Nonlinear Model Predictive Flight Path Control for an Unmanned Powered Paraglider

Fabian Binz, Philipp Hartmann and Dieter Moormann

1 Introduction

Parafoil and payload systems offer high payload capacities, robustness and ease of deployment. Thus, they are used in both military and civil applications like remote sensing, aerial survey or airdrop supply systems. Besides their versatility, powered paragliders (PPGs) are interesting objects of investigation because of their unique flight dynamics. The center of mass being suspended below the parafoil leads to a high roll and pitch stability. At the same time, the high wind-to-airspeed ratio makes the aircraft susceptible to wind and gusts.

A large body of research on parafoil and payload systems focuses on modeling their complex, nonlinear dynamics. References [2, 4, 13] provide multiple detailed, nonlinear system descriptions with 8 or 9 degrees of freedom (DOF). These models are derived on the basis of a two-body system description with a joint representing the flexible connection between the payload and the parafoil. Reference [13] then continues to derive simpler models with only 6 DOF, which are more suitable for controller design. Assuming that the lateral and longitudinal motion are decoupled, model complexity can be further reduced to three DOF [5]. The parameters used in these models are either known or obtained using system identification techniques [8].

Concerning the flight guidance of parafoil and payload systems, several approaches can be found in the literature. At the highest level they share a common structure, namely, a separation between heading control and flight path guidance. The heading control is achieved using (nonlinear) model predictive control (MPC) [10, 11] or classic PID control [7].

To track the reference flight path, several control schemes have been proposed. Most commonly, the distance to a point on the reference flight path is mapped to a commanded heading, see [11]. More recently, the increased computational power of modern avionics allowed the use of MPC directly for flight guidance. In this regard,

F. Binz (✉) · P. Hartmann · D. Moormann
Institute of Flight System Dynamics, RWTH Aachen University,
Aachen, Germany
e-mail: binz@fsd.rwth-aachen.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_20

Ref. [12] presents experiment results of a parallel implementation of an MPC on a graphics processor. Here, the parallel processing capabilities are used to find a flight path which is robust against uncertainties introduced by wind gusts. Similarly, [6] presents a parallel implementation of nonlinear model predictive control (NMPC) on a FPGA-based flight computer enabling sample rates of 10 Hz.

Against this background, the control scheme for the horizontal flight path guidance presented here focuses on simplicity and ease of implementation. Especially in outer flight guidance control loops the inherent properties of PPGs can easily lead to large control deviations. One reason for this is the fact that classic feedback controllers typically only *react* to changes of the reference input. The controller proposed here is a predictive controller and thus can *act* according to predicted, future changes of the reference input. It can be argued that this property makes predictive controllers especially suited for flight guidance problems, since the reference flight path is typically known ahead of time. While the results gathered in this paper were obtained with a PPG, they are applicable to other parafoil and payload systems.

The paper is structured as follows: First, Sect. 2 gives an overview of the characteristics of PPGs along with a brief introduction into NMPC. Based on this, Sect. 3 presents the conceptual approach of the control scheme, then Sect. 4 discusses several aspects of its realization. The results of evaluating the controller in simulations and flight tests are summarized in Sect. 5.

2 Fundamentals

This section introduces basics concerning the flight dynamics of PPGs and will highlight some challanges these characteristics pose for accurate flight path guidance. Since the control scheme used here uses a black-box model of the plant, no analytical description of the flight dynamics of a PPG will be given. For a discussion of different modelling techniques the reader may refer to [8, 13, 14].

The discussion of model predictive control will concentrate on its conceptual ideas and keep formulas to a minimum. Comprehensive material on (nonlinear) model predictive control may be found in [3, 9].

2.1 Powered Paragliders

Powered paragliders belong to the class of parafoil and payload systems, which are characterized by the payload being suspended with lines below the parafoil, see Fig. 1. The only means of steering is the deflection of the trailing edge of the parafoil. The deflection is accomplished using two lines, each connected to one side of the trailing edge, often called brakes. Pulling one line alters the parafoils shape on the correspondig side. This asymmetric brake deflection creates a yawing moment, which in turn leads to a yawing rate. Pulling both brakes symmetrically primarily leads to an

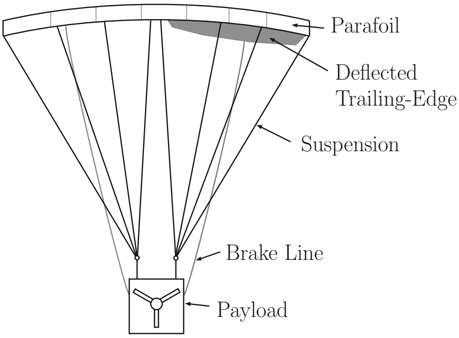

Fig. 1 Conceptual overview of a powered paraglider

increase in drag and can be used to control the airspeed and thus the glide path of the parafoil. Additionally, *powered* paragliders are equipped with a propulsion system, which is located in the same frame as the actual payload. Due to the suspension below the parafoil, an increase in thrust primarily increases the incidence angle of the parafoil and thus the vertical velocity. In comparison, the effect on the horizontal airspeed is small.

Due to their construction, PPGs typically exhibit high stability in both yaw and pitch. The long suspension lines allow relative motion between the parafoil and the payload. Because in the PPG considered here, all sensors are located in the payload, this relative motion has to be taken into account during controller design. With regard to heading control the relative motion in form of a self-induced yawing oscillation has to be considered.

