CEAS-GNC-2022-019

CEAS EuroGNC 2022
"Conference on Guidance, Navigation and Control"
3-5 May 2022 @ Technische Universität Berlin, Germany

Nonlinear Model Predictive Path Planning with Obstacle Avoidance for an Automated Landing Approach of Fixed-Wing Aircraft

Andreas Steinleitner Research associate, Institute of Flight Mechanics and Controls, University of Stuttgart, 70569, Stuttgart, Germany. andreas.steinleitner@ifr.uni-stuttgart.de

Yassine Bensaad Doctoral Researcher, Mercedes-Benz AG, 71063, Sindelfingen, Baden-Württemberg, Germany. yassine.bensaad@daimler.com

Walter Fichter Professor, Institute of Flight Mechanics and Controls, University of Stuttgart, 70569, Stuttgart, Germany. walter.fichter@ifr.uni-stuttgart.de

ABSTRACT

This work addresses the implementation of a final approach guidance for fixed-wing aircraft using a nonlinear model predictive control concept with collision avoidance capability. This comprises an obstacle identification method using Euclidean clustering and least squares approximation. After the provision of initial trajectories with Dubins paths which consider the flight areas that need to be avoided, the optimal control problem for the landing path is solved using a direct method based on a cost function with suitable penalty criteria. Optimization constraints are determined from flight mechanical limitations, performance requirements and path constraints. Drawbacks of a model predictive guidance are discussed and accounted for by a waypoint tracking strategy in combination with feedforward terms derived from the desired trajectory and the aircraft model. The robustness of the path planning and guidance is investigated with Monte Carlo simulations.

Keywords: CEAS EuroGNC; Nonlinear Model Predictive Control; Automated Landing; Aircraft Control

Nomenclature

α	=	Angle of attack $[rad]$
$a_e, b_e, c_e, d_e, e_e, f_e, g_e, h_e, i_e, j_e$	=	Ellipsoid parameters
a_x, a_y, a_z	=	Accelerations in Earth frame $[\frac{m}{s^2}]$
AGL	=	Altitude above ground level $[m]$
$\begin{pmatrix} C_D & C_Y & C_L \end{pmatrix}^\top$	=	Aerodynamic force coefficients
g	=	Gravitational acceleration $[\frac{m}{s^2}]$
m	=	Aircraft mass $[kg]$
r	=	Radius $[m]$
MPC	=	Model predictive control
NMPC	=	Nonlinear model predictive control
Φ	=	Roll angle $[rad]$
p_1, p_2, p_3, p_4, p_5	=	Weights for objective functional

T, δ_F	=	Thrust $[\frac{m}{s}]$, Thrust lever
V_a, V_k	=	Absolute values for air and track speed respectively $[m/s]$
γ	=	Flight path angle $[rad]$
χ	=	Flight path azimuth $[rad]$
$\mathbf{x} = \begin{pmatrix} x & y & z \end{pmatrix}^\top$	=	Position wrt. origin of flat and inertial Earth frame $[m]$
$\omega = \begin{pmatrix} \omega_v & \omega_\gamma & \omega_\chi \end{pmatrix}^\top$	=	Pseudo control variables
$\mathbf{X}_a = \begin{pmatrix} X_a & Y_a & Z_a \end{pmatrix}^\top$	=	Forces along aerod. axes: neg. drag, side force, neg. lift $[N]$
$(\cdot)_f$	=	Desired final value for state or actuator

1 Introduction

Aircraft landings, especially during adverse weather conditions, pose the largest risk for aviation accidents [1]. The majority of flight incidents occur during this period and are due to human operating errors [2, 3]. The automation of this flight phase is an opportunity to reduce the possibilities of mishaps due to pilot mistakes and increase passenger safety. The automation of the landing approach procedure is already well established for large commercial aircraft (CS-25). These rely on various navigation aids that are available at large airports, where these aircraft land. Smaller and lighter aircraft (CS-23) are usually not equipped with such systems and often operate at smaller airports with less ground-based facilities. The necessary approach patterns can also be more complex and require a flexible path planning functionality during landing.

The research on path planning for fixed-wing aircraft has drawn a lot of attention in recent years. One technique, that emerged over time and is now widely adopted, is the generation of an initial trajectory based on geometrical considerations. This serves as a preliminary guess and starting point for an optimization that improves the solution path with regard to certain evaluation criteria. In [4], the derivation of an initial landing trajectory as a starting point for the numerical computation is discussed. The output of this initial trajectory generation are often sequences of waypoints, which need additional smoothing as presented in [5] and [6].

There is a wide amount of works that use the Model Predictive Control (MPC) technique to perform a trajectory optimization and find an optimal path even in the presence of spatial or aerodynamic constraints. This method was applied for missile guidance, e.g. in [7], but is also investigated with regard to fixed-wing aircraft path planning. The deployment of nonlinear MPC for high level path tracking is tested in flight in [8]. In [9], motion planning for agile aerobatics maneuvers is presented including the deliberate traverse into angle of attack regions that cause stalls. Long range trajectories are considered in [10], where the deployment or avoidance of weather phenomenons is important for trajectory generation. It is observed in [11], that path planning parameters for the model predictive control and geometrical features of the corresponding path are correlated. The authors provide methodical guidelines for the choice of the MPC design parameters. Descent trajectories, which represent the relevant paths in this work, are developed in [12], where performance improvements for trajectory optimization based on neighboring extremals are achieved by enabling the restructuring of a nonlinear into a quadratic programming problem. In the absence of enforced approach patterns, this concept yields an optimal trajectory with regard to approach duration and mechanical limitations of the aircraft. This provides the option to trigger a landing procedure at a random aerial position and be guided to the desired runway automatically. In a similar attempt to reduce the computational effort for the optimization, the dynamic model is simplified in [13] with a focus on fewer optimization parameters.

