Optimization-Based Feedforward Path Following for Model Reference Adaptive Control of an Unmanned Helicopter

Johann C. Dauer[*]

German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, Germany

Timm Faulwasser[†]

École Polytechnique Fédérale de Lausanne, Laboratoire d'Automatique, Switzerland

Sven Lorenz[‡]

German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, Germany

Rolf Findeisen[§]

Otto-von-Guericke-University Magdeburg, Institute for Automation Engineering, Magdeburg, Germany

This paper presents an optimization-based approach to let an unmanned helicopter follow a geometrically defined path. In particular, this approach extends reference model following concepts. Instead of using a model of the vehicle dynamics, the optimization is based on the reference model of the controller. By this means, we can calculate the time-wise progress on the path by means of dynamic optimization without exact knowledge of the real helicopter dynamics. The progression on the path, is defined as a dynamic system subject to an additional virtual control input. The inputs of the reference model and that of the timing law are the decision variables used in the dynamic optimization which is so far performed offline. It will be shown that hereby, an accurate following of the path is possible although the actually identified flight mechanical model is limited to a linear hover model. Furthermore, the approach allows to take constraints on inputs and states into account. Simulation results as well as flight tests conducted with the ARTIS testbed underline that good performance and constraints satisfaction can be achieved.

Notation

Reference Model

$\mathbf{A}_{v,r}, \mathbf{A}_{a,r}$	parameters of reference dynamics	(u, v, w)	geodetic kinematic velocities
$\mathbf{e}_i$	unit vector of component i	$\underline{q}$	attitude unit quaternion
$\mathbf{g}$	geodetic gravit. acceleration vector	$\underline{q}_\Delta$	difference non-unit quaternion
$\mathbf{m}$	geodetic force vector acting on helicopter normalized by mass	$\mathbf{q}_\Delta$	vector component of $\underline{q}_\Delta$
		$\mathbf{v}$	body-fixed flight-path velocity vector
$\mathbf{p}$	position vector	$\boldsymbol{\delta}$	vector of actuator commands
$\mathbf{t}$	geodetic thrust vector normalized by mass of the helicopter	ζ_e, ω_e	parameters of the engine model
		$\boldsymbol{\omega}$	vector of body-fixed rotational rates
T	magnitude of $\mathbf{t}$	$(\cdot, ..., \cdot)$	vector collocation with elements $\cdot$
t	time		

[*] Research Scientist, Unmanned Aircraft Department, johann.dauer@dlr.de, AIAA Member

[†] Postdoctoral Researcher, Laboratoire d'Automatique, timm.faulwasser@epfl.ch

[‡] Research Scientist, Unmanned Aircraft Department, sven.lorenz@dlr.de, AIAA Senior Member

[§] Full Professor, Institute for Automation Engineering, rolf.findeisen@ovgu.de

Model Used for Optimization

$\mathbf{e}$	error states
E	integral path error
$\mathcal{D},\mathcal{U},\mathcal{X}$	constraint sets of actuator deflections, inputs and states
$\mathbf{f}$	vector field of the optim. model
$\mathbf{h}$	output map of the optim. model
J	cost functional
$\mathcal{P}$	set of points on reference path
$\mathbf{l}$	timing-law
$\mathbf{Q},\mathbf{R}$	weighting matrices
$\mathbf{u}\in\mathbb{R}^3$	input of the optim. model
$\mathbf{u}(\cdot)$	input trajectory $u : t \in [0,\infty) \rightarrow u(t)$
$\mathbf{x}\in\mathbb{R}^{15}$	state vector
$\mathbf{x}(\cdot)$	state trajectory starting at x_0 driven by the input $u(\cdot)$
$\mathbf{y}\in\mathbb{R}^3$	output of the optim. model
$\mathbf{z}=(\theta,\dot{\theta})$	state vector of the timing law
$\theta\in[\theta_0,\theta_1]$	scalar path parameter
τ	time interval covering the path
τ_p	optimization horizon
Δ	time segmentation of τ

Superscripts

$.'$	signals without actuator delay
$\vec{\cdot}$	vectors of unit length
$\bar{\cdot}$	conjugate of a quaternion
$.^\star$	optimal solution to dynamic optimization problem

Subscripts

$\cdot_0$	initial condition
$\cdot_c$	commanded trajectory
$\cdot_{e,\dot{e},T,\dot{T}\theta,\dot{\theta},u,v}$	elements of weighting matrices concerning $e,\dot{e},T,\dot{T},\theta,\dot{\theta},u,v$
$\cdot_r$	reference trajectory
$\cdot_{rot}$	components influencing the rotation
$\cdot_s$	state (estimated) trajectory
$\cdot_k$	k-th receding horizon iteration
$\cdot_{\mathcal{P}}$	path-specific variables

Abbreviations

ARTIS	autonomous rotorcraft testbed for intelligent systems
MTOW	maximum take-off weight
NN	neural network
ODE	ordinary differential equation
OCP	optimal control problem
PCH	pseudo-control hedging

I. Introduction

The control of unmanned helicopters during fast flight maneuvers is challenging. If a reference path is given geometrically the time-wise progress on the path remains free. Difficulties arise from limits to spatial movement, i.e. hard constraints that specify restrictions in translation and attitude of the helicopter along the path. Furthermore, control during fast flight maneuvers usually requires detailed knowledge of the underlying dynamics.

The first aspect is typically handled in the path and trajectory planning task, and numerous approaches to these problems have been presented in the past, cf. Dadkah[1] and Huang[2] for concise surveys. Roughly, two categories of approaches can be distinguished: The first set of techniques utilizes motion primitives that comprise a time-wise evolution of a trajectory which in turn results in a limited set of possible path geometries. The second set of planning approaches uses motion primitives that consider generic dynamic constraints for path geometries but without an exact time-wise reference. In the work presented here, the optimization of the path's geometry is decoupled from a time-wise evolution along the path. Thus, an example of a hierarchical overall optimization architecture is formed comprising global path planning and local trajectory optimization. The scope of this paper is put on the latter, i.e. on trajectory optimization for a given path geometry.

To partially avoid the difficulties of modeling and identifying complex helicopter dynamics *model reference adaptive control* (MRAC) has been proposed, e. g. by Ioannou[3] or Narendra,[4] and successfully applied to the control of helicopters.[5,6] Basically, MRAC enforces that the helicopter behaves like a designed reference model by means of adaptive control. Furthermore, the reference model is designed so that it provides flyable transitions and corresponding state trajectories between commands for the underlying trajectory controller to track.

However, if maneuvers have to be performed with a high spatial accuracy the reference model of an MRAC strategy can lead to large path deviations. The reason is that the reference model does not enforce the state transients to those geometrically lying on the path. In the context of tasks which require to accurately follow a geometric reference path in geodetic coordinates, different approaches have been proposed to overcome this bottleneck. For example, Lorenz and Dauer[7] consider static velocity limits which are based on maximum bank angle as well as acceleration capacities at certain positions based on the path's geometry. Additionally, they propose a time shift to minimize path deviations caused by the reference transients. However, an adequate time shift has to be found for each specific flight path. Moreover, this approach is limited to static velocity limits and the reserves in acceleration capabilities due to the curvature of the path. It is not straightforward to include further constraints of the vehicle dynamics.