Since PPGs are often used in harsh environments robust sensing equipment is beneficial. To that end, the control scheme presented here only requires heading information of the payload, aswell as ground speed and position. The heading can be obtained using off-the-shelf devices, while the ground speed and position are obtained by a GPS device. Notably, no airspeed information is required. This small number of required sensors is sufficient for our control scheme because of two assumptions which are made concerning the flight mechanics of PPGs: a constant airspeed and a negligible stationary side-slip.

2.2 Model Predictive Control

Model predictive control (MPC) is an umbrella term for a family of control algorithms which offer a conceptually simple solution to a wide range of control problems. While the core concept is simple, the implementation of an MPC often requires substantial processing power. This hindered the application of MPC in fast control loops, as those typically found in aircraft, in the past.

All MPC algorithms share the idea, that control action is based on predictions of the state trajectory of the plant. For different future control input sequences, different trajectories of the plant are predicted. These predicted trajectories are rated using a cost function. Finding the optimal control input sequence with regard to the cost function constitutes solving an optimization problem. In a closed-loop setting this optimization is carried out in every timestep and only the first element of the optimal input sequence is used as an actual input to the plant.

The complexity of solving the optimization problem is highly dependent on the model used. For linear models, analytical solutions can readily be derived. Nonlinear models on the other hand require nonlinear optimization algorithms, which typically require much processing power and a complex implementation.

One can identify four separate components which will be present in any MPC: the model, the observer, the cost function and the optimizer. A general discrete-time model is given by

$$x(k+1) = f\big(x(k), u(k)\big) \tag{1}$$

Here, the next state $x(k+1)$ is a (nonlinear) function f of the previous state $x(k)$ and the previous input $u(k)$. A sequence $x(\cdot)$ is called the (state) trajectory of the system. In general, the state x cannot be measured directly, but has to be calculated based on the measurable outputs and inputs $y(k) = g(x(k), u(k))$. Thus, most model predictive control schemes will include an observer.

The predicted trajectories are rated using a cost function of the form

$$J_H = \sum_{k=0}^{H-1} \ell(x_u(k), u(k)) \tag{2}$$

The index H denotes the prediction horizon, x_u is the state trajectory obtained using the input sequence $u(\cdot)$ and ℓ are the *pointwise* costs. Often, additional terms are included to improve or proof stability, but for this work the simple formulation in (2) is sufficient.

One of the advantages of MPC compared to other control schemes is its ability to incorporate constraints. These can be introduced at several points, either as hard constraints or as soft constraints. Hard constraints, as those often found in inputs, can be easily included as part of the model or the optimizer. Soft constraints (e.g. a flight state, which should be avoided) can be included by assigning a large cost to the corresponding state or input.

Given the cost function, the model and the currently observed state, the optimizer has to find the input sequence $u(\cdot)$ which minimizes the cost function J_H over the prediction horizon H and at the same time satisfies the given constraints. The optimal control problem (OCP) thus is:

$$\begin{aligned}
&\text{Find } u^*(\cdot) = \operatorname*{argmin}_{u(\cdot)} J_H = \operatorname*{argmin}_{u(\cdot)} \sum_{k=0}^{H-1} \ell(x_u(k), u(k)) \\
&\text{where } x_u(0) = x_0 \qquad\qquad\qquad\qquad\qquad\qquad \text{(OCP)} \\
&\text{and } x_u(k+1) = f(x_u(k), u(k))
\end{aligned}$$

while satisfying the given constraints.

Solving (OCP) yields an optimal control sequence $u^*(\cdot)$, where the first element provides the input to the plant during the next timestep.

Prediction Horizon The time covered by the prediction horizon H is $\Delta t = H \cdot T$. Since T effectively introduces a dead time into the control loop it is desirable to keep T small. However, for a fixed Δt the consequences of a smaller timestep T are twofold: First, we increase the number of timesteps we need to predict. Secondly, the computation time per timestep increases, because (OCP) needs to be solved for more variables. Due to this using an MPC quickly becomes infeasible for small timesteps T.

3 Conceptual Approach

A simplified block diagram of the horizontal flight path guidance is shown in Fig. 2. There are two control loops: the outer loop controls the horizontal flight path using a model predictive controller and generates heading commands for the inner loop, the heading controller. Dividing the controller into two loops, the "fast" heading controller loop and the "slow" flight guidance loop, has several advantages over a solution based on model predictive control only. The most apparent advantage is that it allows the reuse of an existing heading controller. At the same time, both controllers can be designed separately. From the viewpoint of the MPC, the heading controller is now part of the plant. One can achieve a mostly linear control response from commanded heading to actual heading by designing the heading controller correspondingly. Thus a linear model can be used to represent the heading-controlled

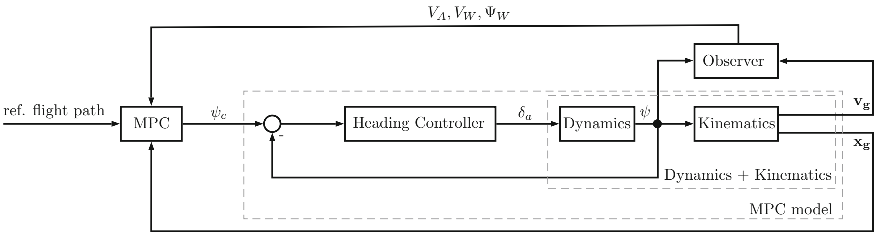

Fig. 2 MPC control loop

PPG. Another, more subtle effect is that, due to this division it is natural to let the “fast” heading controller run at a higher frequency than the “slow” flight path controller.

