

For the further derivation of the trajectory controller, the aircraft dynamics related to the trajectory are given by the kinematic Eqs. (23)–(25). Note that the forces affecting these dynamics require an inner loop controller supplying the respective actuator commands for these loads. In order to achieve this, the commands in the kinematic frame are transformed into roll angle and normal body load factor commands as discussed in Sect. 2.2, which are then utilized by the inner loop controller briefly introduced in Sect. 2.3, whereas the airspeed is controlled by a speed controller as introduced in [16, 17].

3.2 Nonlinear Error Dynamics

In this section, the nonlinear error dynamics between the reference point of the aircraft and a reference trajectory are computed. The relative deviation is defined in (1) and its time derivative is given by Eq. (2). This relative velocity can be further computed as

$$\left(\overrightarrow{\mathbf{V}}_K^{FR}\right)_T^T = \left(\overrightarrow{\mathbf{V}}_K^{FR}\right)_T^E + \left(\overrightarrow{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\overrightarrow{\mathbf{r}}^{FR}\right)_T , \tag{29}$$

where the angular rate is specified as

$$\left(\overrightarrow{\boldsymbol{\omega}}^{TO}\right)_T = \begin{pmatrix} \dot{\chi}_T \sin\gamma_T \\ -\dot{\gamma}_T \\ -\dot{\chi}_T \cos\gamma_T \end{pmatrix} . \tag{30}$$

As mentioned earlier, the controller is designed to use the specific forces as inputs to the inner loop. For this purpose, the error dynamics are differentiated once again with respect to time, yielding

$$\left(\dot{\overrightarrow{\mathbf{V}}}_K^{FR}\right)_T^{TT} = \left(\dot{\overrightarrow{\mathbf{V}}}_K^{FR}\right)_T^{ET} + \left(\dot{\overrightarrow{\boldsymbol{\omega}}}^{TO}\right)_T^T \times \left(\overrightarrow{\mathbf{r}}^{FR}\right)_T + \left(\overrightarrow{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\overrightarrow{\mathbf{V}}_K^{FR}\right)_T^T , \tag{31}$$

which can be further derived using

$$\left(\dot{\overrightarrow{\mathbf{V}}}_K^{FR}\right)_T^{ET} = \left(\dot{\overrightarrow{\mathbf{V}}}_K^{R}\right)_T^{EK} + \left(\overrightarrow{\boldsymbol{\omega}}^{TK}\right)_T \times \left(\overrightarrow{\mathbf{V}}_K^{R}\right)_T^E - \left(\dot{\overrightarrow{\mathbf{V}}}_{K,des}^{F}\right)_T^{ET} . \tag{32}$$

Employing

$$\left(\overrightarrow{\boldsymbol{\omega}}^{TK}\right)_T = \left(\overrightarrow{\boldsymbol{\omega}}^{TO}\right)_T + \left(\overrightarrow{\boldsymbol{\omega}}^{OK}\right)_T , \tag{33}$$

we arrive at

$$\begin{aligned} \left(\dot{\overrightarrow{\mathbf{V}}}_K^{FR}\right)_T^{TT} = {} & \left(\dot{\overrightarrow{\mathbf{V}}}_K^{R}\right)_T^{EK} + \left(\overrightarrow{\boldsymbol{\omega}}^{OK}\right)_T \times \left(\overrightarrow{\mathbf{V}}_K^{R}\right)_T^E - \left(\dot{\overrightarrow{\mathbf{V}}}_{K,des}^{F}\right)_T^{ET} \\ & + \left(\dot{\overrightarrow{\boldsymbol{\omega}}}^{TO}\right)_T^T \times \left(\overrightarrow{\mathbf{r}}^{FR}\right)_T + \left(\overrightarrow{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\left(\overrightarrow{\mathbf{V}}_K^{FR}\right)_T^T + \left(\overrightarrow{\mathbf{V}}_K^{R}\right)_T^E\right) , \end{aligned} \tag{34}$$

where the terms on the top row of (34) contain the kinematic aircraft dynamics (22) with the control inputs $\Delta f_{x,K}$, $f_{y,K}$, $\Delta f_{z,K}$, and the other terms are related to the inputs to the trajectory controller specified in Sect. 2.1.