Here, we present a dynamic optimization approach utilizing the nominal reference model to compute the inputs of the underlying MRAC controller for precise path following. We rely on concepts developed in the context of optimization-based path-following to compute the time evolution of the unmanned helicopter states along the required flight path as well as the inputs to the MRAC controller. In essence we pre-compute by means of a series of dynamic optimization problems the reference signals of the MRAC controller in form of feedforward inputs.* More details on the design of optimization-based and predictive path-following controllers can be found in the work of Faulwasser et al.[8–10]

Instead of solving the dynamic optimization problem for a complete path at once, we propose a receding horizon strategy that solves the problem recursively over short time horizons. This way we reduce the computational effort of the optimization which is performed offline. Furthermore, we consider constraints on the states of the MRAC reference model as well as on the inputs of the real system. Thus flight envelope and mechanical safety constraints are explicitly taken into account in the problem formulation. This way all known dynamic capabilities of the helicopter shall be regarded. Additionally, the control architecture of the underlaying MRAC shall not be altered to retain all its positive properties which include feasible trajectory generation for non-continuous commands. Rather, the proposed concept should be considered as an optional module of the flight software.

The remainder of the paper is structured as follows: A brief overview of the controller architecture is given in Section II. The main focus is on explaining the design of a cascaded reference model. The design of the reference model sets the basis for Section III, which outlines the optimization-based computation of the feedforward controls along the flight path. The test scenario and simulations are presented in Section IV including the considered flight paths as well as expected flight performance. Afterwards, flight test results using the testbed ARTIS (see Figure 1) of the German Aerospace Center, Braunschweig, are discussed.

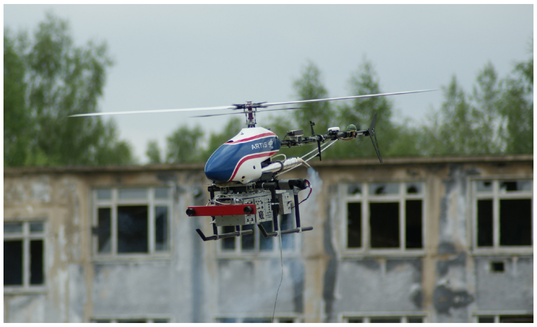

Figure 1. Unmanned helicopter ARTIS (Autonomous Helicopter Testbed for Intelligent Systems), 2 m rotor diameter, 1.5 kW 2-stroke engine, 14 kg MTOW.

II. Control Architecture

To have a proper context for the optimization-based path following, this section introduces the control architecture used for the ARTIS family. A separation of the control system into inner and outer loops is a well-established design method in aerospace applications.[11] One of the main reasons for the decoupling of position control from velocity and attitude control within the ARTIS project is the different levels of autonomy required for the operation of the vehicle. It is desired to have different command sources on different levels of abstraction, for instance, a safety pilot via remote control, ground control station via a joystick, and path-following control as treated in this paper.

*This means that we specify the inputs to the MRAC-controller—i.e., the velocity commands—as feedforward or open-loop inputs, which are pre-computed offline.

Figure 2 gives a compact overview of the architecture of the control system. It shows the cascade of the velocity and attitude control. This concept is similar to the model reference adaptive controller presented by Johnson.[6] Hedging (or hiding) of unconsidered actuator dynamics within the inversion from the adaptive elements is achieved in a feedforward fashion as presented by Lorenz.[12] This way, there is no state feedback to the reference model in order to hide saturations from the adaptive elements. These elements consist of single hidden layer neural networks (NN) supporting the velocity loop. The subsequent paragraphs briefly describe the control architecture of the two inner loops before focusing on the reference models.

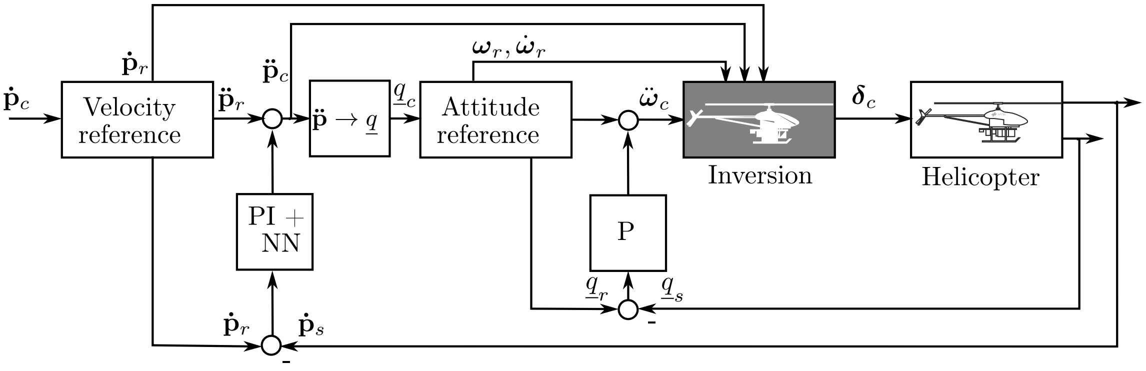

Figure 2. Control architecture of the ARTIS family.

The MRAC controller uses a reference model in each loop to generate feasible reference trajectories (denoted by the index r) for the helicopter which are then used for tracking as shown in Figure 2. The velocity reference model generates the reference acceleration $\ddot{\mathbf{p}}_r$ and velocity $\dot{\mathbf{p}}_r$. This reference velocity is used for error feedback, using a linear PI controller and a single hidden layer neural network. The feedback-modified signal is the command for the inner loop and an attitude command $\underline{q}_c$ is determined which ensures that the thrust of the main rotor is oriented correctly for the required acceleration.

The attitude reference model is used to generate the trajectories for the rotational states $(\underline{q}_r, \boldsymbol{\omega}_r, \dot{\boldsymbol{\omega}}_r)$. The attitude error is fed back using a proportional controller. The body-fixed rotational rates, their derivatives and the velocity are inputs for the inversion represented by the inverse-colored helicopter in Figure 2. The inversion of the helicopter model is based on a linear hover model, as it is often used for this kind of helicopters, see e.g. Mettler.[13] The inversion generates swashplate and tail rotor control inputs $\boldsymbol{\delta}_c$ corresponding to the reference trajectories.

The components of the reference models are outlined in more detail in the next sections, as they form the basis for the optimization. The implemented reference models are more complex than discussed in this paper. For instance, they contain aspects of envelope protection and saturations of actuators. For brevity only the dynamics that are later used for optimization will be discussed. Also note, that the difference between the implemented reference model within the controller and the one used for optimization is intentional. The more complex the equations for the optimization become, the higher is the computational burden of the optimization. Thus, if aspects of the reference modeling are neglected for the purpose of optimization, the computational demand is reduced at the expense of accuracy of the path-following.