However, these works do not include the consideration of virtual obstacles like No-Fly zones or real obstacles that need to be regarded during a real-world aircraft landing approach. Fig. 1 depicts the runway and approach patterns for a small airfield in the south-west of Germany. It is apparent, that an

approach to runway 36 from the western direction entails a very short landing segment after the last turn and therefore prevents a long, straight descent to the runway. The illustration on the right side of the figure shows, how this approach pattern can be enforced by virtual No-Fly obstacles and the desired pattern is achieved during an automatic landing approach with a flexible guidance concept that will be presented in this work.

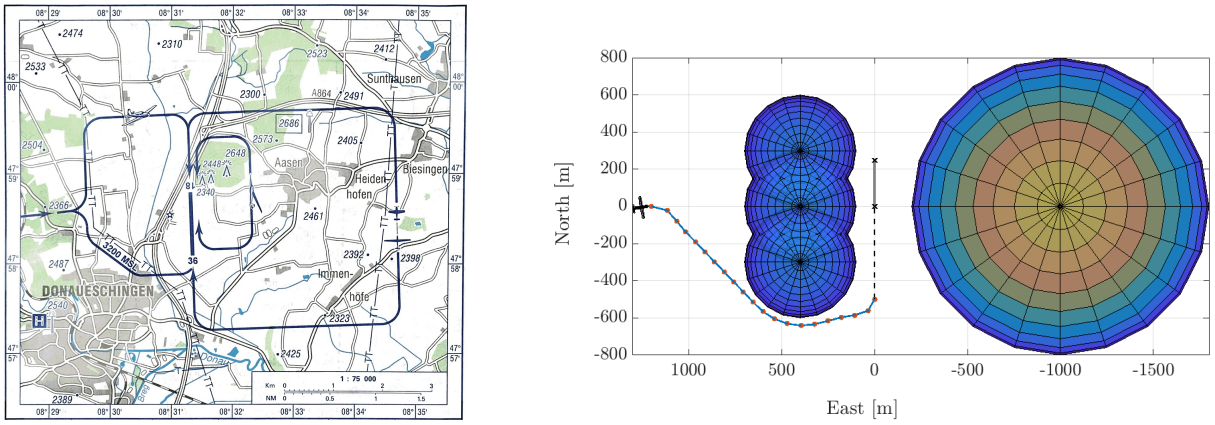


Fig. 1 Airfield Donaueschingen-Villingen EDTD [14]

The obstacle avoidance capability for the landing approach guidance is not only helpful to comply with airspace regulations, but can also improve the autonomous capabilities of an aircraft during emergency landings or at rural airports in rough terrain. In these cases it can be ensured, that objects in ground vicinity posing a collision risk on the aircraft route are avoided. A method for obstacle detection and segmentation was presented in [15]. However, this approach is limited to aerial obstacles. In [16], a concept to fuse electro-optic, infrared, and ADS-B sensors is presented that yields the ability to detect obstacles and track them during a flight. The detection of aviation obstacle based on low altitude data from a dense 3D point cloud is presented in [17].

With the aim to develop a real-world applicable method to generate an optimal path towards a runway without penetrating any prohibited areas regardless of their cause and virtual or physical nature, this work intends to incorporate the results of recent research in both the field of obstacle classification and model predictive trajectory generation and develop a system that can safely guide unmanned fixed-wing aerial vehicles to the desired landing area.

After introducing the aircraft model that is used for the control design, it is explained, how possible collision objects are identified and clustered. This information is fed into a path planning algorithm, which is subsequently explained and provides an optimal trajectory by means of Dubins paths and nonlinear model predictive control. In the last section, a simulation campaign is presented that validates the discussed functionality.

2 Aircraft Dynamics

This section presents the aircraft model and derivation of the equations that represent the basis for the trajectory optimization and the derivation of suitable path constraints. The aircraft model that is used in the remainder of this work consists of kinematic and dynamic models for the translational motion only. The translational kinematics determine the motion of a point mass in three-dimensional space and describe its dependency on the groundspeed V_k, the flight path angle γ, and the flight path azimuth χ:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = V_k \begin{pmatrix} \cos(\chi)\cos(\gamma) \\ \sin(\chi)\cos(\gamma) \\ -\sin(\gamma) \end{pmatrix} \tag{1}$$

The dynamic translational motion assuming a flat, non-rotating Earth is derived from a mass point model and described by the following system of equations [18]. It is modeled, how track speed V_k, flight path angle γ and flight path azimuth χ evolve under the influence of the angle of attack α, the thrust command δ_F and the roll angle Φ. The model deviates from the original derivation due to the assumption $\Phi \approx \mu$, because the roll angle can be determined more easily in practice.

$$\begin{pmatrix} \dot{V}_k \\ \dot{\gamma} \\ \dot{\chi} \end{pmatrix} = \frac{1}{mV_k} \begin{pmatrix} V_k\left(\cos(\alpha)T(\delta_F) + X_a(\alpha, V_a) - mg\sin(\gamma)\right) \\ -\cos(\Phi)\left(Z_a(\alpha, V_a) - \sin(\alpha)T(\delta_F)\right) - mg\cos(\gamma) \\ \frac{\sin(\Phi)}{\cos(\gamma)}\left(-Z_a(\alpha, V_a) + \sin(\alpha)T(\delta_F)\right) \end{pmatrix}, \tag{2}$$

where $T(\delta_F)$ represents the engine thrust. The rotational motion is neglected, since an attitude controller is expected to provide tracking capabilities for a desired attitude.

