CEAS Aeronautical Journal (2024) 15:385–396
https://doi.org/10.1007/s13272-024-00721-1

ORIGINAL PAPER

Helicopter short-term collision avoidance with sampling-based model predictive control

Alexej Dikarew[1]

Received: 25 January 2023 / Revised: 29 January 2024 / Accepted: 30 January 2024 / Published online: 11 March 2024

Abstract
Automatic helicopter flight in uncertain surroundings remains a challenging task due to sudden changes in environment, requiring fast response to guarantee safe and collision-free guidance. Increasing numbers of small unmanned aerial vehicles, which are not covered by air traffic control, pose a potential threat to rotorcraft operating in lower airspace. To provide collision avoidance in this scenario, the capability of reacting immediately to appearing obstacles and guiding the rotorcraft along feasible evasive trajectories is required. This paper presents an approach to short-term collision avoidance based on model predictive techniques. The proposed method, originally developed for automotive applications, finds optimal control inputs by predicting a set of trajectories utilizing a model resembling the helicopter dynamics. Compared to model predictive control, no iterative optimization is adopted, resulting in deterministic execution time. The proposed method is evaluated by closed-loop simulations with a non-linear helicopter model. Additional hardware-in-the-loop simulations are conducted to examine the real-time capability of the approach.

Keywords Model predictive control · Rotorcraft · Helicopter · ACT/FHS · Collision avoidance · Flight control

1 Introduction

Over the past decades, a focus of rotorcraft research has been the development of advanced automatic flight control systems to increase safety and decrease pilot's workload. A challenging aspect of automatic flight is avoidance of obstacles, especially in uncertain environments. One source of uncertainty in low-altitude flight is given, for instance, by the growing numbers of unmanned aerial vehicles (UAV). Driven by the increasing performance of these systems, a broad field of applications has emerged. Besides commercial use, a vast number of small UAVs is operated by private individuals in lower airspace around the world. Despite newly introduced regulations, the operation of UAVs is an area of growing safety concern. The European Aviation Safety Agency (EASA) reported a significant increase in the number of collisions and near collisions in aviation involving UAVs in the last few years [1]. These UAVs are not covered by air traffic control and pose a potential threat to aircraft operating in low altitude, for example helicopter emergency medical services (HEMS). The ability to detect and avoid UAVs and other suddenly appearing obstacles can increase the safety of helicopter missions close to the ground.

Detection and avoidance of obstacles for rotorcraft is being investigated by various authors. Scherer et. al. proposed a system for autonomous flight of a small unmanned helicopter in an unknown environment [2]. The system consists of a laser scanner for obstacle detection, a path planning algorithm, and a reactive collision avoidance method. The control commands are calculated geometrically by evaluating the angles and distances to obstacles and the goal point, respectively. Flight tests with a full-scale helicopter showed the capability of avoiding static obstacles at lower flight speed [3].

Further significant research has been carried out by Goerzen and his colleagues. They presented a system for obstacle field navigation that provides helicopter guidance in low-altitude flight in unknown environment based on sensor data [4]. The capabilities of the system have been demonstrated in extensive flight tests [5]. In their approach, a path to the goal location is generated by creating a risk-based map of the terrain and obstacles in the vicinity of the rotorcraft and

✉ Alexej Dikarew
alexej.dikarew@dlr.de

[1] Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Brunswick, Niedersachsen, Germany

applying a navigation function that provides a minimum-risk path. To command the helicopter to the goal, a velocity command controller is used that follows the navigation function. A simple model based on fixed acceleration limits is utilized to generate feasible velocity commands. In recent work [6], the authors presented an improved method to generate the velocity commands. For this approach, a reachable set is estimated by calculating constant turn-radius paths for different lateral accelerations. The trajectory with lowest risk is chosen and the corresponding command is applied.

An approach to explicitly consider the system dynamics in trajectory generation is given by model predictive control (MPC). The idea is to predict the system dynamics by calculating the output of a model and find control inputs by optimization. A major drawback of MPC is the high computational cost and proof of convergence under real-time constraints. Non-linear MPC has been used for collision-free guidance of an unmanned helicopter [7]; however, a simple non-linear translational model has been used to reduce the computational cost.

To overcome the disadvantages of MPC, model predictive path integral control (MPPI) has been proposed by Williams et. al. [8]. For this method, a finite number of trajectories is being sampled by generating random deviations of the control input and applying them to the model over a prediction horizon. The optimal trajectory is chosen by evaluating a cost function and the control sequence corresponding to it is set as input to the system. The trajectory sampling eliminates the need for explicit optimization and grants the possibility to parallelize the computations, i.e., on graphic processor units. Recently, the MPPI method has been adopted in different fields of aerospace research. Comandur et. al. used the approach for optimizing trajectories in automatic helicopter ship deck landings to compensate for random ship movement [9–11]. In [12], the method is applied for controlling an aircraft on a racecourse, while [13] proposed MPPI for collision avoidance of a fixed-wing aircraft.