A. Velocity Reference Model

This section outlines the structure of the velocity reference model. It generates the reference trajectory for the outer loop and also calculates the command for the attitude reference model. The translation reference model is designed to be of first order in the reference velocity $\dot{\mathbf{p}}_r$

$$\ddot{\mathbf{p}}'_r(t) = \mathbf{A}_{v,r}\big(\dot{\mathbf{p}}_c(t) - \dot{\mathbf{p}}_r(t)\big). \tag{1a}$$

The geodetic velocity command is denoted $\dot{\mathbf{p}}_c(t)$. The matrix $\mathbf{A}_{v,r} \in \mathbb{R}^{3\times 3}$ contains the design parameters for the reference dynamics. The diagonal entries contain the inverses of the time constants in the directions of the geodetic frame. The remaining entries are zero. The meaning of the primed values will become apparent shortly. The resulting trajectories of acceleration and velocity can already be used within the model inversion to calculate the collective swashplate deflection. However, a dynamic of second order additionally copes with the dynamics of the engine similar to the pseudo-control hedging (PCH) approach of Johnson.[6]

Consider the equilibrium of forces as depicted in Figure 3. It follows that the engine has to generate the thrust

$$\mathbf{t}'_r(t) = \ddot{\mathbf{p}}'_r(t) - \mathbf{m}_r(\mathbf{x}_r(t), \boldsymbol{\delta}_{c,rot}(t)) - \mathbf{g}. \tag{1b}$$

In this equation, all forces are normalized by the mass of the helicopter, including the thrust vector $\mathbf{t}_r$ and the forces $\mathbf{m}_r$, like drag or lateral force due to the tail rotor. The forces $\mathbf{t}_r$ are calculated based on the reference states $\mathbf{x}_r = (\mathbf{v}_r^T, \boldsymbol{\omega}_r^T, \dot{\boldsymbol{\omega}}_r^T)^T$ and the states of the three rotation-influencing actuators ($\boldsymbol{\delta}_{c,rot}$). The linear hover model of the dynamic inversion is used to estimate these forces.

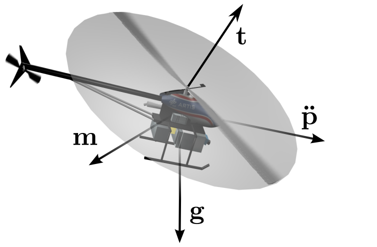

Figure 3. Equilibrium of mass-normalized forces.

The obtained thrust signal is delayed by a model of the engine dynamics and converted back into a corresponding delayed acceleration with a rearranged version of (1b). This process results in the actual reference acceleration $\ddot{\mathbf{p}}_r$. In the equations above the prime denotes the undelayed values. The second-order dynamics of the engine model act on the magnitude of the thrust vector $T' = \|\mathbf{t}'\|$

$$\ddot{T}(t) + 2\zeta_e\omega_e\dot{T}(t) + \omega_e^2 T(t) = \omega_e^2 T'(t). \tag{1c}$$

The parameters of the thrust model are the damping coefficient ζ_e and the eigenfrequency ω_e.

Using the thrust vector of (1b) the required attitude can be calculated to ensure the translational motion. It is assumed that the thrust is directed in negative body-fixed z-direction ($-\vec{\mathbf{e}}_3$). With the notation of $\vec{\cdot}$ to denote normalized vectors the rotation for the thrust orientation can be represented by a unit quaternion

$$\underline{q}_t = \frac{1}{\sqrt{(\vec{\mathbf{t}}'_r(t) - \vec{\mathbf{e}}_3)\cdot(\vec{\mathbf{t}}'_r(t) - \vec{\mathbf{e}}_3)}} \begin{pmatrix} 1 - \vec{\mathbf{t}}'_r(t)\cdot\vec{\mathbf{e}}_3 \\ \vec{\mathbf{t}}'_r(t)\times\vec{\mathbf{e}}_3 \end{pmatrix}. \tag{1d}$$

The rotation of the thrust into the required direction, however, does not define the attitude completely. One degree of freedom—the yaw angle—remains, and thus has to be considered in order to calculate the complete command $\underline{q}_c$ for the attitude reference model, see Appendix B for details.

B. Attitude Reference Model

The attitude reference model is of third order. The rotational dynamics of the helicopter's rigid body motivate a second-order system. The additional order is introduced to handle the flapping of the rotor blades, which is approximated by a first-order ordinary differential equation (ODE). The reference model is defined as

$$\ddot{\boldsymbol{\omega}}'_r(t) = \mathbf{A}_{a,r}\begin{pmatrix}\mathbf{q}_\triangle(t) \\ \boldsymbol{\omega}_r(t) \\ \dot{\boldsymbol{\omega}}_r(t)\end{pmatrix}. \tag{1e}$$

The matrix $\mathbf{A}_{a,r} \in \mathbb{R}^{3\times 9}$ contains the design parameters and $\boldsymbol{\omega}$ the body-fixed rotational rates. The coefficients in $\mathbf{A}_{a,r}$ define the eigenfrequencies and damping of the reference model. This matrix is defined as a composition of three square diagonal matrices without coupling of the different axes. The angles $\mathbf{q}_\triangle$ represent the difference between the commanded attitude $\underline{q}_c$ and the current reference attitude $\underline{q}_r$. These angles can be determined by using the last three components of

$$\underline{q}_\triangle(t) = 2\bar{\underline{q}}_r(t)(\underline{q}_c(t) - \underline{q}_r(t)). \tag{1f}$$

Products involving quaternions use the particular *quaternion product.* The attitude follows from the body-fixed rates using

$$\dot{\underline{q}}_r(t) = \frac{1}{2}\underline{q}_r(t)\begin{pmatrix}0 \\ \boldsymbol{\omega}_r(t)\end{pmatrix}. \tag{1g}$$

In contrast to the velocity reference, only the fast servo dynamics have to be considered here. As the sample rate used for the controller is 50 Hz, a delay of one sample time and saturation achieve the hedging for the adaptive components of the controller. These delays remain in place within the controller loops of Figure 2 but are dropped for optimization. Further simplification is the removal of some envelope protection mechanisms from the implementation of the dynamic model used for optimization. We treat these state limitations as constraints of the optimization as described later on. As the inner loop controller as well as its reference model remain unchanged during flight test, the state protection mechanisms remain in place even if they are not treated entirely in the constraint specification.

III. Optimization-Based Computation of Feedforward Controls

In this paper we aim at the computation of feedforward controls, i.e. the velocity reference signal of the MRAC controller, that achieve small deviations from a given geometric flight path, satisfaction of constraints on the real helicopter inputs, as well as a sufficiently fast motion of the unmanned helicopter along the path. The main idea of the proposed approach is to compute feedforward inputs by means of a receding horizon dynamic optimization, which is closely related to predictive path-following concepts and optimization-based approaches to path following. We refer to Faulwasser[9,10] and Mayne[14] for the former and to Faulwasser[8,15] for the latter.

A. Mathematical Problem Formulation

In the following, we represented the dynamics of the MRAC-controlled helicopter by the reference model (1), i.e., we assume the closed-loop dynamics correspond to the reference model. This assumption is motivated by the fact that in direct MRAC the reference model expresses the desired dynamics of the closed-loop system.[3,4] We assume that deviations from the desired dynamics are sufficiently small.