3 Obstacle Classification and Spline Planning

This work does not consider the sensor-based detection of obstacles. It is assumed, that a system based on camera, distance measurement, or infrared sensor data is able to provide information about surrounding obstacles via point cloud data, as described for example in [19]. Since the number of obstacles is not limited and there is no requirement for an already existing segmentation of individual elements, it is necessary to apply a Euclidean clustering mechanism in order to separate individual objects. This concept groups together all measurement points that are bounded in a neighborhood with a predetermined maximum distance from each other. The tree search for the nearest next point is performed as described in [20].

The clustering yields separate groups in the point cloud that are each approximated by ellipsoid shapes. This reduces the computational effort to consider the information in terms of obstacle avoidance and evasive maneuver planning significantly. The general mathematical description of a three-dimensional ellipsoid in its quadratic form denotes

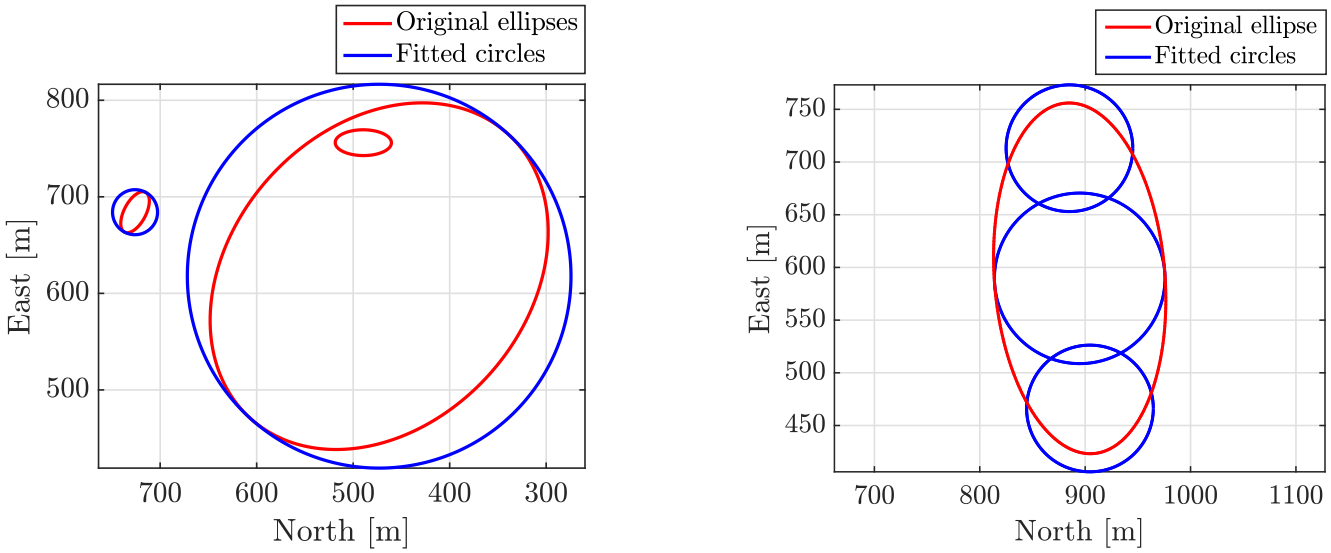
$$a_e x^2 + b_e y^2 + c_e z^2 + d_e xy + e_e xz + f_e yz + g_e x + h_e y + i_e z + j_e = 0. \tag{3}$$

According to [21], the parameters $a_e - j_e$ are determined using a least squares method that minimizes the residual sum of the squared offsets of each point from the fitted ellipsoid. The obstacle avoidance concept is primarily based on the minimum distance to the nearest obstacle. This calculation is computationally expensive using ellipsoid shapes. To increase the computational performance, it is therefore beneficial to transform the obtained objects by disassembling the ellipsoids into spheres, that cover approximately the same area, as shown in Fig. 2 exemplary in two dimensions.

The detected obstacles that are now available in the form of parameterized spheres are used to generate initial trajectories for the path planning and guidance. This is shown in the following section.

4 Nonlinear Model Predictive Guidance

The approach phase during landing is characterized by the requirement on the aircraft to maneuver agile and to be able to follow a fairly dynamic approach pattern with potential turns shortly before the runway. This requirement is satisfied with a guidance algorithm based on Nonlinear Model Predictive Control (NMPC) providing an optimal trajectory until the flare phase close to the runway.



(a) Ellipse with low eccentricity **(b) Ellipse with high eccentricity**

Fig. 2 Transformation of elliptic into circular obstacles

4.1 Initial Trajectory using Dubins paths

When an automated landing procedure is triggered, an initial trajectory to the beginning of a predefined runway considering all the known and classified obstacles is planned based on two Dubins paths that are responsible for the vertical and horizontal plane respectively. This technique promises low computation cost due to the few parameters necessary to define such a trajectory. A detailed presentation of this method can be seen in [22].