Another model predictive approach that avoids explicit optimization is given by sampling a finite set of predicted trajectories but instead of using random deviations of the control input, the allowed range is roughly discretized and held constant for each trajectory over the prediction horizon. This method has been proposed for collision avoidance in automotive applications [14, 15].

This paper presents a short-term collision avoidance algorithm for helicopters based on the aforementioned model predictive approach with control input range discretization and constant input over the prediction horizon. It is intended to support hands-on flight as well as automatic flight with trajectory following in situations with sudden changes in environment and non-cooperative obstacles.

Besides rotorcraft applications, collision avoidance has received a lot of attention in UAV research as well. Yasin et. al. categorized the various approaches in the field into four main categories: geometric, force field, optimization, and sense and avoid [16]. In the following, some recent methods are briefly described and compared to the method proposed in this paper.

In geometric approaches, position and velocity information of the aircraft and obstacles are analyzed to ensure minimum distances between them, as proposed in [17]. As the aircraft dynamics are not considered, these approaches are typically not well suited for short-term collision avoidance.

Force-field methods are based on the artificial potential field algorithm, where first application in collision avoidance has been proposed for robot manipulators [18]. The algorithm calculates forces that attract the manipulator to the target location and repel it from obstacles. The resulting force vector is then transferred into position or velocity commands, resulting in smooth trajectories. The algorithm has been applied to multiple UAVs operating in one environment and enhanced to prevent situations where the target can not be reached due to local minima in the potential field [19, 20]. The position of other UAVs is incorporated in the force field at each time instant to handle them as dynamic obstacles. However, the speed and direction of movement of the obstacle UAVs are not considered. Also, the resulting position commands do not take into account the dynamics of the respective UAV, which may lead to unreachable commanded flight states, especially when obstacles are being detected in short distance and fast evasive maneuvers need to be conducted.

Optimization-based methods are often used for long term planning in static environments, as they introduce high computational complexity. Several authors have developed methods to reduce the computational burden, e.g., [21] proposes a real-time capable method for UAV collision avoidance utilizing sample-based path planning. The evasive trajectories are generated by prediction of a kinematic model [22]. Compared to the approach proposed in the paper at hand, the method does not explore the full range of possible evasive maneuvers, due to the random sampling of waypoints. The authors also state that errors of the prediction due to the simple model can cause the method to fail the task of collision avoidance.

Sense and avoid methods are characterized by short response times and are well suited for dynamic environments. In [23–25], the authors propose an algorithm that supports UAV pilots in manual flight by avoiding collisions with surrounding walls. They use a model of the UAV dynamics to predict a trajectory based on the pilot's current input. If the predicted trajectory collides with an obstacle detected by a 2D ranging sensor, the pilot input is overridden to avoid the collision. To find a collision-free input, the resulting position is approximated based on the predicted trajectory. In the approach proposed in this contribution, no approximation is needed as the control input is chosen based on a set of predicted trajectories.

The remaining paper will be structured as follows: First, an overview of the proposed method is given with a detailed description of the components. Then, the capability of the collision avoidance system is evaluated based on closed-loop simulations. In addition, results for real-time simulations at DLR's flight simulation facility are presented. The paper closes with a conclusion of the findings and an outlook to future work.

2 Method

The proposed model predictive collision avoidance (MPCA) algorithm relies on an underlying flight control system that stabilizes the helicopter, tracks commanded flight states and provides axis decoupling. The implemented method utilizes the model-based control (MBC) flight control system of DLR's research helicopter ACT/FHS [26–28] as the underlying flight controller. The overall system architecture is shown in Fig. 1. The MPCA algorithm generates control inputs to the MBC to guide the rotorcraft around obstacles. To achieve this, an approximation of the closed-loop helicopter flight dynamics is predicted from the current flight state over a receding time horizon for a set of constant control inputs. The resulting predicted trajectories are then evaluated by applying a suitable cost function. The control input corresponding to the trajectory with lowest cost is applied to the MBC flight controller for the current time step. In the next time step, the initial flight state for prediction is updated with the currently measured flight state of the helicopter and the algorithm is executed again. By applying this feedback a closed-loop control system for short-term collision avoidance is established.

The following sections will give a detailed description of the three main components of the MPCA algorithm: control input range discretization, prediction model and cost function.