We begin by reformulating the velocity reference model and the attitude reference model from (1) as a system of first-order ODE

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^{15} \tag{2a}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \tag{2b}$$

where the state vector $\mathbf{x} \in \mathbb{R}^{15}$ and the output vector $\mathbf{y} \in \mathbb{R}^3$ are defined as follows

$$\mathbf{x} = (T, \dot{T}, \boldsymbol{\omega}^T, \dot{\boldsymbol{\omega}}^T, \mathbf{p}_r^T, \dot{\mathbf{p}}_r^T, \underline{q}_r{}^T)^T \tag{3}$$

$$\mathbf{y} = \mathbf{p}_r \tag{4}$$

i.e., the output contains the geodetic position of the reference model. The considered input is the reference velocity commanded to the MRAC controller, i.e., $\mathbf{u} = \dot{\mathbf{p}}_c \in \mathbb{R}^3$. In the following we denote the contrained input space $\mathcal{U}$, and the constrained state space $\mathcal{X}$.

As mentioned before, we aim at the computation of feedforward controls which achieve that the unmanned helicopter follows a given geometric flight path as precisely as possible while flying sufficiently fast. In other words, we want to achieve that the output of (2) travels along a given flight path, which we represent as a geometric curve

$$\mathcal{P} = \left\{ \mathbf{y}_{\mathcal{P}} \in \mathbb{R}^3 \mid \theta \in [\theta_0, \theta_1] \mapsto \mathbf{p}(\theta) \right\}. \tag{5}$$

Here, the scalar variable θ is denoted as path parameter, and the path $\mathcal{P}$ is assumed to be a regular curve given by its parameterization $\mathbf{p}(\theta)$, cf. Topogonov.[16]

In order to compute the desired feedforward controls we rely on path-following concepts, for details we refer to Faulwasser.[8–10,15] The conceptual idea of path following is based on a simple observation: If the desired flight path is described as a reference trajectory it implies an explicit requirement *when to be where* on the reference. In a path-following setup, however, the flight path is considered as a geometric reference. This means that the velocity to fly along the path is a degree of freedom during the design/computation of suitable feedforward controls. To exploit this idea to its full extent we treat the time evolution of θ—i.e., the mapping $t \mapsto \theta(t)$—as a degree of freedom during the design of the feedforward controls for the MRAC controller. More formally, we approach the computation via the following problem setting.

Problem 1 (Constrained Feedforward Path Following) *Given the system* (2)*, and a path* $\mathcal{P}$*, compute a transition time* $\tau \in (0, \infty)$*, a feedforward control* $\mathbf{u}_{\mathcal{P}} : [0, \tau] \to \mathcal{U}$ *and a path evolution* $\theta : [0, \tau] \to [\theta_0, \theta_1]$*, which achieve the following:*

i) ***Minimization of Path Deviation:*** *The path error*

$$E\left(\mathbf{u}_{\mathcal{P}}(\cdot), \theta(\cdot)\right) = \int_0^{\tau} \left\|\mathbf{h}(\mathbf{x}(t)) - \mathbf{p}(\theta(t))\right\|^2 dt$$

is minimized.

ii) ***Forward Motion:*** *The system* (2) *moves in finite time* τ *from* $\mathbf{h}(\mathbf{x}_0)$ *in forward direction to the final point* θ_1*, i.e., for all* $t \in [0, \tau] : \dot{\theta}(t) \geq 0$ *specified by a reference behavior introduced later and* $\mathbf{h}(\mathbf{x}(\tau)) = \mathbf{p}(\theta_1)$*.*

iii) ***Constraint Satisfaction:*** *The input and state constraints of* (2) *are satisfied for all times, i.e., for all* $t \in [0, \tau] : \mathbf{u}_{\mathcal{P}}(t) \in \mathcal{U}$ *and* $\mathbf{x}(t, x_0; \mathbf{u}_{\mathcal{P}}(\cdot)) \in \mathcal{X}$*.*

B. Receding Horizon Optimization

Exact solutions to this problem are in general difficult to obtain.[†] To allow a sufficiently fast solution, we limit our attention to a reasonable approximation, i.e., a solution leading to a sufficiently small but not necessarily to the minimal path deviation.

To this end we describe the time-wise progress of θ via an additional ODE, which is denoted as *timing law*,

$$\dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t) \quad = \mathbf{l}(\mathbf{z}(t), v(t)), \quad \mathbf{z}(0) = (\theta_0, 0)^T \tag{6a}$$

$$\theta(t) = z_1(t). \tag{6b}$$

It is important to note that the time evolution of θ can be controlled via the *virtual* input $v : [0, \tau] \to \mathbb{R}$. Using the timing law an approximate solution to Problem 1 can be obtained via the solution of a modified optimal control problem. This means we want to minimize a cost functional which is given by

$$J\left(\mathbf{x}_0, \mathbf{u}(\cdot)\right) \quad = \quad \int_0^{\tau} \underbrace{\left\|(\mathbf{e}_{\mathcal{P}}^T(t), \dot{\mathbf{e}}_{\mathcal{P}}^T(t))\right\|_{\mathbf{Q}_e}^2}_{\text{path deviation}} + \underbrace{\left\|(z_1(t), z_2(t)) - (\theta_r, \dot{\theta}_r)\right\|_{\mathbf{Q}_z}^2}_{\text{reference behavior}} + \underbrace{\left\|(\mathbf{u}^T, v, \dot{T})\right\|_{\mathbf{R}}^2}_{\text{regularization}} dt, \tag{7a}$$

where the weighting matrices $\mathbf{Q}_e$ and preferably $\mathbf{R}$ are positive semi-definite, and $\|\mathbf{Q}_e\| \gg \|\mathbf{R}\|$. The first term of the cost functional penalizes path and attitude deviations and the second one penalizes the distance to the end of the path, or to a desired speed along the path. One should note that in the first term of (7a) we penalize the path error $\mathbf{e}_{\mathcal{P}}(t) = \mathbf{h}(\mathbf{x}) - \mathbf{p}(z_1) = \mathbf{p}_r(t) - \mathbf{z}_1(t)$ as well as its time derivative $\dot{\mathbf{e}}_{\mathcal{P}}(t)$. By including the time derivative we avoid solutions which oscillate around the path. The last term is added as regularization term, i.e., to avoid chattering of the thrust trajectory $T(t)$, the system inputs $\boldsymbol{\delta}(t)$, or the path evolution $\theta(t)$. Essentially, this term improves numerical properties of the optimal control problem (OCP) and leads to smoother results.

The yaw angle is fixed to the path parameter using an algebraic equation and is commanded such that the heading is approximately tangential (for details see Appendix B). The approximation and the delay of the reference dynamics thus cause a side-slip angle. To achieve side-slip-free flights, the heading command has to be introduced as decision variable and the heading deviation from the tangential direction has to be penalized in (7a), which might be considered in future work.