There are several path constraints that confine the possible Dubins paths: These include the minimum turn radius $r_{\min} = \frac{V_{k,f}^2 \cos(\gamma(\Phi_{\max}))}{g \tan(\Phi_{\max})}$ as a dependency of the maximum admissible roll angle $\Phi_{\max}$ and the desired ground speed at the end $V_{k,f}$ as well as a maximum possible load factor $-\frac{a_z}{g} \leq n_{z,\max}$.

After the generation of an optimal Dubins path and its discretization into N equidistant waypoints along the trajectory, the obstacle avoidance algorithm modifies this path to consider the additional restrictions to a suitable path: The function $\text{dist}_{i,j} : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ defines the distance of the vehicle position $\mathbf{x}_i = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^\top$ on the Dubins path at the discrete time t_i under consideration of the size of the UAV with r_{UAV} to an obstacle parameterized by $\mathbf{x}_j = \begin{pmatrix} x_{0,j} & y_{0,j} & z_{0,j} \end{pmatrix}^\top$ and radius $r_{0,j}$ with $j \in \mathbb{J}$ denoting one member of the set of identified obstacles:

$$\text{dist} : \big((\mathbf{x}_i, r_{\text{UAV}}), (\mathbf{x}_{0,j}, r_{0,j})\big) \to \sqrt{(x_i - x_{0,j})^2 + (y_i - y_{0,j})^2 + (z_i - z_{0,j})^2} - (r_{0,j} + r_{\text{UAV}}) \tag{4}$$

An obstacle on collision course is identified with the minimization problem

$$j^\star = \underset{i \in \{1,\ldots,N\}, j \in \mathbb{J}}{\arg\min} \ \text{dist} \left(\begin{pmatrix} x_i \\ y_i \\ z_i \\ r_{\text{UAV}} \end{pmatrix}, \begin{pmatrix} x_{0,j} \\ y_{0,j} \\ z_{0,j} \\ r_{0,j} \end{pmatrix} \right), \tag{5}$$

where the distance function yields a negative result in case of an imminent collision:

$$\text{dist}\left(\begin{pmatrix} x_i \\ y_i \\ z_i \\ r_{\text{UAV}} \end{pmatrix}, \begin{pmatrix} x_{0,j^\star} \\ y_{0,j^\star} \\ z_{0,j^\star} \\ r_{0,j^\star} \end{pmatrix}\right) < 0 \tag{6}$$

All discretized waypoints on the trajectory that meet the criterion in Eq. (6) are relocated in the following way: While the waypoint heights are maintained, the alternative route should follow a circle in the horizontal plane with radius $r_a = r_{0,j^\star} + r_{\text{UAV}}$ and center $\begin{pmatrix} x_i & y_i \end{pmatrix}^\top$. At the same time, the waypoints should be moved only in the direction perpendicular to the tangent of the trajectory at the point $\begin{pmatrix} x_i & y_i \end{pmatrix}^\top$. These requirements compose the following system of equations with the unknown alternative waypoint $\mathbf{x}_a = \begin{pmatrix} x_a & y_a \end{pmatrix}^\top$:

$$\left\langle \begin{pmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \end{pmatrix}, \begin{pmatrix} x_a - x_i \\ y_a - y_i \end{pmatrix} \right\rangle = 0 \tag{7}$$

$$(x_a - x_{0,j^\star})^2 + (y_a - y_{0,j^\star})^2 + (z_a - z_{0,j^\star})^2 - r_a^2 = 0 \tag{8}$$

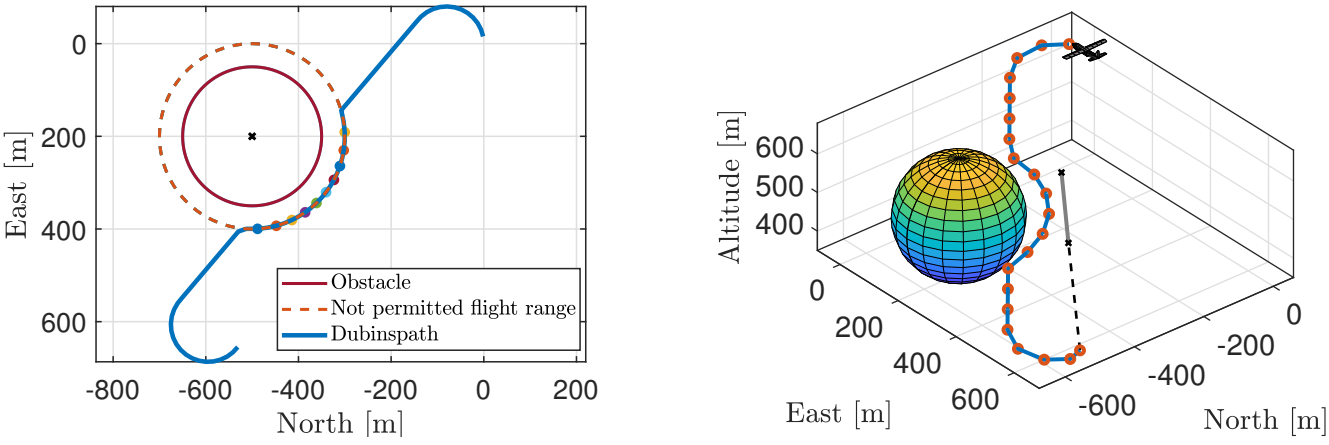
The two solutions of this system of equations represent the alternative routes on the left and right hand side of the collision obstacle, respectively. Mathematically, the shorter option for each waypoint $\mathbf{x}_i$ can be identified using

$$q_i = \text{sign}\left((x_i - x_{i-1})(y_{0,j^\star} - y_i) - (y_i - y_{i-1})(x_{0,j^\star} - x_i)\right). \tag{9}$$