2.1 Control input range discretization

For the generation of predicted trajectories, the corresponding control input δ for the respective control axis needs to be specified. In the proposed approach, this is done by mapping the continuous range of δ into a finite set of values. For this discretization of the control input range, two aspects need to be considered. First, the set of discrete values should cover the full range of δ to represent the entire control authority and hence include the full range of possible maneuvers at the current flight state. Second, the difference between consecutive values should be as small as possible to achieve a high resolution. Applying both requirements to a fixed discretization of δ will result in a large number of control values and therefore a large number of trajectories for prediction. Hence the computational cost will rise.

To reduce the number of trajectories, an adaptive discretization method described in [15] is applied. δ is first discretized over the full range $[\delta_{\mathrm{min}}, \delta_{\mathrm{max}}]$ with a small number of equally distanced values. For each element δ_i of the resulting set of equally distanced values, a polynomial function depending on the current control input value δ_{c} is applied, which increases the density of values in the vicinity of the current value while preserving the full range. The piecewise defined function reads as

$$\delta_{d,i} = \begin{cases} \left(\frac{\delta_{\mathrm{min}}-\delta_{\mathrm{c}}}{\delta_{\mathrm{min}}^3}\right) \cdot \delta_i^3 + \delta_{\mathrm{c}} & \text{for } \delta_{\mathrm{min}} \leq \delta_i < \delta_{\mathrm{c}} \ , \\ \left(\frac{\delta_{\mathrm{max}}-\delta_{\mathrm{c}}}{\delta_{\mathrm{max}}^3}\right) \cdot \delta_i^3 + \delta_{\mathrm{c}} & \text{for } \delta_{\mathrm{c}} \leq \delta_i \leq \delta_{\mathrm{max}} \ . \end{cases} \tag{1}$$