Using this cost functional the OCP to be solved is given by

$$\underset{u(\cdot), v(\cdot)}{\text{minimize}} \quad J\left(\mathbf{x}_0, \mathbf{u}(\cdot)\right) \tag{7b}$$

[†] Exceptions are differentially flat systems, which allow to exploit the structure of the ODE (2), cf. Faulwasser et al.[8,15]

subject to the dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{7c}$$

$$\dot{\mathbf{z}}(t) = \mathbf{l}(\mathbf{z}(t), v(t)), \quad \mathbf{z}(0) = (\theta_0, 0)^T \tag{7d}$$

$$\dot{\mathbf{e}}_{\mathcal{P}}(t) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) - \frac{\partial \mathbf{p}}{\partial z_1} z_2(t). \tag{7e}$$

Furthermore, the constraints on states and inputs of (2), as well as additional constraints are considered

$$\forall t \in [0, \tau] : \mathbf{x}(t) \in \mathcal{X}, \quad \mathbf{u}(t) \in \mathcal{U} \tag{7f}$$

$$\forall t \in [0, \tau] : z_2(t) = \dot{\theta}(t) \geq 0 \tag{7g}$$

$$\mathbf{z}(\tau) = (\theta_1, 0)^T. \tag{7h}$$

The dynamic constraints (7c-e) include the reference model (2), the path parameter dynamics (6), and the path error ODE (7e). Besides the input and state constraints (7f) we consider the forward motion condition in (7g). Details on the considered input and state constraints can be found in Appendix A. Finally, (7h) expresses that we want to stop at the end of the path $\mathcal{P}$.[‡] Solving OCP (7) over the complete horizon $[0, \tau]$ can be challenging, since—depending on the length of the considered flight path—the time τ needed to reach the end of the path might be large. Furthermore, the end time is usually not known a priori, and should be considered as an additional decision variable of the optimization.

One might attempt to solve the OCP (7) by adding the time to reach the end of the path as a decision variable—i.e., to reformulate the OCP as a free end time optimal control problem. This would, however, lead to an increased computational burden caused by the free end time τ, cf. Bryson.[17] In order to avoid these difficulties we propose solving the OCP in a receding horizon fashion, i.e., we split the horizon $[0, \tau]$ into smaller time intervals $[k\Delta, k\Delta + \tau_p]$, where the index parameter is $k \in \{0, 1, \ldots, \hat{k}\} \subset \mathbb{N}_0$, Δ is the time interval and τ_p the prediction horizon. For each value of k we obtain an optimal input $\mathbf{u}_k^\star : [k\Delta, k\Delta + \tau_p] \to \mathcal{U}$. From each solution $\mathbf{u}_k^\star(t)$ we only consider the first part, i.e., the solution within the open interval $[k\Delta, (k+1)\Delta)$. In other words, the feedforward control $\mathbf{u}_{\mathcal{P}} : [0, \tau] \to \mathcal{U}$ to be applied to the MRAC-controlled helicopter is obtained by the concatenation of the solutions $\mathbf{u}_k^\star(t)$

$$\mathbf{u}_{\mathcal{P}}(t) = \begin{cases} \mathbf{u}_0^\star(t) & \text{if } t \in [0, \Delta) \\ \mathbf{u}_1^\star(t) & \text{if } t \in [\Delta, 2\Delta) \\ \vdots & \\ \mathbf{u}_{\hat{k}}^\star(t) & \text{if } t \in [(\hat{k}-1)\Delta, \hat{k}\Delta] \end{cases} .$$

The end time τ—the time when the final path point $\mathbf{p}(\theta_1)$ is reached—is given by $\tau = \hat{k}\Delta$. This end time is obtained while solving the sequence of optimization problems, i.e., one interrupts this sequence as soon as the path parameter $\theta(k\Delta)$ is sufficiently close to θ_1 and $\mathbf{h}(\mathbf{x}(k\Delta))$ is close to $\mathbf{p}(\theta_1)$. In essence this approach can be understood as obtaining the feedforward controls via the simulation of a model predictive path-following controller which is based on the OCP (7). The recomputation time $\Delta > 0$ and the horizon $\tau_p > \Delta > 0$ can, respectively, be understood as sampling time and prediction horizon of the simulated predictive controller. For details on predictive path following we refer to Faulwasser.[8–10]

These computations of the desired paths are performed offline via an open source C++ implementation of a direct multiple shooting algorithm.[18] The feedforward control $\mathbf{u}_{\mathcal{P}}(t)$ of each path is stored in a library of look-up tables, and read from these tables during the flight tests. We point out that via this receding horizon computations we merely compute an approximate solution to Problem 1.

IV. Simulation and Flight Test

In this section we present simulation and flight test results of two flight paths using the dynamic optimization-based approach. The simulation results are achieved using the environment of the ARTIS family. The next section focuses on the simulation of the two nonlinear example paths and the achievable accuracy. Subsequently, flight test results of on example will be outlined, which confirm the applicability of this approach and also show high accuracy path following.

[‡] If following a closed curve with a prescribed velocity is desired the constraint (7h) should be dropped.

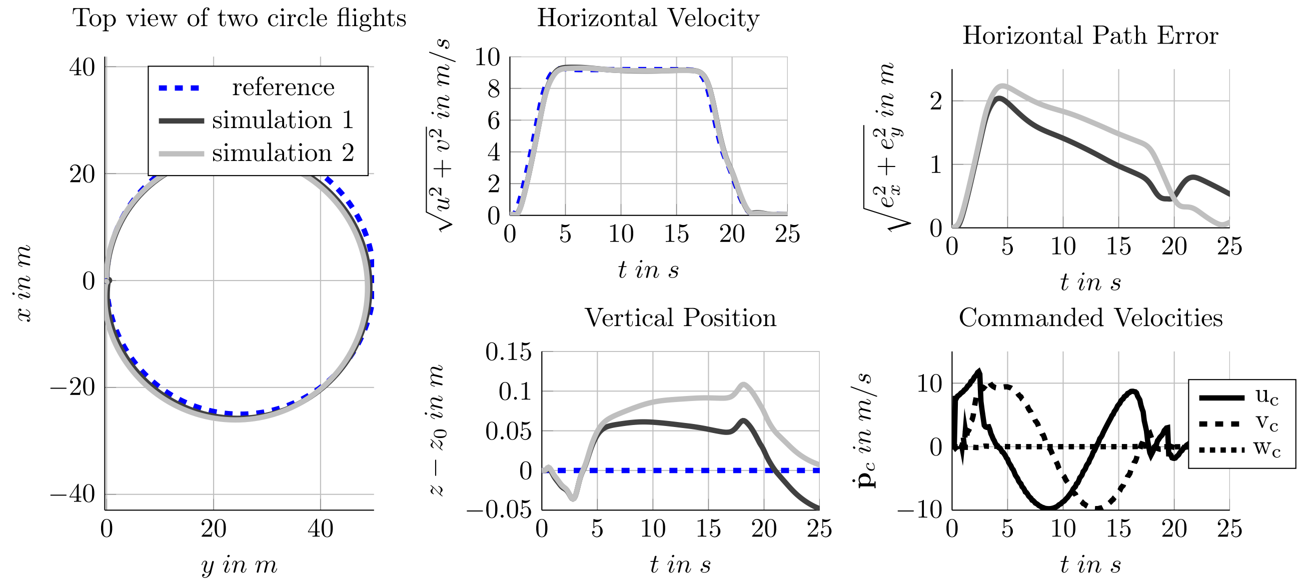

Figure 4. Simulation results of the circle path: nonlinear reference model for optimization and no additional position feedback.