To guarantee that all waypoints that are modified to avoid the collision with obstacle $j^\star$ follow the same detour direction, Eq. (9) is extended to a recursive form:

$$q_i = \text{sign}\left(\text{sign}\left((x_i - x_{i-1})(y_{0,j^\star} - y_i) - (y_i - y_{i-1})(x_{0,j^\star} - x_i)\right) + 2q_{i-1}\right) \quad \text{with} \quad q_1 = 0 \tag{10}$$

In Fig. 3, an initial trajectory with obstacle avoidance is shown. The resulting discretized trajectory is converted into segments of cubic splines using the concept in [23].



(a) Obstacle avoidance in 2D **(b) Obstacle avoidance in 3D**

Fig. 3 Cubic spline with circular obstacle avoidance

4.2 NMPC Optimization

Using the generated cubic spline trajectory as an initial estimate, the calculation of the optimal trajectory considering path constraints and boundary conditions represents an optimum control problem subject to the dynamic model. Aligned to the approach in [13], we introduce the following pseudo variables:

$$\begin{pmatrix} \dot{V}_k \\ \dot{\gamma} \\ \dot{\chi} \end{pmatrix} = \begin{pmatrix} \omega_v \\ \omega_\gamma \\ \omega_\chi \end{pmatrix} \tag{11}$$

Combined with Eq. (1), this system of equations represents the dynamic model $X(t)$, which serves together with the input signal to be optimized $\bar{U}(\cdot;t)$ at time step t as the foundation for the optimal control problem that is solved by minimizing the objective functional

$$\begin{aligned} J(X(t),\bar{U}(\cdot;t)) \;=\; & \int_t^{t+T} \Bigg(\underbrace{p_0\sqrt{\dot{x}(\tau)^2+\dot{y}(\tau)^2+\dot{z}(\tau)^2}}_{L1} + \frac{1}{T}\Big(\underbrace{p_1(\tau^2+(T-\tau)^2)\omega_\gamma(T)}_{L2} \\ & + \underbrace{p_2\omega_v(\tau)^2+p_3\omega_\gamma(\tau)^2+p_4\omega_\chi(\tau)^2}_{L3} + \underbrace{p_5\left(V_k(\tau)-V_f\right)^2}_{L4} \Big) \Bigg) \mathrm{d}\tau \end{aligned} \tag{12}$$

This functional contains four Lagrange-type criteria that influence the optimality of the calculated path. These are the trajectory length (L1) and the amount of flight path angle oscillation at the beginning and end of the flight (L2). The latter part has a positive effect on the overall stability of the total system. Furthermore, the necessary actuating effort (L3) and the deviation from the desired final speed (L4) are considered. The influence of the individual parts varies and is specified by the weight factors $p_{\{0,\dots,5\}}$. For the simulations shown in the latter sections of this work, the following values are used:

$$p_0 = p_2 = p_3 = 100; \qquad p_1 = 1000; \qquad p_4 = p_5 = 10$$

The boundary conditions of the optimum control problem regard the desired aircraft states at the start ($t = t_0$) and end ($t = t_f$) point of the trajectory. The aircraft state at the end is characterized by a desired flight path angle γ_f and azimuth χ_f at this point and an end position that can be derived from the start point of the runway assuming a linear path segment with length l_{fa} during the final approach:

$$x_f = x_l + l_{\mathrm{fa}}\cos(\chi_f) \qquad y_f = y_l + l_{\mathrm{fa}}\sin(\chi_f) \qquad z_f = z_l + l_{\mathrm{fa}}\sin(\gamma_f) \tag{13}$$

The path constraints with respect to obstacle avoidance are denoted

$$0 \leq \mathrm{dist}\left(\begin{pmatrix} x_i \\ y_i \\ z_i \\ r_{\mathrm{UAV}} \end{pmatrix}, \begin{pmatrix} x_{0,j^\star} \\ y_{0,j^\star} \\ z_{0,j^\star} \\ r_{0,j^\star} \end{pmatrix} \right). \tag{14}$$

It is also useful to enforce a smooth actuator command without discontinuities to decrease actuator oscillations. This is ensured with the following constraints:

$$\delta\omega_{\mathrm{v,min}} \leq \quad \omega_{\mathrm{v,i}} - \omega_{\mathrm{v,i\text{-}1}} \quad \leq \delta\omega_{\mathrm{v,max}} \tag{15}$$

$$\delta\omega_{\gamma,\mathrm{min}} \leq \quad \omega_{\gamma,i} - \omega_{\gamma,i-1} \quad \leq \delta\omega_{\gamma,\mathrm{max}} \tag{16}$$

$$\delta\omega_{\chi,\mathrm{min}} \leq \quad \omega_{\chi,i} - \omega_{\chi,i-1} \quad \leq \delta\omega_{\chi,\mathrm{max}} \tag{17}$$

Constraints for the pseudo control variables can be derived from Eq. (2): From the lift-to-weight equilibrium, one can approximate the minimum speed. The pseudo controls ω_γ and ω_χ are limited due to the mechanically constrained deflection of the actuators, the limit of the producible lift and the permitted maximum roll angle $\Phi_{\max}$. The auxiliary conditions on the optimization problem are listed in Table 1.