3.3 Nonlinear Dynamic Inversion Controller

The error dynamics (34) are now transformed using nonlinear dynamic inversion [11, 12]. For this purpose, (34) is written as

$$\ddot{\vec{\mathbf{y}}}_{full} = \mathbf{G}_{full}\left(\gamma_K, \gamma_T, \Delta\chi\right) \vec{\mathbf{u}}_{full} + \vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right), \tag{35}$$

where $\Delta\chi \triangleq \chi_K - \chi_T$ and

$$\kappa \triangleq \left(\left(\vec{\mathbf{r}}^{FR}\right)_T, \left(\vec{\mathbf{V}}_K^{FR}\right)_T^T\right) \tag{36}$$

for brevity. The input vector $\vec{\mathbf{u}}_{full} \in \mathbb{R}^3$ is given by

$$\vec{\mathbf{u}}_{full} = \left[\Delta f_{x,K} \quad f_{y,K} \quad \Delta f_{z,K}\right]^{\mathrm{T}}. \tag{37}$$

Additionally, the matrix $\mathbf{G}_{full}\left(\gamma_K, \gamma_T, \Delta\chi\right) \in \mathbb{R}^{3\times3}$ essentially consists of the transformation from the kinematic frame into the trajectory frame

$$\mathbf{T}_{TK} = \mathbf{T}_{TK}\left(-\gamma_K, -\Delta\chi, \gamma_T\right) \in \mathbb{R}^{3\times3}, \tag{38}$$

and the terms of (34) not containing the aircraft dynamics are incorporated in the nonlinearity vector $\vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right) \in \mathbb{R}^3$. Furthermore, $\ddot{\vec{\mathbf{y}}}_{full} \in \mathbb{R}^3$ represents the second time derivative of the output according to the methodology of nonlinear dynamic inversion, where the output is chosen

$$\vec{\mathbf{y}}_{full} = \left(\vec{\mathbf{r}}^{FR}\right). \tag{39}$$

Now, by choosing the control law

$$\vec{\mathbf{u}}_{full} = \mathbf{G}_{full}^{-1}\left(\gamma_K, \gamma_T, \Delta\chi\right)\left(\vec{\boldsymbol{\nu}}_{full} - \vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right)\right), \tag{40}$$

the plant (35) would be transformed into

$$\ddot{\vec{\mathbf{y}}}_{full} = \vec{\boldsymbol{\nu}}_{full}, \tag{41}$$

where $\vec{\boldsymbol{\nu}}_{full} \in \mathbb{R}^3$ denotes the pseudo control, which is chosen by the designer. Note that (41) represents an integrator chain, which is stable for an adequate choice of this pseudo control.

3.3.1 Reduced Dynamic Inversion

In an aircraft application, the focus lies on aerodynamic integrity of the aircraft. Hence, for application to general aviation aircraft with lower never exceedance speeds, the ground speed is not the control variable. Therefore, the control law (40) is not applied in this paper since it is designed for geometric deviations incorporating the kinematic x-axis.

In case of the DA42, the calibrated airspeed is used as a control variable employing an auto-thrust to compute thrust lever commands to the ECU. In case of waypoint navigation and altitude hold modes, the speed is controlled by thrust whereas speed by pitch is active for flight level change modes. Note that the trajectory controller always implies speed by thrust as it is used to drive geometrical deviations in both y- and z axes to zero. A planned climb from one waypoint to another is consequently performed using speed by thrust. In case of unachievable vertical changes, certain protections occur, which are addressed in Ref. [17].

As the x-axis is surrendered in the DA42 set-up, the specific force in direction of the kinematic x-axis, $\Delta f_{x,K}$, is not used as a control variable anymore. Consequently, the control system is reduced to second order and $\Delta f_{x,K}$ is taken as a measurement. In order to compute the pseudo control $\nu_{full,x}$, the corresponding row of (40) is employed using $\Delta x = 0$, $\Delta\dot{x} = 0$ and by insertion of $\nu_{full,x}$, we arrive at the reduced system given by

$$\mathbf{u} = \mathbf{G}^{-1}\left(\gamma_K, \gamma_T, \Delta\chi\right)\left(\boldsymbol{\nu} - \mathbf{f}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa, \Delta f_{x,K}\right)\right), \tag{42}$$

where $\mathbf{u} = \begin{bmatrix} f_{y,K} & \Delta f_{z,K} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ is the control input, $\mathbf{f}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa, \Delta f_{x,K}\right) = \begin{bmatrix} f_y & f_z \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ is the nonlinearity vector with the components