A. Simulation Results

The simulation environment which is used for evaluation of the optimization approach consists of the complete unmanned aircraft system. The ground control station is in the loop as well as mission management which coordinates missions, the flight controller, sensor emulation, and the sensor fusion containing an extended Kalman filter. The flight mechanical model is based on a linear hover model including flapping dynamics which has been proven to be sufficiently close to the real flight behavior of the helicopter, cf. Lorenz.[7,19] The software components in simulation are identical to the ones implemented on-board the helicopter.

In order to test the proposed strategy we use two different test paths—a planar circle and a three-leaved clover leaf. The parameterizations of these curves are listed in Table 1 in Appendix C. In Table 1 we also list the numerical parameters τ_P and Δ which were adjusted due to the different difficulties of the two paths to achieve satisfactory calculation time. The computations are performed using the cost functional (7a), where the weighting matrices

$$\begin{aligned} \mathbf{Q}_e &= \operatorname{diag}\left(q_e \cdot \mathbf{I}^{3,3},\ q_{\dot{e}} \cdot \mathbf{I}^{3,3}\right) \\ \mathbf{Q}_z &= \operatorname{diag}(q_\theta,\ q_{\dot{\theta}}), \qquad \mathbf{R} = \operatorname{diag}(r_{\ u_{1,2,3}},\ r_v,\ r_{\dot{T}}), \end{aligned}$$

are adjusted for each path. Here, $\mathbf{I}^{3,3}$ is the identity matrix of $\mathbb{R}^{3\times3}$, and diag denotes a (block-)diagonal matrix. The numerical values used in the computations can be found in Table 1 in Appendix C. For the major part of the paths we penalize the path deviation $\mathbf{e}$ and its time derivative $\dot{\mathbf{e}}$ as well as $\dot{\theta} = z_2$. In order to enforce that the helicopter stops at the end of the path, the weighting matrix Q_z is switched at the end of the path, i.e., we penalize the difference $\theta - \theta_1 = z_1 - \theta_1$ via $q_\theta > 0,\ q_{\dot{\theta}} = 0$ when the path parameter reaches a threshold close to the end of the path.

The following outlines the results of simulations at the example of the two paths, both having radii of 25 m. After performing the optimization, which results in time series of the velocity commands $\dot{\mathbf{p}}_c$, the simulation is performed without disturbances, and the results are investigated. The position feedback loop has not yet been closed to make the achievable quality of the path following clearly visible. In an actual application, the position feedback loop has to be closed.

Figure 4 shows the simulation results of the circle path. The vehicle starts in trimmed condition at the origin and flies the circle clockwise. It is important to note that this circle is not a trimmed circle as it contains the transients from hover to forward movement along the path as well as the deceleration. The figure shows the top view of two simulated flights. The optimization assumed the initial attitude in north (x) direction exactly. However, the flights have actually been started with yaw angles of 3° and 4° respectively caused by the time history of the previously performed flight. The velocity profile is also depicted as well as the absolute deviation of the altitude with respect to the beginning of the maneuver. The horizontal path error illustrates the magnitude of the difference of the actual position and the reference position on the path at that particular point in time. Finally, the commanded flight-path velocities $\dot{\mathbf{p}}_c = (u_c, v_c, w_c)^T$ are also displayed to evaluate their smoothness. As the commands are identical for both flights they are depicted only once.

The achieved absolute horizontal accuracy is about 2 m. The two flown paths are so close to each other that no difference is apparent in the figure. The slight difference between the two flights is caused by different starting conditions of the yaw angle. The drift in height is neglectable. The velocity commands have some more aggressive movement in the acceleration and deceleration phases, in order to quickly reach the maximum velocity of 9 m/s and stop at the correct position. The control surface deflections remain sufficiently smooth and stay within the allowable range, as these had been introduced as constraints of the optimization.

Figure 5 presents results of a 3-leaved clover leaf. The vehicle starts at the center of the clover leaf and begins flying southwards. Again the position error in vertical direction is marginal. In this case, the starting yaw angle is about the same for both flights resulting in equivalent path errors of around 1 m. The velocity commands have more movement as the path is more complex than in the first example, yet they remain smooth. Only in the acceleration part the command is a bit more aggressive in order to achieve a quick convergence towards the maximum velocity of 8 m/s.

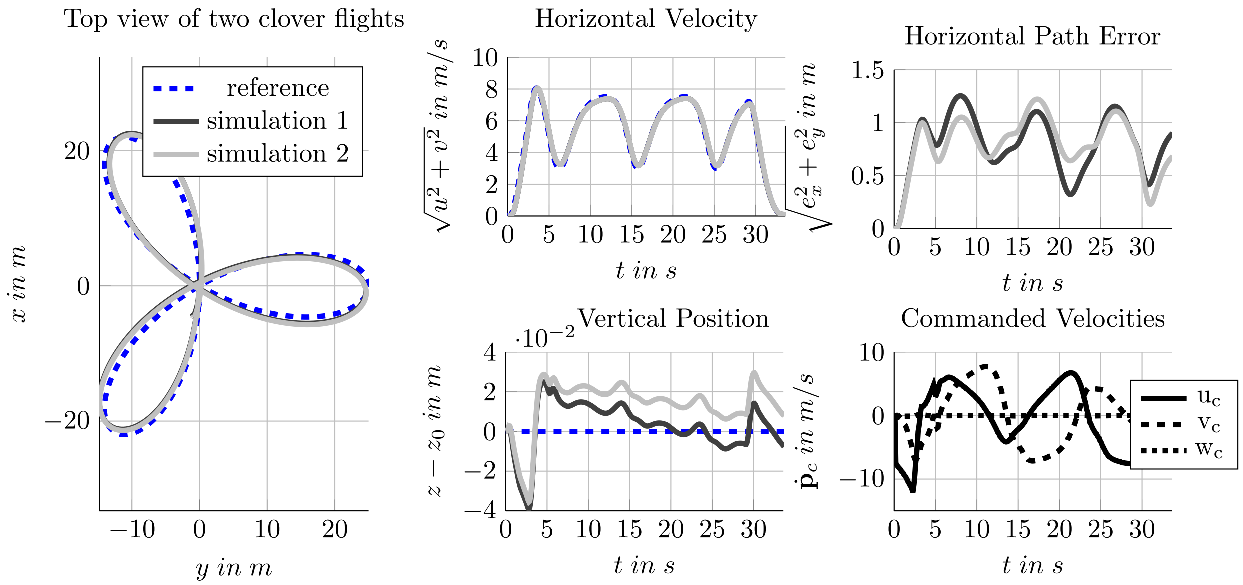

Figure 5. Simulation results of the clover leaf path: nonlinear reference model for optimization and without position feedback.

B. Flight Test

This chapter shows the flight test results of the circle path. Figure 6 presents two flights in the same way as the simulation results. The two flights have been performed in similar conditions around one minute apart. Wind conditions were measured to be about 4 m/s in mean velocity and up to 8 m/s during gusts. The flight was started from hovering condition. The helicopter is oriented approximately northwards initially (yaw angles of 5° and 6° respectively) and the circle is commanded clockwise after the helicopter was hovering calmly for a few seconds. Again, the position feedback loop remains open.