Parameter	Lower limit	Upper limit
z	z_f	-
γ	$\gamma_{\min}$	$\gamma_{\max}$
χ	$\chi_{\min}$	$\chi_{\max}$
V	$\sqrt{\frac{2mg}{\rho S C_L}}$	$V_{\max}$
ω_v	$\frac{T_{\min}-\frac{\rho}{2}V_{k,\max}^2 S C_D}{m} - g\sin(\gamma_{\max})$	$\frac{T_{\max}-\frac{\rho}{2}V_{k,\min}^2 S C_D}{m} - g\sin(\gamma_{\min})$
ω_γ	$\frac{\alpha_{\max}\frac{\rho}{2}V_{k,\max}^2 S C_L + T_{\max} - mg\cos(\gamma_{\min})}{m V_{k,\max}}$	$\frac{\alpha_{\max}\frac{\rho}{2}V_{k,\max}^2 S C_L + T_{\min} - mg\cos(\gamma_{\min})}{m V_{k,\max}}$
ω_χ	$\frac{\tan(\Phi_{\min})(V_{k,\min}\omega_{\gamma,\min}+g\cos(\gamma_{\min}))}{V_{k,\min}\cos(\gamma_{\min})}$	$\frac{\tan(\Phi_{\max})(V_{k,\min}\omega_{\gamma,\min}+g\cos(\gamma_{\min}))}{V_{k,\min}\cos(\gamma_{\min})}$

Table 1 Auxiliary conditions

The optimization problem described with the nonlinear dynamics in Eqs. (1, 11) and the objective functional and conditions described in this section are solved using the IPOPT solver [24] using a classic Runge–Kutta method (4^{th} order) for integration. Fig. 4 depicts two different landing approaches to the runway drawn as a black line. The dashed straight line represents the straight approach segment before the touchdown with length l_{fa} and the path angles γ_f and χ_f and the red dots illustrate the discretization of the trajectory into N waypoints $\mathbf{x}_i$.

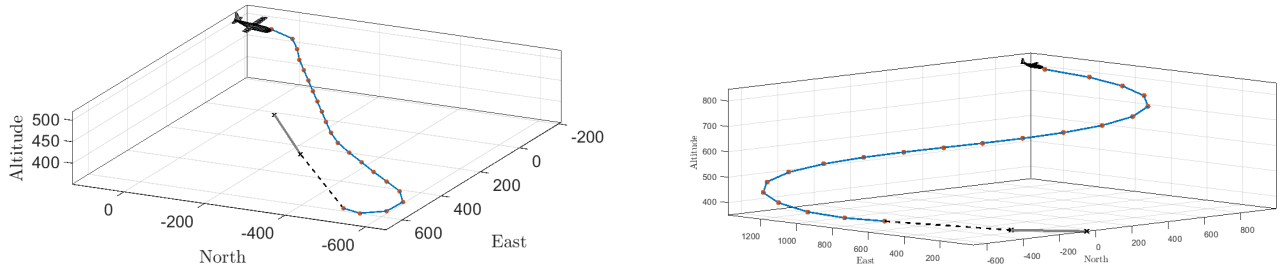


Fig. 4 Trajectories for landing approach

4.3 Feed Forward

The result of the NMPC algorithm are values for all states and pseudo control variables at each discrete time step over a finite time horizon. Since the desired output is a trajectory and hence the progression of merely the position states over this time period, the result for other optimization variables could be disregarded. They are not used as direct control commands for the inner control loops, because the sampling rate of attitude and rate control is higher than the necessary requirements for the bandwidth of the model predictive guidance which is limited by computational resources. It is therefore suitable to

use the next discretized waypoint along the optimized trajectory as the reference position for a waypoint tracking algorithm that provides the inner loops with command values for attitude and thrust until an update by the NMPC algorithm is available.

The consideration of the translational dynamics in Eq. (2) is primarily important for the derivation of the constraints of the pseudo variables and ensures that the resulting trajectory does not require the aircraft to exceed its flight envelope boundaries. However, from Eqs. (2) and (11), one can derive control commands for aerodynamic angles and the thrust command:

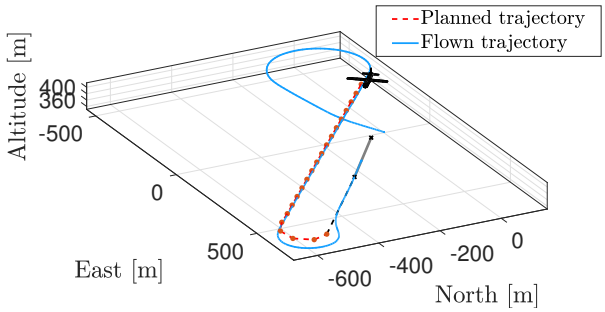
$$T_{\text{ffw}} = m(\omega_v + g\sin(\gamma)) + \frac{\rho}{2}V_a^2 S C_D \tag{18}$$

$$\Phi_{\text{ffw}} = \arctan\left(\frac{V_k \cos(\gamma)\omega_\chi}{V_k\omega_\gamma + g\cos(\gamma)}\right) \tag{19}$$