In addition to the linear model of the heading-controlled PPG, the underlying model of the MPC also includes the flight path kinematics, which are inherently nonlinear. Due to this, the optimization problem (OCP) also becomes nonlinear. To predict the future flight path, the current wind speed has to be estimated. This is done using a non-standard observer, which is only based on heading information and ground velocity measurements.

The reference flight path is constructed using straight-line segments. Turns are constructed in compliance with the maximum achievable turn rate and then approximated using multiple straight-line segments.

In the following sections, each component briefly presented above will be discussed in more detail.

3.1 Model of Heading-Controlled PPG

The model consists of two parts: the dynamics and the kinematics. The dynamics model the behaviour of the heading-controlled plant, thus the input is the commanded heading Ψ_c and the output is the actual heading Ψ. As mentioned before, the suspension of the payload leads to heading oscillations. While high-DOF models, which model the relative motion of the canopy and the payload, exist, considering these effects is not useful for the purposes of flight path guidance. This is due to the fact that the actuators are too slow to counteract the oscillations, thus they ideally should not react to them at all. Additionally, considering the flight path, heading oscillations about a setpoint will in effect still lead to an approximately straight-line path.

The model of the heading-controlled PPG is a transfer function obtained using black-box system identification, see Sect. 4.2. It is of the form

$$G_{\Psi\Psi_c} = \frac{1 + n_1 s + n_2 s^2 \dots}{1 + d_1 s + d_2 s^2 + \dots} \tag{3}$$

Note that, while most of the coefficients of the numerator and denominator polynomials need to be estimated, we fix $n_0 = 1$. In other words, the static-gain of $G_{\Psi\Psi_c}$ is 1, because we assume that the heading controller will achieve zero stationary control deviation. Additionally, $G_{\Psi\Psi_c}$ shall be strictly proper, since there should be no direct feed-through. For the actual discrete-time implementation, the dynamics (3) need to be discretized.

The kinematic model is based on the assumptions that airspeed, wind speed and wind direction only vary slowly and that the side-slip of the PPG is negligible. Thus the kinematics are given by

$$\begin{aligned} \mathbf{V_g} &= \mathbf{V_A} + \mathbf{V_W} \\ \mathbf{x_g} &= \int \mathbf{V_g}\, dt \end{aligned} \tag{4}$$

Here, $\mathbf{V_A}$ and $\mathbf{V_W}$ denote the aerodynamic and the wind speed respectively. As before, (4) needs to be discretzied for the actual implementation.

The complete model consists of the dynamics of the heading controller (3) and the kinematics (4).

To predict a trajectory of a system the initial states need to be known. The states of the kinematic model are comprised of the position $\mathbf{x_g}$, the airspeed V_A, the wind speed V_W and the wind direction Ψ_W. The states of the dynamic model are dependent on the actual dynamic model chosen in the system identification process. In the case of a discrete-time transfer function model without direct feed-through the output is a function of the previous outputs and the previous inputs:

$$\Psi(k) = -a_1\Psi(k-1) - a_2\Psi(k-2) - \ldots + b_1\Psi_c(k-1) + b_2\Psi_c(k-2) + \ldots$$

Thus the previous inputs and outputs can be considered states of the model. Akin to internal model control (IMC), the previous outputs $\Psi(k-1), \Psi(k-2), \ldots$ are obtained by updating an internal model at each timestep k with the previously chosen input $\Psi_c(k-1)$. This has the advantage, that the heading oscillations of the PPG do not affect the MPC model. However, this approach only works, if the plant and the model do not diverge over time. We assume that this is given, since the dynamics model has a static gain of 1 and the heading controller is designed accordingly. Using this technique effectively decouples the flight guidance loop from the dynamics of the PPG.

3.2 Observer

As mentioned in Sect. 3.1 the model state is given by

$$\mathbf{x} = \left[\mathbf{x_g}\ V_A\ V_W\ \Psi_W\ \Psi(k-1)\ \Psi(k-2)\ \ldots\ \Psi_c(k-1)\ \Psi_c(k-2)\ldots\right]^T \tag{5}$$

All state variables, but the airspeed V_A, wind speed V_W and wind direction Ψ_W, are either directly measurable or known. Based on the assumption of negligible side-slip, slowly varying airspeed, wind speed and wind direction (c.f. Sect. 3.1) the unknown state variables V_A, V_W and Ψ_W can be observed from available measurements.

We first introduce the ground speed in direction of the current heading $V_{g,\Psi}$, which can be calculated using the following formula:

$$V_{g,\Psi} = \left|\begin{pmatrix} V_{g,x} \\ V_{g,y} \end{pmatrix}\right| \cos(\chi - \Psi) = V_g \cos(\chi - \Psi) \quad \text{where} \quad \chi = \operatorname{atan2}(V_{g,y}, V_{g,x}) \tag{6}$$

Fig. 3 Velocity vectors

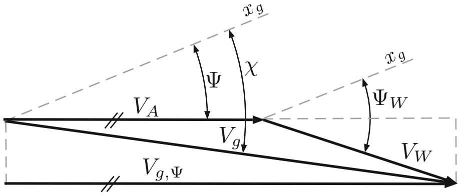

Figure 3 shows the relationship between $\mathbf{V_A}$, $\mathbf{V_W}$ and $\mathbf{V_g}$ and gives the following equation for $V_{g,\Psi}$:

$$V_{g,\Psi} = V_A + V_W \cos(\Psi - \Psi_W) \tag{7}$$

In principle, given a set of observations $V_{g,\Psi}$ and Ψ the unknown parameters V_A, V_W and Ψ_W can be determined using nonlinear regression. A more efficient approach is to introduce a nonlinear transformation such that the unknown parameters can be determined using linear regression:

$$\begin{aligned} V_{g,\Psi} &= V_A + V_W \cos(\Psi - \Psi_W) \\ &= V_W \cos(\Psi_W)\cos(\Psi) + V_W \sin(\Psi_W)\sin(\Psi) + V_A \\ &= V_{Wx} \cdot \cos(\Psi) + V_{Wy} \cdot \sin(\Psi) + V_A \end{aligned} \tag{8}$$