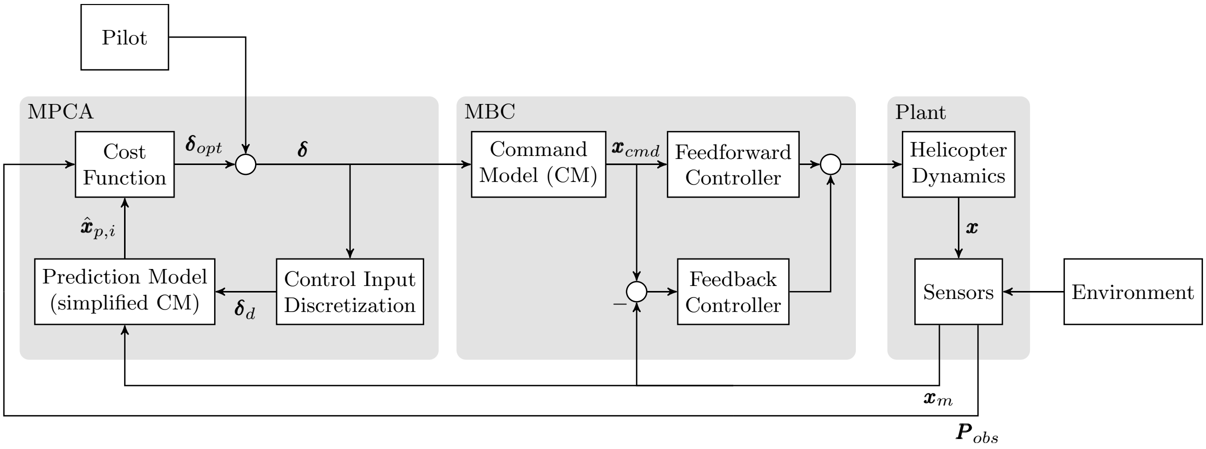


Fig. 1 MPCA system architecture

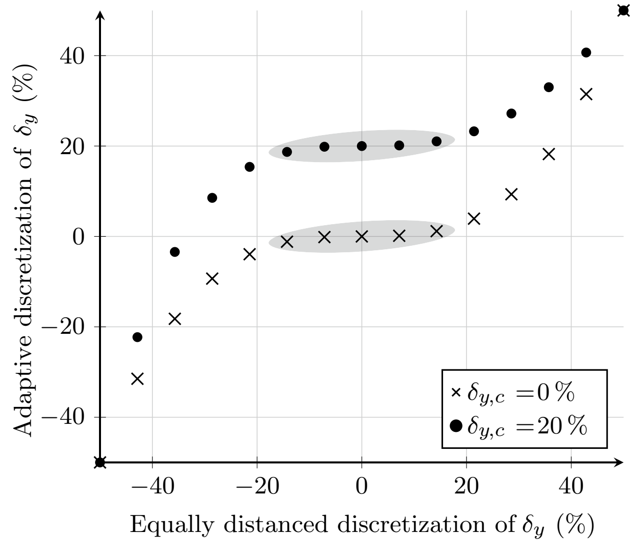
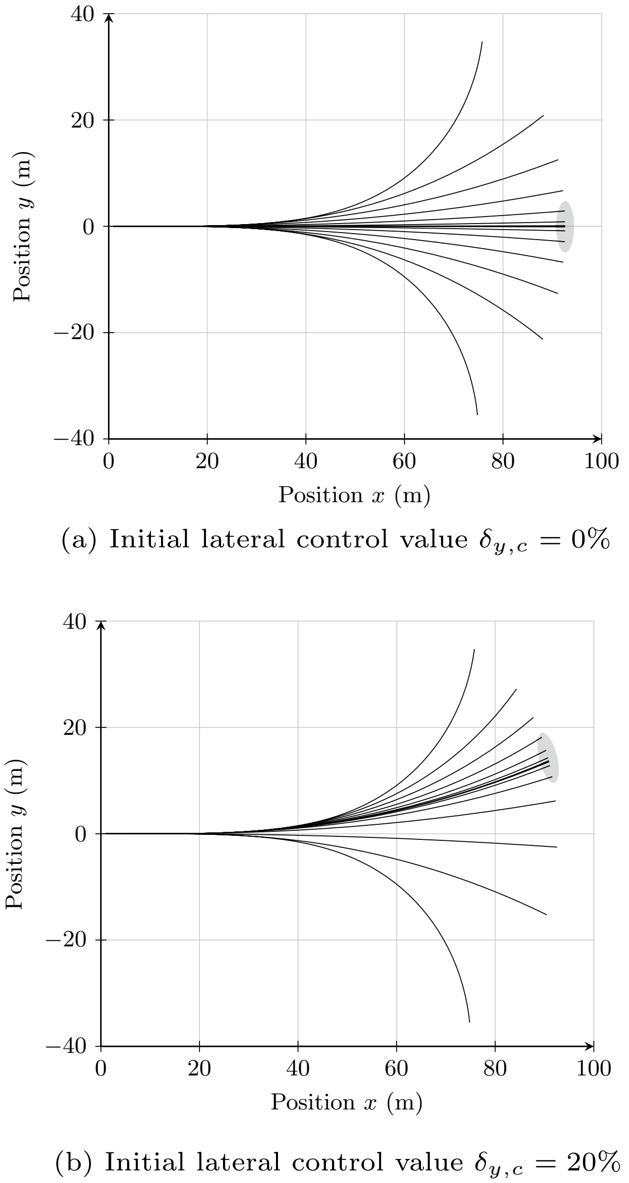


Fig. 2 Discretization of lateral control input

Compared to the method from [15], instead of a second-order polynomial, a third-order equivalent is applied. This provides a higher resolution around the current control value δ_c. Figure 2 shows an example of the applied method for discretization of the lateral cyclic input δ_y. The plot displays the value distributions for $\delta_{y,c}$ = 0% and $\delta_{y,c}$ = 20%, respectively, with δ_{min} = −50% and δ_{max} = 50%. The number of discrete values in the set for both cases is 15. As can be seen, the area with highest density (highlighted areas in the plot) is located around the respective current value of δ_y.

Utilizing a suitable model, which will be described in the next section, the approximated closed-loop helicopter dynamics can be predicted for the set of discretized control input values $\boldsymbol{\delta}_d$. To do so, for each element in the set, the dynamics are calculated with the respective control input value held constant over the prediction horizon. This yields one predicted trajectory for each element in the set of control input values. Figure 3 shows the results of the rotorcraft's predicted position in the x–y-plane over a prediction time t_p = 3 s for the lateral control input $\boldsymbol{\delta}_{y,d}$ shown in Fig. 2. The initial state is steady flight with a forward velocity of 30m/s. Figure 3a corresponds to $\delta_{y,c}$ = 0%, while Fig. 3b complies to $\delta_{y,c}$ = 20%. The high resolution of samples around the initial control value $\delta_{y,c}$ results in a high density of predicted positions around the trajectory corresponding to the initial control value (highlighted areas in the plots). At the same time, the range of reachable predicted positions is not altered by the initial value of $\delta_{y,c}$, thus preserving the maximum range of maneuvers.

The described discretization method is executed for each control axis at every time step of the MPCA algorithm. The resulting set of discrete control inputs $\boldsymbol{\delta}_d$ is then passed as the input to the prediction model.



(a) Initial lateral control value $\delta_{y,c} = 0\%$

(b) Initial lateral control value $\delta_{y,c} = 20\%$

Fig. 3 Predicted trajectories for adaptively discretized lateral control input

2.2 Prediction model

To predict the rotorcraft's flight state for a given input, a suitable numerical model is needed. This prediction model should resemble the helicopter dynamics with sufficient fidelity and at the same time be computationally inexpensive to achieve real-time computation.