The commanded velocities are identical to the ones presented in Section IV.A. As can be seen from the top view of the figure, the resulting path itself is a circle. The direction of the path at the start is directed a few degrees in east direction. The plot of the horizontal path error shows that it remains below 5 m. The vertical path error remains below 2 m. The figure showing the velocity profile displays a close match. During first part of the circle, the velocity is slightly below the optimal. After completion of half the circle, the

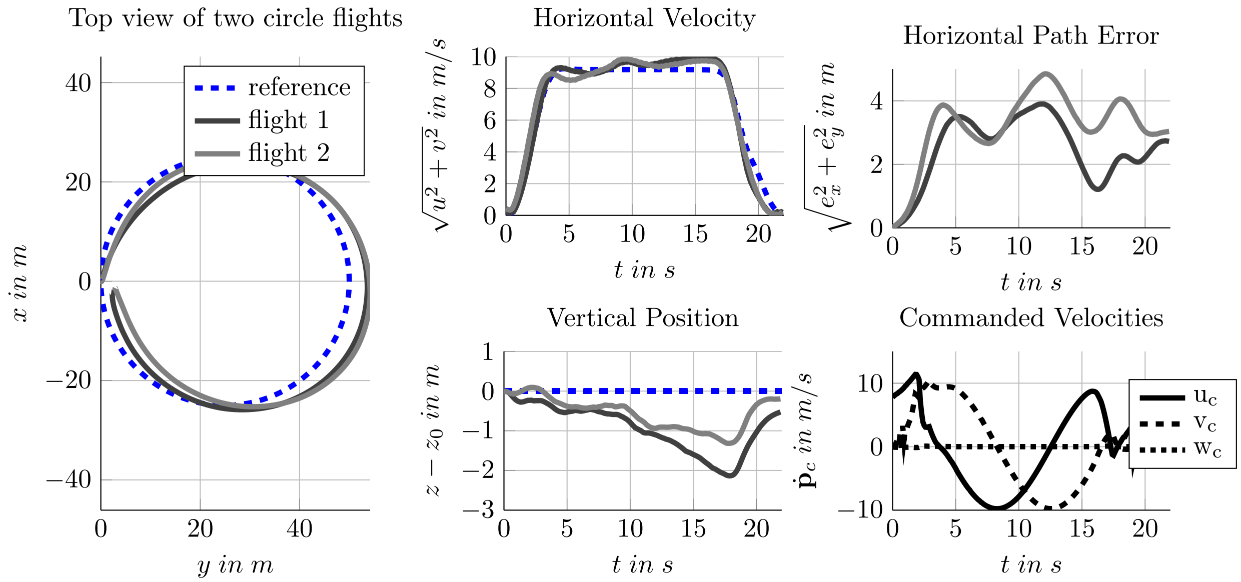

Figure 6. Flight test results of the circle path without position feedback.

velocity is slightly above the optimal.

The difference to the horizontal reference velocity profile is caused by the wind conditions. The controller regulates the inertial velocity. As can be seen from the horizontal velocity at the beginning of the circle, the helicopter has to overcome additional drag. This drag is caused by the increased airspeed induced by the wind. On the opposite side of the circle, the airspeed is reduced causing a decrease in drag. Consequently, the inertial velocity of the helicopter increases. This effect is also one of the main causes for the path deviation. An additional cause is found to be the accuracy of the sensor fusion. The used fusion algorithm currently exhibits a yaw-dependent roll angle estimation error. This estimation error can be up to 4°. Flight tests of the clover leaf thus resulted in higher path errors in the range of 10 m. The roll estimation error has a higher impact on the clover leaf than on the circle as it has significantly more transients in the roll angle. It is apparent that flying these nonlinear paths has higher accuracy requirements regarding the sensor fusion than linear paths. Additionally, the differences in the initial conditions of the vehicle including yaw and trim angles in comparison to the ones assumed during optimization have an impact on the tracking accuracy.

The flight tests proved that the proposed approach is feasible and results in good path tracking performance even without position feedback. We expect that improvements will be achieved by introduction of a position feedback loop for robustness against external disturbances.

V. Conclusion

This paper presents a novel approach to tackle high-accuracy path-following problems for unmanned helicopters. In particular, an optional module extends a model reference adaptive controller that is used to optimize the time-wise progress on the path without changing the controller's architecture and stability properties. In contrast to existing approaches the progression on the path is optimized using the reference model rather than the vehicle dynamics. The optimization procedure is explained and used on two different planar curves, a circle and a three-leaved clover leaf. Simulation results for these examples show that in the wind-free case a path following accuracy of less than 2 m could be expected. Flight test results proved the applicability of the presented approach. For the circle path an accuracy of below 5 m could be shown under the influence of wind. In order to improve this result further and to achieve robustness against external disturbances an additional position feedback will be introduced.

At this point, the optimization has been performed on ground. Next steps will include a more efficient formulation using this approach, which enables an on-board optimization. Furthermore, the yaw angle is yet excluded from the degree of freedom for the optimization. So far, the yaw command is a function of the path parameter. Considering the yaw angle will enable optimization goals for desired attitudes. Lastly, the optimization might be used for curves which include a height profile to enable full 3d paths.

Appendices

A. Input and State Constraints of the Optimization

The input constraints of the helicopter model (2) are

$$\mathcal{U} = \left\{ \dot{\mathbf{p}}_c \in \mathbb{R}^3 \mid \left\| \dot{\mathbf{p}}_c \right\|_2 \leq 12 \frac{\mathrm{m}}{\mathrm{s}} \right\}. \tag{8a}$$

The state constraints of the helicopter model (2) consist of two parts

$$\mathcal{X} = \mathcal{X}_x \cap \mathcal{X}_\delta,$$

where the first set $\mathcal{X}_x \subset \mathbb{R}^{15}$ refers to the constraints which are directly expressed in terms of the state $\mathbf{x}$ from (3).

$$\mathcal{X}_x = \left\{ (T, \dot{T}, \boldsymbol{\omega}^T, \dot{\boldsymbol{\omega}}^T, \mathbf{p}_r^T, \dot{\mathbf{p}}_r^T, \underline{q}_r{}^T)^T \mid \mathbf{t}' + \mathbf{g} \in \left([-4, 4],\ [-4, 4],\ [-4, 4] \right)^T \frac{\mathrm{m}}{\mathrm{s}^2},\ \dot{T} \in [-3, 3] \frac{\mathrm{m}}{\mathrm{s}^3} \right\}. \tag{8b}$$

Furthermore, the actuator deflections of the MRAC controller have to be limited. For sake of convenience these constraints are directly expressed in terms of the normalized actuator deflections $\boldsymbol{\delta}$

$$\mathcal{D} = \left\{ \boldsymbol{\delta}_c \;\middle|\; \boldsymbol{\delta}_c \in \left([-0.8, 0.04],\ [-0.4, 0.4],\ [-0.4, 0.4],\ [-0.4, 0.4] \right)^T \right\}. \tag{8c}$$

The deflections $\boldsymbol{\delta}$ can be estimated as function of the states $\mathbf{x}$ from (3) using the helicopter model of the inversion. This estimation leads to the second set of state constraints $\mathcal{X}_\delta$.