Equation (8) is linear in the parameters V_{Wx}, V_{Wy} and V_A. The transformation back to the original parameters is given by

$$V_W = \sqrt{V_{Wx}^2 + V_{Wy}^2} \tag{9}$$

$$\Psi_W = \text{atan2}(V_{Wy}, V_{Wx}) \tag{10}$$

For N observations the following system of equations can be formulated:

$$\begin{pmatrix} \cos(\Psi_1) & \sin(\Psi_1) & 1 \\ \cos(\Psi_2) & \sin(\Psi_2) & 1 \\ \vdots & \vdots & \vdots \\ \cos(\Psi_N) & \sin(\Psi_N) & 1 \end{pmatrix} \begin{pmatrix} V_{Wx} \\ V_{Wy} \\ V_A \end{pmatrix} = \begin{pmatrix} V_{g,\Psi,1} \\ V_{g,\Psi,2} \\ \vdots \\ V_{g,\Psi,N} \end{pmatrix} \Leftrightarrow \mathbf{Ax} = \mathbf{b} \tag{11}$$

The least-squares problem corresponding to Eq. (11) can be solved using standard algorithms. However, the choice of observations to include in the regression is not as obvious. Always choosing the last N observations can quickly lead to a case where $\mathbf{A}$ becomes singular, for example when the PPG is flying with a constant heading. In this case, all rows of $\mathbf{A}$ will be nearly equal and the least-squares problem becomes badly conditioned. Instead, we divide the heading range from -180 to 180° into N equal slices. For each slice, P observations (Ψ, $V_{g,\Psi}$) are recorded and thus a mean and a variance can be calculated. A large variance in a slice will likely be due to changes in the wind vector, thus for the regression this slice should have a smaller influence on the estimate than a section with low variance. By introducing the diagonal matrix $\mathbf{W} = \mathrm{diag}({}^{1}\!/_{\sigma_i})$, where σ_i is the standard deviation in the i-th slice, a new regression problem can be formulated:

$$\hat{\mathbf{b}} = \hat{\mathbf{A}}\mathbf{x} \quad \text{where} \quad \hat{\mathbf{b}} = \mathbf{W}\overline{\mathbf{b}} \quad \text{and} \quad \hat{\mathbf{A}} = \mathbf{W}\overline{\mathbf{A}} \tag{12}$$

$\overline{\mathbf{b}}$ and $\overline{\mathbf{A}}$ are defined using mean values in each slice as described above. The observer can be classified as an online least-squares estimator with directional forgetting [1].

3.3 Cost Function

The cost function penalizes deviations of a predicted flight path from the reference flight path. As stated in Eq. (2) the total cost is the sum of pointwise costs ℓ. The main goal is to minimize the distance to the reference flight path. We use the standard practice to minimize the squared distance, since it enables a more efficient implementation. In addition to penalizing the distance we also penalize deviations from the reference flight path azimuth. This prevents the optimizer from finding solutions which are close to the reference flight path, but opposite to the direction associated with the current line segment. To allow approaching the reference flight path in a 90° angle, this term is weighted with the inverse of the distance to the reference flight path. The pointwise cost function is thus

$$\ell(x, u) = \mu_1 d^2(x, x_{ref}(\cdot)) + \mu_2 \frac{\Delta\chi^2}{d(x, x_{ref}(\cdot))} \tag{13}$$

where the current fligh path azimuth is given by $\chi_i = \mathrm{atan2}(y_{g,i+1} - y_{g,i}, x_{g,i+1} - g_{g,i})$, $\mathbf{x_i} = (x_{g,i}, y_{g,i})$ is the i-th element of the predicted flight path and $d(x, x_{ref}(\cdot))$ denotes the distance from the reference flight path. Equation (13) introduces the additional weighting factors μ_1 and μ_2, which make the corresponding terms dimensionless. For the optimization, the relative weighting of both terms is essential, so μ_1 can be fixed to $1/\mathrm{m}^2$. μ_2 should be large enough to prevent the optimizer from choosing the wrong directions. Suitable values can be found in simulation studies. The cost function (13) only penalizes states, not inputs, since the hard input constraints are handled during optimization.

3.4 Optimizer

The optimizer's task is to minimize the cost function J_H (2) with regard to the input sequence $u(\cdot)$. Given the dynamics of the heading-controlled PPG we know that a maximum absolute commanded turn rate $|\dot{\Psi}_{c,\max}|$ can be achieved. This creates

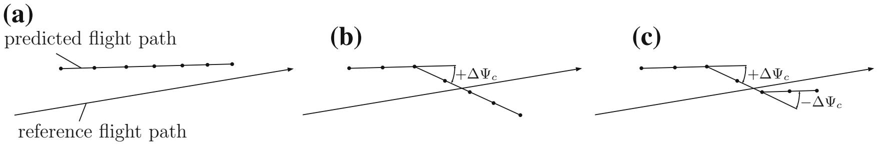

Fig. 4 Optimizer working principle

a constraint, which has to be satisfied during optimization. Since the optimization problem (OCP) is formulated in a discrete-time setting, the maximum absolute commanded turn rate $\dot{\Psi}_{c,\max}$ is replaced by a maximum change in the commanded heading $|\Delta\Psi_{c,\max}|$.