The MBC flight controller is based upon a model following scheme where the pilot commands a model with desired helicopter dynamics. This command model (CM) resembles a stable helicopter with decoupled axes and good handling qualities. Based on the pilot input, the CM generates desired flight states $\boldsymbol{x}_{cmd}$ that are being tracked by a combination of a feedforward and a feedback controller, see Fig. 1. A detailed description of the CM used in the MBC flight controller can be found in [26] and [27].

Provided that the controller tracks the commanded states with sufficient performance, the CM approximately represents the closed-loop dynamics of the helicopter with the MBC flight controller. Therefore, it is suited to serve as the MPCA algorithm's prediction model. Compared to a linear helicopter model, the computational effort of the CM is higher; however, it provides predictions with higher fidelity, which is necessary as the MPCA is intended to be used in critical situations close to obstacles. For the implementation of the prediction model, several modifications and simplifications compared to the original CM have been applied. The resulting prediction model is a six degree-of-freedom model accounting for rigid-body dynamics. The state-space equations are given by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + f\left(\boldsymbol{x} + \boldsymbol{x}_{\mathrm{tc}} + \boldsymbol{x}_{\mathrm{trim}}\right) - f(\boldsymbol{x}_{\mathrm{trim}}) + \boldsymbol{B}\boldsymbol{u}\,, \\ \boldsymbol{x}_{\mathrm{mo}} &= \boldsymbol{x} + \boldsymbol{x}_{\mathrm{tc}} + \boldsymbol{x}_{\mathrm{trim}}\,, \end{aligned} \tag{2}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ denote the stability derivative and control derivative matrices, $\boldsymbol{x} = \left[u_f\ v_f\ w_f\ p\ q\ r\right]^T$ denotes the state vector in body frame of reference with the velocities u_f, v_f, w_f and angular rates p, q, r. The velocities for trimmed flight states based on recorded trim angles of the ACT/FHS are incorporated by $\boldsymbol{x}_{\mathrm{trim}}$, while $f(\boldsymbol{x} + \boldsymbol{x}_{\mathrm{tc}} + \boldsymbol{x}_{\mathrm{trim}})$ and $f(\boldsymbol{x}_{\mathrm{trim}})$ account for the gravitational forces at the flight states $\boldsymbol{x} + \boldsymbol{x}_{\mathrm{tc}} + \boldsymbol{x}_{\mathrm{trim}}$ and $\boldsymbol{x}_{\mathrm{trim}}$, respectively. Coordinated turns with $\dot{v}_f = 0$ are achieved by applying a correction of angular rates $\boldsymbol{x}_{\mathrm{tc}}$. The model output state vector accounting for trim angles and turn coordination is given by $\boldsymbol{x}_{\mathrm{mo}}$. The vector $\boldsymbol{u}$ incorporates the control inputs and is being generated by the upper modes, which calculate velocity and angular rate commands based on the control input $\boldsymbol{\delta}$. For cyclic inputs, the attitude command attitude hold (ACAH) mode has been implemented. It commands pitch angles proportional to the longitudinal cyclic stick deflection and roll angles proportional to the lateral cyclic stick deflection, respectively. As only coordinated turns shall be flown by the MPCA and the turn coordination (TC) is implemented in the CM itself, no pedal inputs and hence no upper mode for the yaw axis needs to be considered. For the heave axis the Rate Command Height Hold (RCHH) mode is implemented, which commands a vertical velocity proportional to the collective stick deflection and holds the current altitude when no input is given.

Compared to the original CM, the states filter for the calculation of the second derivative of the state vector $\ddot{\boldsymbol{x}}$ has been omitted, as $\ddot{\boldsymbol{x}}$ is only needed for feedforward control. Furthermore, the original CM implements additional upper modes, i.e., rate command attitude hold (RCAH) and translational rate command (TRC), to provide various control responses. As for the MPCA only one upper mode per axis is required, additional modes have not been implemented in the prediction model.

The resulting model provides a close approximation of the command model dynamics and thus allows for predictions of the closed-loop helicopter response. However, it does not consider controller tracking errors and external disturbances, i.e., the influence of wind.

To calculate trajectories utilizing the prediction model, a differential equation solver has to be applied. To minimize the computational effort of the calculation, the forward Euler method has been chosen. Adapting the model Eq. 2 to the solver yields

$$\begin{aligned} \hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{x}}(k) & \\ + h \cdot \Bigg(\boldsymbol{A}\hat{\boldsymbol{x}}(k) &+ f\Big(\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{x}}_{tc}(k) + \hat{\boldsymbol{x}}_{\mathrm{trim}}(k)\Big) \\ &- f\big(\hat{\boldsymbol{x}}_{\mathrm{trim}}(k)\big) + \boldsymbol{B}\hat{\boldsymbol{u}}(k)\Bigg)\,, \\ \hat{\boldsymbol{x}}_{\mathrm{mo}}(k) = \hat{\boldsymbol{x}}(k) &+ \hat{\boldsymbol{x}}_{tc}(k) + \hat{\boldsymbol{x}}_{\mathrm{trim}}(k)\,, \end{aligned} \tag{3}$$

with h denoting the integration step size, k resembling the prediction step at time t_k and $k+1$ resembling the subsequent prediction at $t_{k+1} = t_k + h$. The notation with hat symbol denotes quantities calculated with the solver.