B. Yaw Command Approximation

In order to completely specify the attitude command for the inner loop, the yaw angle has to be specified. So far, the yaw command is fixed to the path parameter θ using an algebraic relation and is not a degree of freedom for the optimization. The complete command quaternion follows using the thrust ensuring attitude according to (1d) by

$$\underline{q}_c = \underline{q}_\psi \underline{q}_t. \tag{9}$$

The derivation of the yaw quaternion is beyond the scope of this paper. In words however, the task is to find a quaternion $\underline{q}_\psi(\psi_c)$ that does not compromise the thrust direction and sets the yaw angle component of $\underline{q}_c$ to the commanded yaw angle ψ_c, which is specified by the mission requirements. Details on this topic can be found in Dauer et al.[20] In this context the yaw angle has been commanded so that the heading is approximately tangential to the path itself.

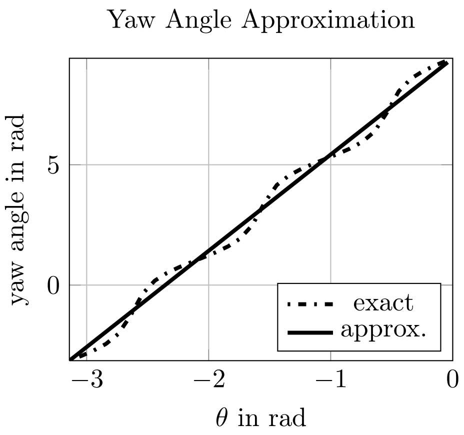

Figure 7. Comparison of the clover yaw command approximation and the exact.

In the case of the three-leaved clover leaf path, the yaw angle itself is analytically complex, thus a linear approximation has been used for ψ_c as shown in Figure 7. In future work, the yaw angle will be used as system input for the path optimization enabling more accurate yaw angles and more sophisticated mission-based specifications.

C. Path Parameterization and Optimization Parameters

Table 1. Considered paths, their parameterizations, and optimization parameters.

Path	Parametrization $\mathbf{p}(\theta)$	Num. Parameters
planar circle	$\mathbf{p}_0 + \rho \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z_0 \end{pmatrix},$ $\mathbf{p}_0 \in \mathbb{R}^3,\ z_0, \rho \in \mathbb{R},\ \theta \in [-\pi, \pi]$	$\tau_p = 1.5\,s,\ \Delta = 0.1\,s$ $q_e = 10^2,\ q_{\dot{e}} = 20$ $q_\theta = 0,\ q_{\dot{\theta}} = 10^2$ $r_{u_{1,2,3}} = 10^{-3},\ r_{\dot{T}} = 5 \cdot 10^{-3},\ r_v = 0$
planar clover	$\mathbf{p}_0 + \rho \sin(3\theta - \frac{3}{2}\pi) \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z_0 \end{pmatrix},$ $\mathbf{p}_0 \in \mathbb{R}^3,\ z_0, \rho \in \mathbb{R},\ \theta \in [-\pi, 0]$	$\tau_p = 0.5\,s, \Delta = 0.05\,s$ $q_e = 10^2, q_{\dot{e}} = 20$ $q_\theta = 0, = q_{\dot{\theta}} = 10^3$ $r_{u_{1,2,3}} = r_{\dot{T}} = 10^{-2}, r_v = 0$

References

[1] Dadkhah, N. and Mettler, B., "Survey of Motion Planning Literature in the Presence of Uncertainty: Considerations for UAV Guidance," *Journal of Intelligent & Robotic Systems*, Vol. 65, No. 1-4, 2012, pp. 233–246.

[2] Huang, G. Q., Lu, Y. P., and Nan, Y., "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Science China Technological Sciences*, Vol. 55, No. 9, 2012, pp. 2538–2560.

[3] Ioannou, P. and Sun, J., *Robust Adaptive Control*, Prentice Hall, Upper Saddle River NJ, 1996.

[4] Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems*, Dover Publications, New York, 2005.

[5] Hovakimyan, N., Kim, N., Calise, A. J., Prasad, J. V. R., and Corban, E., "Adaptive Output Feedback for High-Bandwidth Control of an Unmanned Helicopter," *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2001, AIAA 2001–4181*, Montreal,Canada, 2001.

[6] Johnson, E. N. and Kannan, S. K., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 524–538.

[7] Lorenz, S. and Dauer, J. C., "Evaluation of Time-Shifted Feedforward Control for Unmanned Helicopter Path Tracking," *Proceedings of AIAA Guidance, Navigation, and Control Conference, AIAA -2012-4609*, Minneapolis, Minnesota, 2012.

[8] Faulwasser, T., *Optimization-based solutions to constrained trajectory-tracking and path-following problems*, No. 3 in Contributions in Systems Theory and Automatic Control, Shaker, Aachen, Germany, 2013, ISBN 978-3-8440-1594-2.

[9] Faulwasser, T., Kern, B., and Findeisen, R., "Model predictive path-following for constrained nonlinear systems," *Proc. 48th IEEE Conf. on Decision and Control held jointly with the 2009 28th Chinese Control Conf. CDC/CCC 2009*, Shanghai, China, Dec. 15–18, 2009, pp. 8642–8647.

[10] Faulwasser, T. and Findeisen, R., "Ein prädiktiver Ansatz zur Lösung nichtlinearer Pfadverfolgungsprobleme unter Beschränkungen," *at - Automatisierungstechnik*, Vol. 57, No. 8, 2009, pp. 386–394.

[11] Park, S., Deyst, J., and How, J. P., "A New Nonlinear Guidance Logic for Trajectory Tracking," *Proceedings of AIAA Guidance, Navigation, and Control Conference,AIAA 2004-4900*, Providence, Rhode Island, 2004.

[12] Lorenz, S., "Open-Loop Reference System for Nonlinear Control Applied to Unmanned Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 259–269.

[13] Mettler, B., *Identification Modeling and Characteristics of Miniature Rotorcraft*, Springer, New York, 2002.

[14] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., "Constrained model predictive control: Stability and optimality," *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814.

[15] Faulwasser, T., Hagenmeyer, V., and Findeisen, R., "Optimal Exact Path-Following for Constrained Differentially Flat Systems," *Proc. of 18th IFAC World Congress*, Milano, Italy, 2011, pp. 9875–9880.

[16] Topogonov, V., *Differential Geometry of Curves and Surfaces - A Concise Guide*, Birkhäuser, Boston, 2006.

[17] Bryson, A., *Dynamic Optimization*, Addison-Wesley, Menlo Park, California, 1999.

[18] Houska, B., Ferreau, H., and Diehl, M., "ACADO toolkit – An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, Vol. 32, No. 3, 2011, pp. 298–312.

[19] Lorenz, S. and Adolf, F., "A Decoupled Approach for Trajectory Generation for an Unmanned Rotorcraft," *Advances in Aerospace Guidance, Navigation and Control*, edited by F. Holzapfel and S. Theil, Springer, Berlin, 2011, pp. 3–14.

[20] Dauer, J., Lorenz, S., and Dittrich, J. S., "Advanced Attitude Command Generation for a Helicopter UAV in the Context of a Feedback Free Reference System," *AHS International Specialists' Meeting on Unmanned Rotorcraft*, Tempe, AZ, USA, January 2011.