A common problem in solving (OCP) is that changes in the inputs have non-local effects, i.e. a change in the commanded heading affects the entire predicted flight path in the following time steps. A consequence of this property is that minimizing J_H by optimizing the input sequence at one point can easily lead to the situation depicted in Fig. 4b. Initially the pointwise cost ℓ decreases, but increases again after the predicted trajectory crosses the reference flight path. In total, the costs J_H might not be lower for the predicted flight path in Fig. 4b than for the initial path in Fig. 4a. Thus the predicted flight path in Fig. 4b will not be considered further during gradient-based optimization. An effective way to circumvent this problem is to introduce changes to the commanded heading not only at one, but at two points in the predicted flight path. This leads to the predicted flight path shown in Fig. 4c.

The optimization algorithm works by successively introducing opposing changes at two prediction points, denoted by indices i and j. The index i runs from 1 to $H-1$, while the index j runs from $i+1$ to H. For each pair (i, j) a small change $\Delta\Psi_c$ is introduced to the commanded heading (added at i, substracted at j and vice versa). This process continues as long as the total cost decreases, otherwise the algorithm advances to the next pair (i, j). The number of iterations per pair (i, j) is bounded, because the introduced change is limited to $\pm\Delta\Psi_{c,\max}$. Additionally, the total number of iterations is bounded, to ensure that the algorithm produces a result in the required time T. The number of index pairs (i, j) is in the order of $\frac{1}{2}H^2$. At each pair, the model has to be executed H times to give the predicted flight path. Thus the total complexity of the algorithm is of order H^3.

If it takes T_{once} seconds to step the model once and calculate the pointwise cost ℓ, predicting a complete flight path takes $H \cdot T_{\text{once}}$ seconds. In the worst case, the optimizer needs to evaluate $\frac{\Delta\Psi_{c,\max}}{\Delta\Psi} \cdot H^2$ complete trajectories. For a fixed prediction time Δt and an MPC timestep of T, the number of prediction steps H is $\frac{\Delta t}{T}$. The smallest MPC timestep is thus found to be

$$T_{\min} = \sqrt[4]{\frac{\Delta\Psi_{c,\max}}{\Delta\Psi_c}\Delta t^3 T_{\text{once}}} \tag{14}$$

The values of Δt and $\Delta\Psi_{c,\max}$ depend on the plant dynamics, while T_{once} and $\Delta\Psi_c$ are design variables.

4 Implementation

The following section discusses the implementation of the previously presented control scheme.

Technical Data The actual PPG used was a SUSI62 provided by Dr. H.-P. Thamm Geo-Technic (Fig. 5). Its technical data are summarized in the Table 1.

The flight control hardware was developed at the Institute of Flight System Dynamics of the RWTH Aachen University. Besides an integrated IMU and a GPS module it has two dedicated microcontrollers for running control algorithms. Both microcontrollers are clocked at 164 MHz and contain a floating point unit (FPU). Due to this architecture, the previously discussed division between heading control and flight path control could be easily transferred to the actual implementation. While the heading controller runs at a frequency of 200 Hz, the flight path controller runs only at 1 Hz, see Sect. 4.4.

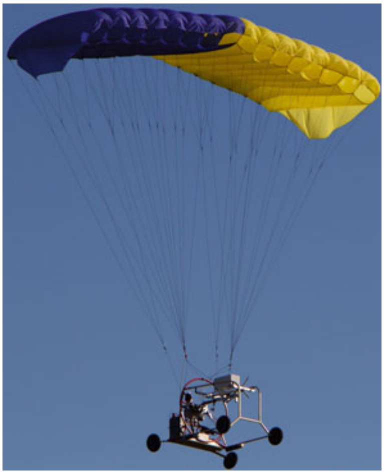

Fig. 5 The powered paraglider SUSI62

Table 1 Technical data

Wing area	6 m^2
Empty weight	$\approx$ 10 kg
Payload capacity	up to 5 kg
Airspeed	$\approx$ 8 m/s
Motor	62 cm^3, 2-stroke, gasoline
Flight time	up to 1 h

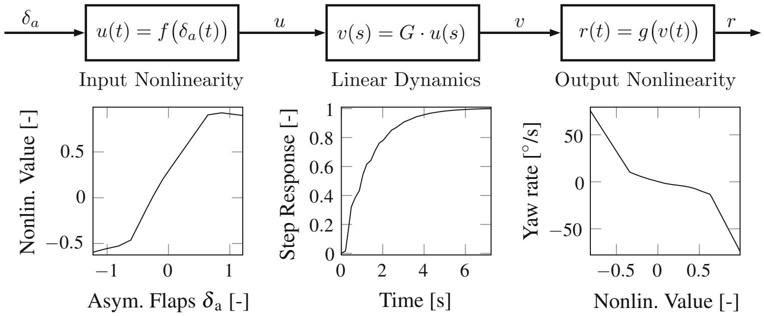

Fig. 6 Nonlinear Hammerstein–Wiener model

4.1 Heading Controller

Heading control is achieved by a PID-based controller. In contrast to other works [7] the process model is a black-box model obtained using system identification. The input to the system is the asymmetric brake deflection $\delta_a = \delta_{left} - \delta_{right}$. The output is the yawing rate r, which approximates the heading rate of change $\dot{\Psi}$ for small bank angles. To capture the nonlinear properties of both the actuators and the flight dynamics of the PPG we use a nonlinear Hammerstein–Wiener model. Hammerstein–Wiener models combine static input and output nonlinearities with a linear system in between.