$$\begin{aligned} f_y = {} & \Delta f_{x,K}\sin(\Delta\chi)\cos(\gamma_T) - V_K^R\dot{\chi}_T\cos(\gamma_K)\cos(\Delta\chi) \\ & - \left(\dot{\chi}_T\dot{\gamma}_T\cos(\gamma_T) + \ddot{\chi}_T\sin(\gamma_T)\right)\Delta z - \dot{\chi}\sin(\gamma_T)\Delta\dot{z}, \end{aligned} \tag{43}$$

$$\begin{aligned} f_z = {} & \Delta f_{x,K}\left(\cos(\gamma_K)\sin(\gamma_T)\left(\cos(\Delta\chi) - 1\right) - \sin(\gamma_K - \gamma_T)\right) \\ & + V_K^R\dot{\gamma}_T\left(\cos(\gamma_K)\cos(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T)\right) \\ & + V_K^R\dot{\chi}_T\cos(\gamma_K)\sin(\gamma_T)\sin(\Delta\chi) + \dot{\chi}\sin(\gamma_T)\Delta\dot{y} \\ & + \left(\dot{\chi}_T\dot{\gamma}_T\cos(\gamma_T) + \ddot{\chi}_T\sin(\gamma_T)\right)\Delta y, \end{aligned} \tag{44}$$

and the input matrix $\mathbf{G} \in \mathbb{R}^{2\times2}$ is given by

$$\mathbf{G} \triangleq \begin{bmatrix} \cos(\Delta\chi) & \sin(\Delta\chi)\sin(\gamma_K) \\ -\sin(\Delta\chi)\sin(\gamma_T) & \sin(\gamma_K)\sin(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T) \end{bmatrix}. \tag{45}$$

Note that $\mathbf{G}$ is not invertible if the determinant given by

$$\det\left(\mathbf{G}\right) = \cos(\gamma_K)\cos(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T) \tag{46}$$

is zero. This is the case for $|\Delta\chi| \approx 90°$ as the remaining term $\sin(\gamma_K)\sin(\gamma_T)$ is zero for horizontal trajectories and / or horizontal flight. Since fly-by trajectories are planned smoothly, no large deviations of the course angles occur and hence, the inverse can be computed in these cases. Actually, the difference in course angles $\Delta\chi$ stays close to zero even in the presence of wind as the trajectory controller is designed to follow the geometric path rather than using a cascaded structure with course angle commands as commonly used in the literature [1–10]. However, since the determinant takes effect as an additional gain, the difference in course angles is limited to $|\Delta\chi| = 60°$, which corresponds to a gain of 2 if $\gamma_T = 0$ and $\gamma_K = 0$.

For the ideal case of $\Delta\chi = 0$, the inverse of the input matrix becomes diagonal

$$\mathbf{G}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\cos(\gamma_K - \gamma_T)}, \end{bmatrix} \tag{47}$$

which facilitates the possibility to evaluate the effects onto different axes, while the small terms due to the difference in course angles $\Delta\chi$ also disappear from the nonlinearity terms (43), (44), yielding

$$f_y = -V_K^R \dot{\chi}_T \cos(\gamma_K) - (\dot{\chi}_T \dot{\gamma}_T \cos(\gamma_T) + \ddot{\chi}_T \sin(\gamma_T))\, \Delta z - \dot{\chi} \sin(\gamma_T) \Delta\dot{z}, \tag{48}$$

$$f_z = V_K^R \dot{\gamma}_T \cos(\gamma_K - \gamma_T) + (\dot{\chi}_T \dot{\gamma}_T \cos(\gamma_T) + \ddot{\chi}_T \sin(\gamma_T))\, \Delta y + \sin(\gamma_T) \Delta\dot{y} \\ - \Delta f_{x,K} \sin(\gamma_K - \gamma_T), \tag{49}$$

where the first terms correspond to feedforward elements due to the curvature $\dot{\chi}_T$ and $\dot{\gamma}_T$, respectively, the specific force along the kinematic x-axis $\Delta f_{x,K}$ is incorporated by means of another feedforward element to deal with accelerations, and the remaining terms serve as elements to rotate the deviations into the changing trajectory frame. While the ideal case $\Delta\chi = 0$ renders Eqs. (47)–(49), the small terms for $\Delta\chi \neq 0$ correlate to additional coupling terms.