For the evaluation of the cost function described in the next section, besides the predicted body frame velocities and angular rates, further states are required. Therefore, at each prediction step, additional states are being calculated, including positions

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{y}(k+1) \\ \hat{z}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{z}(k) \end{bmatrix} + \hat{\boldsymbol{T}}_{gf}(k) \cdot h \cdot \begin{bmatrix} \hat{u}_f(k) \\ \hat{v}_f(k) \\ \hat{w}_f(k) \end{bmatrix}\,, \tag{4}$$

Euler angles

$$\begin{bmatrix} \hat{\phi}(k+1) \\ \hat{\theta}(k+1) \\ \hat{\psi}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{\phi}(k) \\ \hat{\theta}(k) \\ \hat{\psi}(k) \end{bmatrix} + \hat{\boldsymbol{T}}_{vf}(k) \cdot h \cdot \begin{bmatrix} \hat{p}(k) \\ \hat{q}(k) \\ \hat{r}(k) \end{bmatrix}\,, \tag{5}$$

and additional velocities

$$\begin{bmatrix} \hat{u}_v(k) \\ \hat{v}_v(k) \\ \hat{w}_v(k) \end{bmatrix} = \hat{\boldsymbol{T}}_{vf}(k) \cdot \begin{bmatrix} \hat{u}_f(k) \\ \hat{v}_f(k) \\ \hat{w}_f(k) \end{bmatrix}\,, \tag{6}$$

where $\hat{\boldsymbol{T}}_{gf}(k)$ denotes the transformation from body fixed frame to local North-East-Down (NED) frame and $\hat{\boldsymbol{T}}_{vf}(k)$ denotes the transformation from body fixed frame to a frame where the z-axis is aligned with the local NED z-axis and the x-axis is horizontally aligned with the body x-axis (vertical frame). The overall predicted state vector at prediction step k reads as $\hat{\boldsymbol{x}}_p(k) = \left[\hat{u}_f\ \hat{v}_f\ \hat{w}_f\ \hat{p}\ \hat{q}\ \hat{r}\ \hat{x}\ \hat{y}\ \hat{z}\ \hat{\phi}\ \hat{\theta}\ \hat{\psi}\ \hat{u}_v\ \hat{v}_v\ \hat{w}_v\right]^T$. With

Eqs. 3–6, the prediction model dynamics are being computed from $k = 1$, with $\hat{\boldsymbol{x}}_p(k = 1) = \boldsymbol{x}_m$ holding the measured state vector, to $k = n_\mathrm{p}$ with $n_\mathrm{p} = t_\mathrm{p}/h$ holding the prediction horizon. Performing this calculations for the vector of discretized control input values $\boldsymbol{\delta}_\mathrm{d}$ provides a set of predicted trajectories $\hat{\boldsymbol{x}}_{p,i}$.

2.3 Cost function

To guide the rotorcraft along a collision-free path, at each time step the most suitable predicted trajectory out of the complete set needs to be identified. This is realized by applying a cost function to the predictions. The trajectory with the smallest cost is considered as optimal with respect to the cost function and the corresponding control input $\boldsymbol{\delta}_\mathrm{opt}$ is set as input to the flight controller for the current time step.

For the design and weighting of the cost function components, the insights from [14] and [15] have been used and adopted to the application in rotorcraft flight. Therefore, three aspects have been considered: First, the cost function shall provide collision-free trajectories. Second, the dynamic reaction of the rotorcraft while avoiding obstacles shall be minimized. Third, if no collision is predicted, the rotorcraft shall follow reference flight states. The resulting cost function reads as

$$J = \sum_{k=1}^{n_\mathrm{p}} \left(W_1 J_{1,k} + W_2 J_{2,k} + W_3 J_{3,k} \right) . \tag{7}$$

The cost function value is calculated at each prediction step k and summed up over the prediction horizon n_p to obtain the overall cost for each trajectory, respectively. J_1, J_2, J_3 resemble the cost function components, which will be explained in the following. W_1, W_2, W_3 hold the weight for each corresponding component.