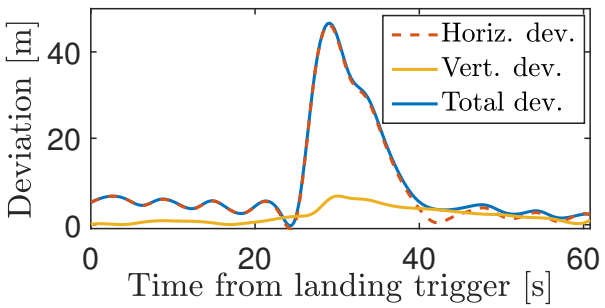
$$\alpha_{\text{ffw}} = m\frac{V_k\omega_\gamma + g\cos(\gamma)}{\cos(\Phi)\frac{\rho}{2}V_a^2 S C_L + F_{\text{UAV}}} \tag{20}$$

The inverted translational dynamics are therefore considered to generate feedforward control terms that improve the tracking performance. The feedforward term α_{ffw} shows a performance improvement especially during descents and is therefore used only with flight path angles $\gamma < 0$.

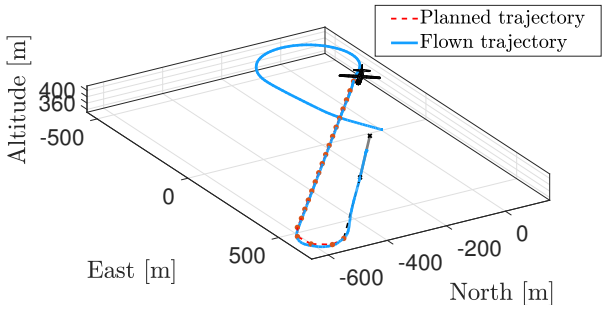
Because the waypoint tracking controller calculates straight paths between the NMPC waypoints and hence splits the previously smooth trajectory into a sequence of linear segments, the use of the mentioned feedforward terms is only beneficial, if the length of these segments does not exceed a critical threshold after which the feedforward counteracts the tracking control law output. If this requirement is respected, an improvement in tracking performance as illustrated in Fig. 5 can be observed qualitatively and quantitatively.



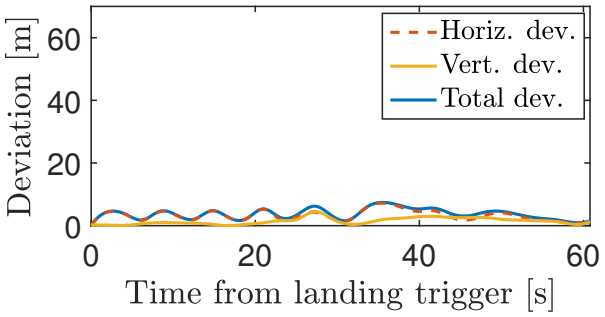
(a) Flight mission guided without feedforward



(b) Position error during landing



(c) Flight mission guided with feedforward



(d) Position error during landing

Fig. 5 Effect of roll angle feed forward

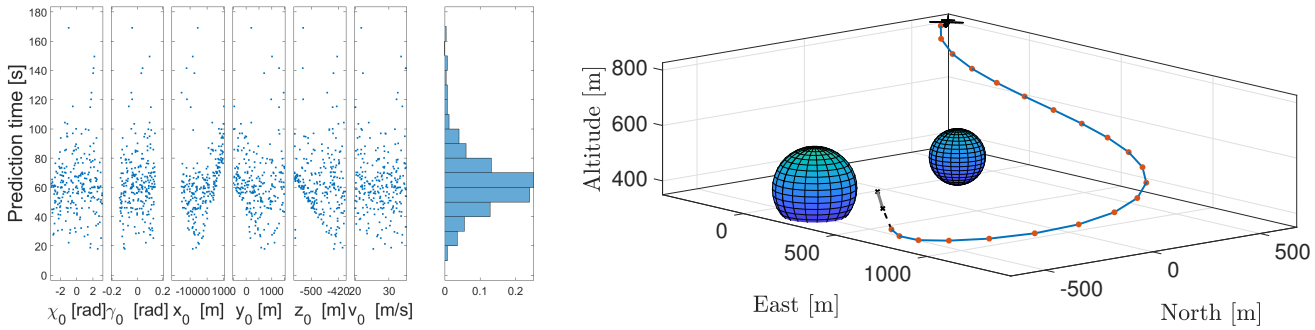
5 Monte Carlo Simulations

The suggested MPC guidance algorithm for landing is tested in an extensive Monte-Carlo simulation campaign that is presented in this section. The aircraft state is varied for the stochastic analysis in the range that is outlined in Table 2.

Parameter	Unit	Lower limit	Upper limit
x	m	-1500	1000
y	m	-500	1500
AGL	m	50	250
V_a	$\frac{\text{m}}{\text{s}}$	20	35
γ	deg	-8	8
χ	deg	-180	180

Table 2 Monte Carlo parameter range

All parameters are drawn from a uniform distribution, assuming that there is no preference state in which an automatic landing procedure is triggered. The probability density for each parameter is therefore constant across its value range. Fig. 6a shows the result of 250 simulations with randomized starting conditions.