3.3.2 Pseudo Control Law

The pseudo controls are chosen as PD controller as per

$$\boldsymbol{\nu} \triangleq \begin{pmatrix} \nu_y \\ \nu_z \end{pmatrix} = \begin{pmatrix} k_{p,y} \Delta y + k_{d,y} \Delta\dot{y} \\ k_{p,z} \Delta z + k_{d,z} \Delta\dot{z} \end{pmatrix}, \tag{50}$$

where the gains $k_i \in \mathbb{R}^-$ can be computed according to

$$\begin{aligned} k_{p,y} &= -\omega_y^2, \\ k_{d,y} &= -2\omega_y, \\ k_{p,z} &= -\omega_z^2, \\ k_{d,z} &= -2\omega_z, \end{aligned} \tag{51}$$

where ω_y and ω_y are the desired natural frequencies for an aperiodic second order system since no overshoots are desired for trajectory tracking.

4 Illustrative Examples

In this section, the functionality and the performance is further analyzed using illustrative examples. For this purpose, two simulation studies are presented. In the first simulation example, the nonlinear dynamic inversion controller is compared to a linear controller to evaluate the performance with respect to classical linear approaches. Then, the trajectory controller's performance when subject to wind is discussed using another example. Both tests are conducted with the same NDI controller as reference at a commanded airspeed of 90kts.

For the comparison with the linear controller, all input and output treatments applied to the controller remain as designed for the NDI approach. That is, the implicit gain given by the inverse of the input matrix, $\mathbf{G}^{-1}$, the command limits, and the command transformation to the inner loop are retained. Furthermore, the exact same PD control law (51) is used, where $\omega_y = 0.23\frac{rad}{s}$ and $\omega_z = 0.45\frac{rad}{s}$ are utilized in both cases. Consequently, the nonlinearity term given in Eq. (43) is the only remaining difference between the control laws.

For the comparison, a flightplan in triangular form, which consists of various fly-by maneuvers and a loiter at the end, is simulated. The flightplan and the actual flight paths are displayed in Fig. 3. Note that the planned trajectory is identical as the kinematic velocity used for radius computation does not vary throughout the simulation as no wind is applied to address the conceptual differences.

It is evident that the linear controller cannot follow the trajectory in turns and overshoots significantly both when entering and leaving the turn. This fact is further supported by Fig. 4, where large deviations are obvious. In comparison, the NDI controller as illustrated in Fig. 5 remains within deviations of 10 m when building up the roll angle required for turning into or leaving the clothoid, respectively. This deviation can be further reduced by a faster inner loop controller, which, however, is not in the focus right now as conservatism is of higher interest for the current phase of flight testing.

Furthermore, the linear controller exhibits significant steady-state deviations in turns. Although also the NDI controller cannot achieve steady-state accuracy during turns due to the lack of an integrator (in particular during the loiter at the end of the flightplan in Fig. 5), the nonlinearity term approximately succeeds in driving the deviation from the trajectory to zero by means of the feedforward term. Just using a linear error controller as made use of in various applications inherently results in large deviations with respect to curved trajectories as an error needs to be built up for the controller to react. The specific forces commanded by the trajectory controller are in the same magnitude for both controllers, where the NDI controller achieves the slightly faster reaction and the basic turn specific force command by the feedforward term while the linear controller generates its command amplitude by the control error.

For the second example, a spiral pattern is considered, which also consists of various fly-by maneuvers and ends with a loiter. This pattern is chosen to solely contain turns with changing kinematic velocities as the trajectory generation module employs a combination of kinematic velocity, wind estimate, airspeed command, and