Figure 6 shows the characteristics of the individual components of the identified model. Several conclusions can be drawn from these figures: The input nonlinearity shows a non-zero output in case of a zero input. Additionally the flap's effect does not change beyond approximately −0.2 in the negative or 0.3 in the positive direction. The output nonlinearity shows a plateau from −0.4 to 0.5 units, which has the effect of a dead-zone as well as of a dead-time. The step response of the linear block shows a rise time of about 2 s.

On the basis of these nonlinear dynamics we tuned a PID controller which achieves zero stationary control deviation and a linear response for reasonably small input changes. Figure 7 shows data from flight tests, where the commanded heading was controlled manually. Even though the heading controlled plant exhibits large overshoots, this behaviour is acceptable as long as a suitable model can be found.

4.2 Kinematic and Dynamic Model

Dynamics As implied by the considerable overshoots, a second-order model is needed to describe the plant behaviour to a sufficient accuracy. Based on flight-test data the following continuous-time transfer function model was identified:

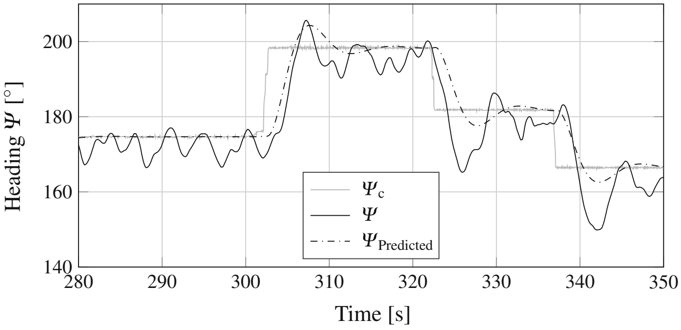

Fig. 7 Response of heading controller

$$
\begin{aligned}
G(s) &= \tfrac{K\cdot(1+T_z s)}{1+2\zeta T_w s+(T_w s)^2} \\
\text{where}\quad K = 1;\ T_z &= -0.29\ \text{sec};\ T_w = 1.47\ \text{sec};\ \zeta = 0.40
\end{aligned}
\tag{15}
$$

In addition to the plant behaviour, Fig. 7 shows the model response Ψ_{Model}. The model shows a similar behaviour with regard to the step response aswell as the overshoots to some degree. The large oscillations caused by the suspension cannot be represented since these are not caused by a measurable input to the plant. Since the MPC is formulated in a discrete-time setting, Eq. (15) was discretzied using a zero-order hold method and the MPC sample time T.

Kinematics The observer discussed in Sect. 3.2 estimates the airspeed, wind speed and wind direction. Due to this, the velocities in (4) are expressed in polar coordinates. Discretizing (4) with a sample time of T yields

$$
\begin{aligned}
\mathbf{v_g}(k) &= V_A(k)\begin{bmatrix}\cos\Psi(k)\\ \sin\Psi(k)\end{bmatrix} + V_W(k)\begin{bmatrix}\cos\Psi_W(k)\\ \sin\Psi_W(k)\end{bmatrix} \\
\mathbf{x_g}(k+1) &= \mathbf{x_g}(k) + \mathbf{v_g}(k)\cdot T
\end{aligned}
\tag{16}
$$

We use euler-integration because simulation results indicated no improvement when using more elaborate integration schemes.

4.3 Wind Estimation

Section 3.2 introduced the observer that is used to estimate the aerodynamic speed and the wind velocity. The two essential parameters of this observer are the number of sections N and the number of observations per section P. As a first approximation, N influences the convergence time and P the noise of the estimate. In simulation studies $N = 12$ and $P = 10$ were found to result in a good tradeoff between convergence speed and noise level.

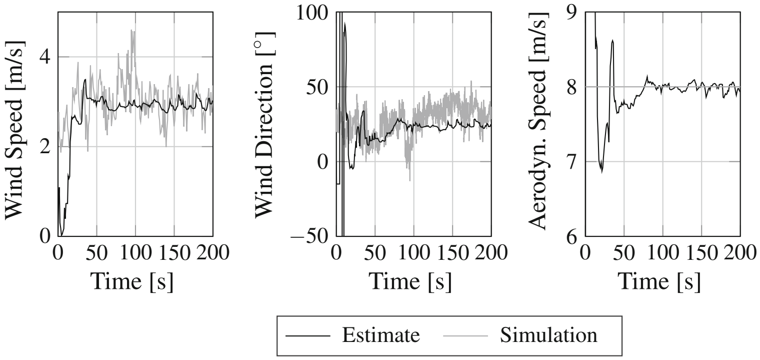

Fig. 8 Observer simulation results

Figure 8 shows one such simulation study. The simulated wind is based on measured wind data. During the first 50 s the heading slices are filled with new observations, leading to large changes in the estimate. The length of this initial phase depends on the actual flight path. After this initial phase, the solution stabilizes, effectively smoothing the simulated velocities.

4.4 Optimizer

The main advantage of the optimization algorithm presented in Sect. 3.4 is its conceptual simplicity. While this simplicity also translates to its implementation, the runtime of the MPC is still the major limiting factor for an application to faster control loops.