The first component J_1 accounts for avoiding collisions. Therefore, at each prediction step, the distance $d_{\mathrm{obs},k}$ between the rotorcraft's predicted position $\left[x_{p,k}\ y_{p,k}\ z_{p,k}\right]$ and the obstacle's position $\boldsymbol{P}_{\mathrm{obs},k}$ is calculated. Depending on the distance, the collision avoidance component is given by

$$J_1 = \begin{cases} 2 - \dfrac{1}{{d_\mathrm{s}}^2} d_{\mathrm{obs},k} & \text{if } d_{\mathrm{obs},k} \leq d_\mathrm{s} \ , \\ \dfrac{\left(d_{\mathrm{obs},k} - (d_\mathrm{s} + d_f)\right)^2}{{d_f}^2} & \text{if } d_\mathrm{s} < d_{\mathrm{obs},k} < d_\mathrm{s} + d_f \ , \\ 0 & \text{if } d_{\mathrm{obs},k} \geq d_\mathrm{s} + d_f \ , \end{cases} \tag{8}$$

and is diagrammed in Fig. 4. If the predicted distance to the obstacle $d_{\mathrm{obs},k}$ is less than the minimum safety distance d_s, a collision is predicted and the cost value is set to a value greater than 1. A quadratic function is applied that increases the cost value with decreasing distance. This ensures that trajectories with higher distance to the obstacle are chosen when all predicted trajectories collide with the safety margin. If the predicted position of the helicopter is not colliding but is within a certain range to the obstacle $d_\mathrm{s} < d_{\mathrm{obs},k} < d_\mathrm{s} + d_f$, another quadratic function is applied that generates rising cost with decreasing distance. The range for this fade-in of the cost has been set to $d_f = 5$ m. This measure removes sharp discontinuities of the cost value in the vicinity of the obstacle, which caused undesired behavior in early experiments, i.e., oscillations of the control inputs. Finally, if the rotorcraft's position is outside the fading distance $d_{\mathrm{obs},k} \geq d_\mathrm{s} + d_f$, the cost is set to $J_1 = 0$.

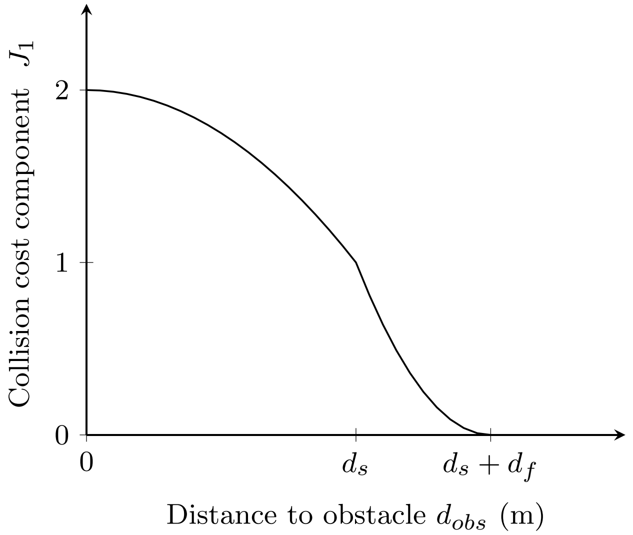


Fig. 4 Collision cost dependent on distance to obstacle

The cost function component J_2 accounts for minimizing the dynamic reaction of the helicopter while avoiding a collision with a nearby obstacle to prevent unstable flight conditions. This is achieved by minimizing the angular rates p and q. The turn rate r will not be considered, as this would penalize steady turn flight. Therefore, the cost function component for minimizing dynamic reactions yields

$$J_2 = |\hat{p}| + |\hat{q}| . \tag{9}$$

The third cost function component J_3 accounts for the rotorcraft to maintain a reference flight state while no collision is predicted along the current flight path or right after a collision has been avoided. If evasive maneuvers with only lateral cyclic control inputs shall be performed, the deviation of the predicted yaw angle $\hat{\psi}_p$ to the reference track angle χ_ref has to be considered

$$J_{3,y} = |\hat{\psi}_p - \chi_\mathrm{ref}| . \tag{10}$$

If maneuvers with longitudinal and lateral cyclic inputs shall be executed, additionally the deviation of the predicted forward velocity $\hat{u}_{v,p}$ to the reference velocity $u_{v,\mathrm{ref}}$ needs to be incorporated

$$J_{3,x,y} = |\hat{\psi}_p - \chi_{\mathrm{ref}}| + |\hat{u}_{v,p} - u_{v,\mathrm{ref}}|. \tag{11}$$

Weighting of the different components is a crucial step in the design of the cost function, as the overall system behavior strongly relies on correct ratios between the several weights. Badly set weights can even cause the algorithm to force collisions instead of avoiding them. For this reason, the cost function has been designed to utilize only few weights which can be tuned manually.