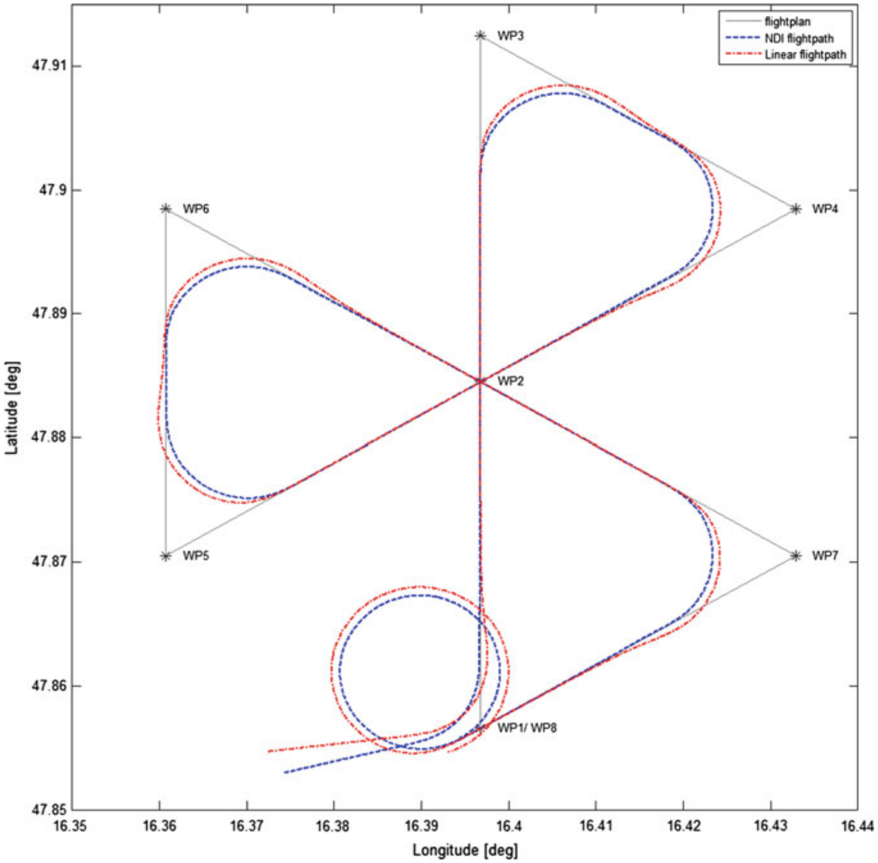

Fig. 3 Flight plan: triangular pattern

the actual airspeed to compute the trajectory's curve as noted in Sect. 2.1. Hence, the trajectory controller can be analyzed with respect to changing velocities, especially with respect to the planned curve. The flightplan displayed in Fig. 6 exhibits the different curves at south wind of 0, 20 and 40kts. As the wind streams towards north, a larger kinematic velocity occurs flying "up" whereas flying south employs the minimum kinematic velocity. As the trajectory generation module utilizes the maximum expected velocity throughout a turn, the planned radii become larger with increasing wind. In particular, this is evident for the loiter maneuver at the end of the flightplan.

Figures 7 and 8 show that there are almost no differences in deviations and course angles between no wind and 20kts of wind. The commands, however, change depending on the kinematic velocity, i.e. the commands required to follow the curve vary significantly to achieve the same objectives. Whereas the turn commands in Fig. 7 retain a certain amplitude, the command's magnitudes change depending on the