MPC timestep Equation (14) allows a calculation of the minimum MPC timestep T_{min}. It takes about $T_{\mathrm{once}} = 19\mu\mathrm{s}$ to step the model once and calculate the pointwise cost ℓ. Given the rise time of about 2 s of the plant (see Sect. 4.1), the prediction horizon was $\Delta t = 10$ s. The maximum heading change in one timestep was set to $\Delta\Psi_{c,\mathrm{max}} = 0.3\mathrm{rad}$ ($\approx 17°$), while the incremental change was set to $\Delta\Psi_c = 0.05$ rad ($\approx 3°$). This results in a minimum MPC timestep of $T_{\mathrm{min}} = 0.58$ s (i.e. a maximum control frequency of 1.72 Hz). Since T_{min} only constitutes the MPC computation time, we set the actual MPC timestep to 1 s to allow the microcontroller to execute other tasks besides computing the heading command.

Initial solution Starting optimization from a good initial solution considerably reduces the required iterations to find the optimal solution. If the model is sufficiently accurate we assume that the predicted flight path will be close to the actual future flight path. The next optimal solution will then be close to the current optimal solution, minus the first element. Using the last optimal solution to initialize the current solution reduces the required number of iterations considerably. Once a solution is found, it takes only about 50 iterations to find the optimal solution in the subsequent timesteps. This results in a computation time of about 5 ms, which is well below the theoretical worst-case of about 500 ms.

5 Evaluation

The following section presents an evaluation of the previously described control scheme. The individual components of the MPC will be discussed separately where applicable. First, results from a simulation study are discussed and then compared to results from flight tests.

Reference Flight Path All results are obtained using the same reference flight path. The flight tests were conducted at a small airfield with a limited area of about 250 m ×500 m. Thus the primary requirement for the reference flight path was to stay within these allowed limits. We chose a flight path in shape of the number 8, featuring both left and right turns. The minimum turn radius is 70 m. With an airspeed of 8 m/s and a maximum expected wind speed of 5 m/s this corresponds to a maximum turn rate of 0.16 rad/s (≈9°/s). This is well within the maximum turn rate of $\dot{\Psi}_{c,\max} = 0.3$ rad/s (≈17°/s) imposed by the heading controller.

5.1 Simulation Study

The following simulation results were obtained with a mean wind speed of $V_W = 3$ m/s in direction of the positive y-axis ($\Psi_W = 90°$). Wind gusts up to 1 m/s were added to the mean wind velocity. The estimated velocities behave similiarly to the estimates in Fig. 8 and thus are not discussed further. For the simulation, we used the nonlinear model described in Sect. 4.2.

Figure 9 shows the reference flight path and the simulated flight path. The simulated flight path stays within approximately ±10 m of the reference flight path. As is to be expected, the largest deviations occur when the velocity estimation is inaccurate. This inaccuracy is however part of the overall design, since the actuators of the PPG are too slow to counteract wind gusts. Additionally, the deviation from the reference flight path rises during fast turn maneuvers, where the PPG achieves comparatively large turn rates. This suggests an insufficient model accuracy for this flight state.

Effects of inaccurate velocity estimates The observer is based on the assumption that the observed variables are only varying slowly. Due to its working principle, changes in the wind direction or speed can only be detected reliably if enough new observations are gathered in new slices. Especially during long straight-line flight segments this condition will not be satisfied. To quantify the effect of erroneous estimtates of wind velocity and airspeed, we carried out simulation studies. In these

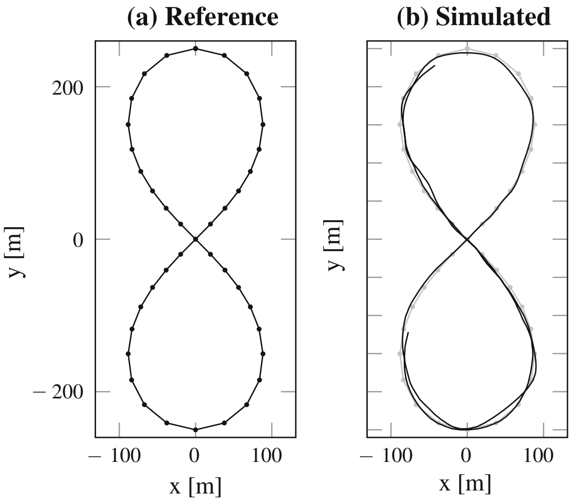

Fig. 9 Simulated flight along reference flight path

studies, the velocity estimates were set to a fixed value while an additional crosswind was introduced in the simulation. We found that the deviation from the reference path rises linearly with the error in wind speed estimation. An erroneous wind speed estimate of 5 m/s will lead to an error of about 30 m. The relevance of this result depends on the properties of the reference flight path and the required accuracy.

It should be noted that even in the case of an unaltered estimate, a mean deviation of about 2 m is present. This seems to result from the optimizers discretized search space. The optimizer is not able to find a commanded heading that leads to zero mean deviation.

5.2 Flight Test

A flight test was conducted to validate the results obtained in the simulation study described previously. Due to a biased heading measurement, the flight path guidance results are distorted and not included here. The effects of this bias are the same as those of inaccurate velocity estimates. Thus, resulting from an incorrect wind correction angle the actual flight path was offset from the reference flight path by about 20 m. Nevertheless, the resulting flight path was similar to the simulation results with respect to its variation and reproducability.

Only those components of the MPC not affected by the biased heading measurement will be discussed below.