(a) Landing duration distribution **(b) Descent from high altitude**

Fig. 6 Monte Carlo simulation results

The optimization algorithm converges against an optimal trajectory for all performed simulations without violating any auxiliary conditions or path constraints. Depending on the initial state, the duration of the landing maneuvers ranged between $\Delta t = 20$s and $\Delta t = 170$s.

Special attention should be paid to landings triggered at a significant height above ground. As is evident from Fig. 6b, due to the limited descent flight path angle it is often not possible to generate a trajectory that satisfies the objective of the shortest path, but instead considers other parts of the objective functional J to prolong the descent artificially with an additional turn in order to allow enough time to lose height.

6 Conclusion

This work shows the successful implementation of a landing guidance concept based on nonlinear model predictive control. The benefits of the presented concept in terms of flexibility of executable approach patterns and obstacle avoidance capabilities are demonstrated. A future work may extend the contents of this paper by alleviating the restriction on ground-based, non-moving obstacles and improving the functionality towards avoiding other moving vehicles.

Acknowledgments

The research was supported by the project SiFlA (grant number 100433574) of the German Federal Aeronautical Research Program VI (LuFo VI-1), founded and supported by the Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

References

[1] Airbus. A Statistical Analysis of Commercial Aviation Accidents 1958 - 2021. Brochure, February 2022. https://accidentstats.airbus.com/sites/default/files/2022-02/Statistical-Analysis-of-Commercial-Aviation-Accidents-1958-2021.pdf.

[2] Roalt Aalmoes, Ronald Erkamp, Yuk Shan Cheung, and R van Nieuwpoort. Performance improvements for calculations of third party risk around airports. 2013.

[3] Brittnee Nicholle Branham. Analysis of fatal general aviation accidents occurring from loss of control on approach and landing. 2013.

[4] Panagiotis Tsiotras, Efstathios Bakolas, and Yiming Zhao. Initial guess generation for aircraft landing trajectory optimization. In *AIAA Guidance, Navigation, and Control Conference*, page 6689, 2011.

[5] Carlo L Bottasso, Domenico Leonello, and Barbara Savini. Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology*, 16(6):1152–1168, 2008.

[6] Federico Pinchetti, Alexander Joos, and Walter Fichter. Efficient continuous curvature path generation with pseudo-parametrized algebraic splines. *CEAS Aeronautical Journal*, 9(4):557–570, 2018.

[7] Jongho Park, Youngil Kim, and Jong-Han Kim. Integrated guidance and control using model predictive control with flight path angle prediction against pull-up maneuvering target. *Sensors*, 20(11):3143, 2020.

[8] Thomas Stastny and Roland Siegwart. Nonlinear model predictive guidance for fixed-wing UAVs using identified control augmented dynamics. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 432–442. IEEE, 2018.

[9] Max Basescu and Joseph Moore. Direct nmpc for post-stall motion planning with fixed-wing UAVs. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9592–9598. IEEE, 2020.

[10] Federico Mothes. Trajectory planning in time-varying adverse weather for fixed-wing aircraft using robust model predictive control. *Aerospace*, 6(6):68, 2019.

[11] Alexander Joos and Walter Fichter. Nonlinear model predictive control parameters and path geometry. *Journal of Guidance, Control, and Dynamics*, 42(1):175–180, 2019.

[12] Ramon Dalmau, Xavier Prats, and Brian Baxley. Sensitivity-based non-linear model predictive control for aircraft descent operations subject to time constraints. *Aerospace*, 8(12):377, 2021.

[13] Min Prasad Adhikari and Anton HJ de Ruiter. Online feasible trajectory generation for collision avoidance in fixed-wing unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 43(6):1201–1209, 2020.

[14] Luftfahrthandbuch - Aeronautical Information Publication. *DFS Deutsche Flugsicherung GmbH*, 2014.

[15] Tim G McGee, Raja Sengupta, and Karl Hedrick. Obstacle detection for small autonomous aircraft using sky segmentation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4679–4684. IEEE, 2005.

[16] Kevin Theuma, Roger Archer, Kenneth Chircop, David Zammit-Mangion, and Jason Gauci. Multi-sensor obstacle detection and tracking for rpas situation awareness and guidance. In *Euro GNC-4th CEAS Specialist Conference on Guidance, Navigation & Control*, 2017.

[17] Marta Lalak and Damian Wierzbicki. Methodology of detection and classification of selected aviation obstacles based on uav dense image matching. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022.

[18] W Fichter and W Grimm. Flugmechanik, shaker. *Dueren, Germany*, pages 29–36, 2009.

[19] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[20] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[21] Yury. Ellipsoid fit. `https://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit`. Online; accessed 06 Dec 2021.

[22] Johannes Stephan, Stefan Notter, Ole Pfeifle, Federico Pinchetti, and Walter Fichter. Spline trajectory planning and guidance for fixed-wing drones. In *AIAA Scitech 2020 Forum*, page 0372, 2020. DOI: 10.2514/6.2020-0372.

[23] Michael S Floater. On the deviation of a parametric cubic spline interpolant from its data polygon. *Computer Aided Geometric Design*, 25(3):148–156, 2008.

[24] A Wächter, CD Laird, and Y Kawajir. Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT. Technical report, Tech. Rep., Carnegie Mellon University, 2006.