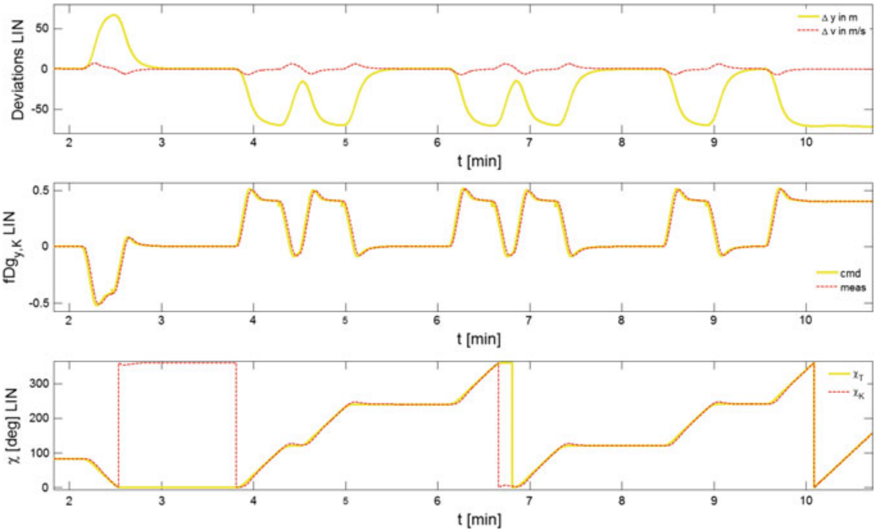

Fig. 4 Linear controller

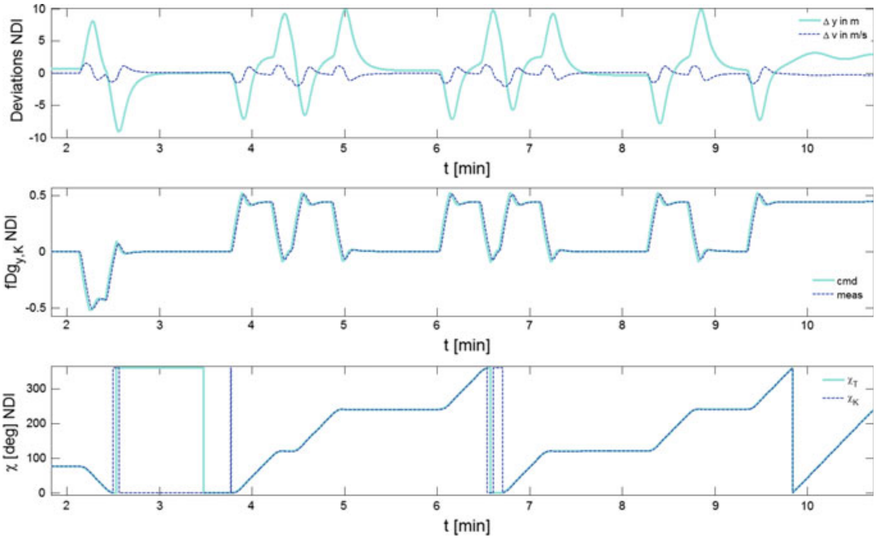

Fig. 5 NDI controller

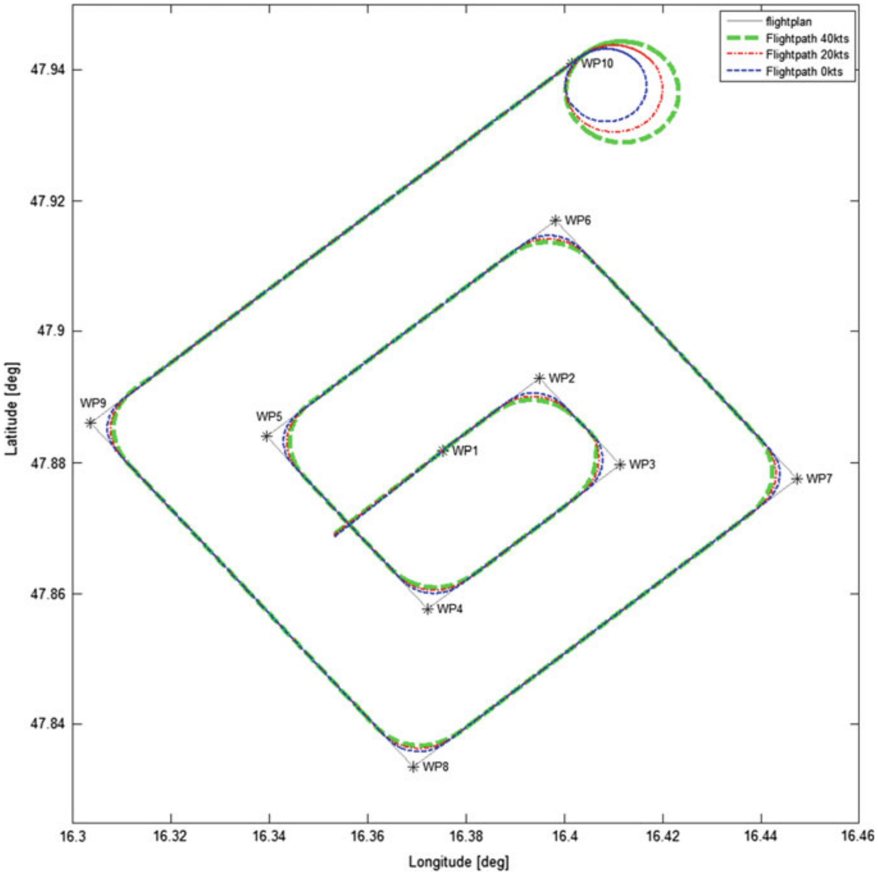

Fig. 6 Flight plan: spiral pattern, wind comparison

direction in the case of 20kts wind as displayed in Fig. 8. When performing the loiter at the end of the flightplan, the change is most evident.

As a conclusion, the trajectory controller achieves superior performance with respect of a similar linear PD controller and is robust against the influences of wind.