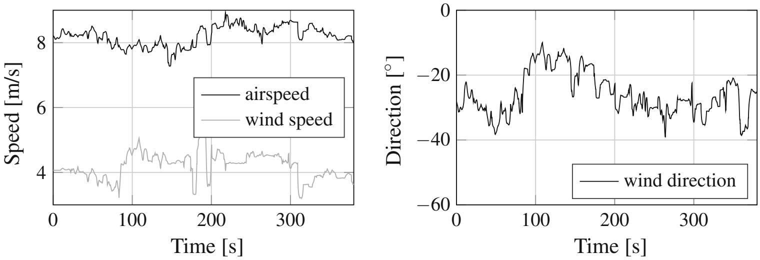

Fig. 10 Velocity estimation in flight test

5.2.1 Observer

Figure 10 shows the estimated wind speed and airspeed during one test flight. After take-off, the estimated airspeed converges to approximately 8 m/s, which is in the expected range. Due to the lack of airspeed sensors the estimated velocities could only be checked for plausibility and couldn't be validated further.

5.2.2 Model

Figure 11 compares the measured heading during a flight with the predicted heading. The left plot shows the heading commanded by the model predictive flight path controller. The heading controller has a considerable lag of approximately 2 s. The right plot shows the heading predicted by the internal dynamics model. Of course, not all dynamics are described by the simple linear model. A typical example of this can be seen around 610 s. Here, shortly after the turn starts, the payload swings back again before following the motion of the parafoil. Besides such nonlinear behaviour, the lag and the rise time of the plant are represented well by the linear model.

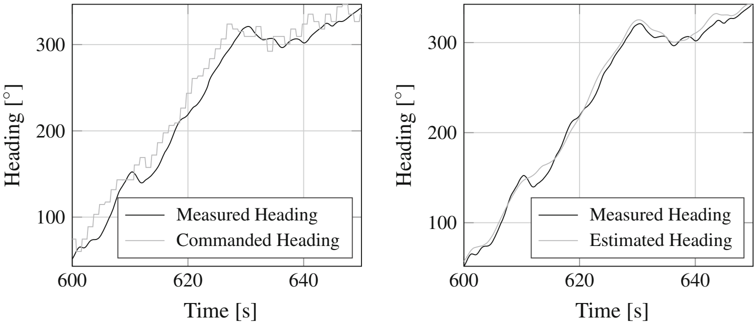

Fig. 11 Comparison between measured and predicted heading

It should be noted, that the model was identified during a flight where the heading was commanded manually. In contrast, the data underlying Fig. 11 was produced during an MPC-guided flight. Performing another system identification procedure on these flight data results in a model, very similar to the original model. Thus, despite the difference in excitation, the previously identified model could be used for the MPC without change.

6 Conclusion

We presented a nonlinear model predictive control scheme for the horizontal flight path guidance of a PPG. Model predictive controllers are inherently more complex than many classic control solutions. Because of this, the control scheme presented here was designed with a focus on simplicity, both conceptual and regarding its implementation. Using a heuristic optimization algorithm we were able to implement the control scheme on an embedded flight control platform. The control scheme adopts the common division of flight path control into heading control and flight path guidanace. While the heading controller was designed as a classic feedback controller, the flight path guidance was realized using an MPC. To predict the future flight path of the underlying model, an estimation of the mean wind velocity and airspeed was implemented. The estimation was transformed to a linear least-squares problem, yielding a robust estimation of the observed variables at a reduced computational cost.

The control scheme was developed and tested in a simulation environment. Here, a maximum deviation from the reference fligth path of 10 m was achieved. Flight tests were then conducted to validate this simulation result. While the actual flight path guidance could not be validated due to a biased heading measurement, the individual components of the MPC could be validated successfully. It was shown that the linear model of the heading dynamics describes the behaviour of the heading-controlled PPG well. Also, the estimation of the wind speed and airspeed yielded plausible results.

References

1. Astrom KJ, Wittenmark B (2008) Adaptive control, 2 edn. Dover Publications, New York
2. Cumer C, Toussaint C, Le Moing T, Poquillon E, Coquet Y (2012) Simulation of generic dynamics flight equations of a parafoil/payload system. In: 2012 20th mediterranean conference on control & automation (MED)
3. Grüne L, Pannek J (2011) Nonlinear model predictive control. Springer, London
4. Hur G, Valasek J (2003) System identification of powered parafoil-vehicle from flight test data. In: AIAA atmospheric fligth mechanics conference and exhibit
5. Jann T (2001) Aerodynamic model identification and GNC design for the parafoil-load system ALEX. In: 16th AIAA aerodynamic decelerator systems technology conference and seminar

6. Joos A, Fichter W (2011) Parallel implementation of constrained nonlinear model predictive controller for an FPGA-Based onboard flight computer. Springer, Berlin
7. Ochi Y, Kondo H, Watanabe M (2009) Linear dynamics and PID flight control of a powered paraglider. In: AIAA guidance, navigation and control conference
8. Rogers RM, et al. (2002) Aerodynamic parameter estimation for controlled parachutes. In: AIAA atmospheric flight mechanics conference and exhibit
9. Rossiter JA (2003) Model-based predictive control. Taylor & Francis Inc., London
10. Slegers N, Costello M (2005) Model predictive control of a parafoil and payload system. J Guid Control Dyn 28(4)
11. Slegers N, Kyle J, Costello M (2006) Nonlinear model predictive control technique for unmanned air vehicles. J Guid Control Dyn, vol 29
12. Slegers N, Brown A, Rogers J (2015) Experimental investigation of stochastic parafoil guidance using a graphics processing unit. Control Eng Pract, vol: 36
13. Toglia C, Vendittelli M (2010) Modeling and motion analysis of autonomous paragliders. Department of computer and system sciences Antonio Ruberti technical reports, 2(5)
14. Watanabe M, Ochi Y (2007) Modeling and motion analysis for a powered paraglider (ppg). In: SICE, 2007 annual conference, IEEE, pp 3007–3012