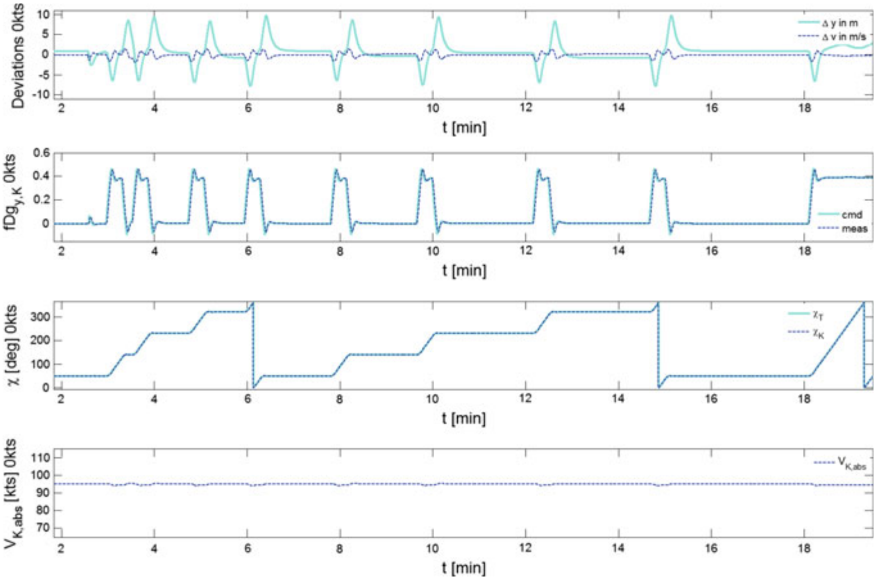

Fig. 7 NDI controller, no wind

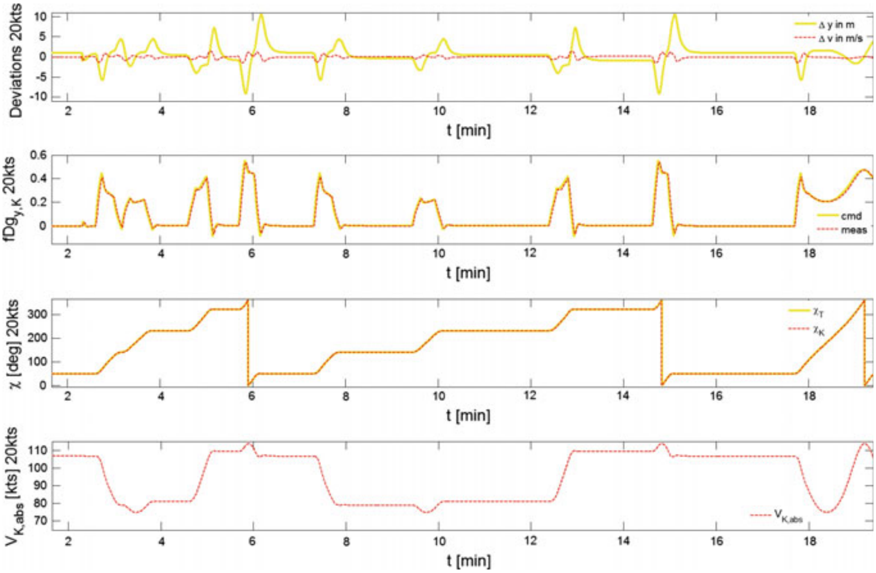

Fig. 8 NDI controller, 20kts N-wind

5 Conclusions

In this paper, a modular trajectory controller was proposed and the concept for fly-by trajectories was described in detail. Specifically, the environment including the interfaces was presented and a brief introduction to the inner loop was given. In simulations, the performance of the nonlinear controller was compared to a linear controller of the same order and the influence of significant wind was analyzed.

In addition to previously-published principal flight tests, further flight tests with the DA42 experimental aircraft of the institute are being conducted focusing on automatic landing by generation of smooth landing trajectories. The modular design of the overall control architecture enables fast applications to other aircraft. Specifically, flight tests with a motor glider and a fixed-wing aircraft of 6 tons are about to be conducted while a small UAV of 150 kg is being tested in hardware-in-the-loop environments.

Acknowledgements I would like to offer my special thanks to my colleague Volker Schneider, who is responsible for the trajectory generation module, which was also used to generate the figures of this paper. Furthermore, I would also like to extend my thanks to Agnes Gabrys, Erik Karlsson, and Alexander Zollitsch, who provided the inner loop, the autopilot module including energy protections, and the high-fidelity simulation model, respectively.

References

1. Beard RW, Kingston D, Quigley M, Snyder D, Christiansen R, Johnson W, McLain T, Goodrich M (2005) Autonomous vehicle technologies for small fixed-wing uavs. J Aerosp Comput Inf Commun 2(1):92–108
2. Low CB (2010) A trajectory tracking control design for fixed-wing unmanned aerial vehicles. In: IEEE multi-conference on systems and control, pp 2118–2123
3. Ren W, Beard RW (2004) Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. IEEE Trans Control Syst Technol 12(5):706–716
4. Subbarao K, Ahmed M (2014) Nonlinear guidance and control laws for three-dimensional target tracking applied to unmanned aerial vehicles. J Aerosp Eng 27(3):604–610
5. Nelson DR, Barber DB, McLain TW, Beard RW (2007) Vector field path following for miniature air vehicles. IEEE Trans Robot 23(3):519–529
6. Zhu S, Wang D, Low CB (2013) Ground target tracking using uav with input constraints. J Intell Robot Syst 69(1–4):417–429
7. Beard RW, Ferrin J, Humpherys J (2014) Fixed wing uav path following in wind with input constraints. IEEE Trans Control Syst Technol :1
8. Zhufeng Xie, Yuanqing Xia, Mengyin Fu (2011) Robust trajectory-tracking method for uav using nonlinear dynamic inversion. In: IEEE 5th international conference on cybernetics and intelligent systems (CIS) pp 93–98
9. Pashilkar AA, Ismail S, Ayyagari R, Sundararajan N (2013) Design of a nonlinear dynamic inversion controller for trajectory following and maneuvering for fixed wing aircraft. In: IEEE symposium on computational intelligence for security and defense applications (CISDA) pp 64–71
10. Oliveira T, Encarnação P (2013) Ground target tracking control system for unmanned aerial vehicles. J Intell Robot Syst 69(1–4):373–387
11. Slotine J-JE, Li W (1991) Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, NJ

12. Khalil HK (2002) Nonlinear Systems, 3rd edn. Prentice Hall, Englewood Cliffs, NJ
13. Schatz SP, Holzapfel F (2014) Modular trajectory, path following controller using nonlinear error dynamics. In: IEEE international aerospace electronics and remote sensing technology (ICARES) pp 157–163
14. Schatz SP, Schneider V, Karlsson E, Holzapfel F et al (2016) Flightplan flight tests of an experimental da42 general aviation aircraft. In: The 14th International Conference on Control, Automation, Robotics, and Vision
15. Krause C, Holzapfel F (2016) Designing a system automation for a novel uav demonstrator. In: The 14th International Conference on Control, Automation, Robotics, and Vision
16. Karlsson E, Schatz SP et al (2016) Automatic flight path control of an experimental da42 general aviation aircraft. In: The 14th International Conference on Control, Automation, Robotics, and Vision
17. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: The 14th International Conference on Control, Automation, Robotics, and Vision
18. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2014 IEEE International Aerospace Electronics and Remote Sensing Technology (ICARES). IEEE
19. Schneider V, Piprek P, Schatz SP et al (2016) Online trajectory generation using clothoid segments. In: The 14th international conference on control, automation, robotics, and vision
20. Meek DS, Walton DJ (2004) A note on finding clothoids. J Comput Appl Math 170(2):433–453
21. Henrie J, Wilde D (2007) Planning continuous curvature paths using constructive polylines. J Aerosp Comput Inf Commun 4(12):1143–1157
22. Wilde DK (2009) Computing clothoid segments for trajectory generation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS) pp 2440–2445
23. Lavretsky E, Wise KA (2013) Robust and Adaptive Control: With Aerospace Applications, Ser. Advanced textbooks in control and signal processing. Springer, New York
24. Andry AN, Shapiro EY, Chung J (1983) Eigenstructure assignment for linear systems. IEEE Trans Aerosp Electron Syst AES–19(5):711–729
25. Etkin B, Reid LD (1996) Dynamics of Flight: Stability and Control, 3rd edn. Wiley, New York
26. Stengel RF (2004) Flight Dynamics. Princeton University Press, Princeton and NJ