First, critical situations will be considered: As avoiding collisions has highest priority, setting $W_1 >> W_2$ and $W_1 >> W_3$ is mandatory. For non-critical situations, obtaining reference states needs to be weighted higher than minimizing dynamic reactions, as a change of flight state can only be achieved by applying angular rates, hence $W_3 >> W_2$ can be set. Combining both requirements provides the overall setup rule $W_1 >> W_3 >> W_2$.

3 Evaluation

To evaluate the capability of the proposed model predictive approach to short-term collision avoidance and tune the algorithm's parameters, software-in-the-loop simulations have been conducted. The simulation setup utilizes the flight controls development tool chain of the research helicopter ACT/FHS. The MPCA algorithm has been added to the implementation of the MBC in MATLAB/Simulink®. The helicopter flight dynamics are resembled by a physics-based non-linear model of the ACT/FHS which is also used for real-time simulations at DLR.

First, the parameter values of the algorithm are defined. The number of sampled trajectories has been set to 15 per control axis. This value has been found to be a feasible trade-off between sufficiently high discretization and computational effort. With 15 trajectories, each in the lateral and the longitudinal cyclic axis, a total of 225 trajectories is being predicted. For the definition of the prediction horizon, typically the time is considered that is needed to decelerate the rotorcraft to hover. This ensures that a collision with a straight-ahead obstacle, which can not be dodged laterally, can be avoided. For this study, the prediction horizon is intentionally set to a lower value, as only obstacles in the order of magnitude of the helicopter are considered, which can be avoided by lateral maneuvers. Furthermore, the performance of the system shall be evaluated for short-term collision avoidance, therefore the prediction horizon has been set to $t_\mathrm{p} = 3\,\mathrm{s}$. To achieve a sufficiently high quality of the prediction, the integration step size is set to $h = 30\,\mathrm{ms}$, thus each predicted trajectory consists of $n_\mathrm{p} = 100$ prediction steps. This grants a spatial resolution of the predicted trajectories of about 1 m at 30 m/s airspeed. Finally, the cost function weights are being set. Applying the setup rule described in the previous section and evaluation in different scenarios results in $W_1 = 10, W_2 = 0.1, W_3 = 1$.

To evaluate the algorithm's collision avoidance capability in cruise flight, a scenario with two spherical, stationary obstacles has been defined. The obstacles have a radius $R_\mathrm{obs} = 10\,\mathrm{m}$ with a safety margin $d_\mathrm{s} = 10\,\mathrm{m}$. The initial state of the helicopter is steady flight with $u_v = 30\,\mathrm{m/s}$ and $\psi = 0$. The flight path collides with the obstacle, forcing the helicopter to conduct an evasive maneuver. The resulting trajectory as well as attitude and control input are shown in Fig. 5. The upper three plots show the trajectory of the helicopter in the x–y-plane at different points in time. At each point in time the predicted trajectories for lateral control inputs are shown. The coloring separates collision-free from colliding trajectories and highlights the trajectory with lowest cost.

At the beginning of the maneuver, the helicopter approaches the first obstacle. At a position $x \approx 40\,\mathrm{m}$, the predicted trajectories meet the obstacle, resulting in a predicted collision and causing the helicopter to introduce a left turn with a maximum roll angle $\phi > 30\,\mathrm{deg}$. While closing in on the obstacle, the roll angle is reduced to conduct a right turn to capture the reference track angle $\chi_\mathrm{ref} = 0$. At a position $x \approx 150$ m a collision with the second obstacle is predicted, causing the helicopter to increase the roll angle to the right. After passing the second obstacle the reference track angle $\chi_\mathrm{ref} = 0$ is reestablished. During the maneuver the helicopter stays clear of the safety margins and hence successfully avoids collisions. To minimize the roll rate, however, it is passing the obstacles very close to their boundaries. The evasive maneuver requires high amplitude lateral cyclic control inputs exceeding 40%. At the same time, there are only low-amplitude inputs in the longitudinal cyclic axis and hence small change of pitch angle, resulting in a nearly constant velocity. This is due to the design of the cost function favoring lateral maneuvers over variation of speed.

The evaluation shows the capability of the method to provide collision avoidance for obstacles appearing close to the aircraft, yet there are still some shortcomings that need further improvement. As the cost function does not include a direct penalization of control inputs, the algorithm generates high amplitude and high rate control inputs which may excite oscillations. Furthermore, the prediction model reflects the desired closed-loop dynamics and hence does not explicitly include limitations of the helicopter performance and does not account for tracking errors of the controller. This may lead to predictions diverging from the aircraft behavior, especially close to the borders of the flight envelope. These shortcomings result in oscillations of the controls, as can be seen in the plot of δ_y, especially for positions $140\,\mathrm{m} \leq x \leq 220\mathrm{m}$.

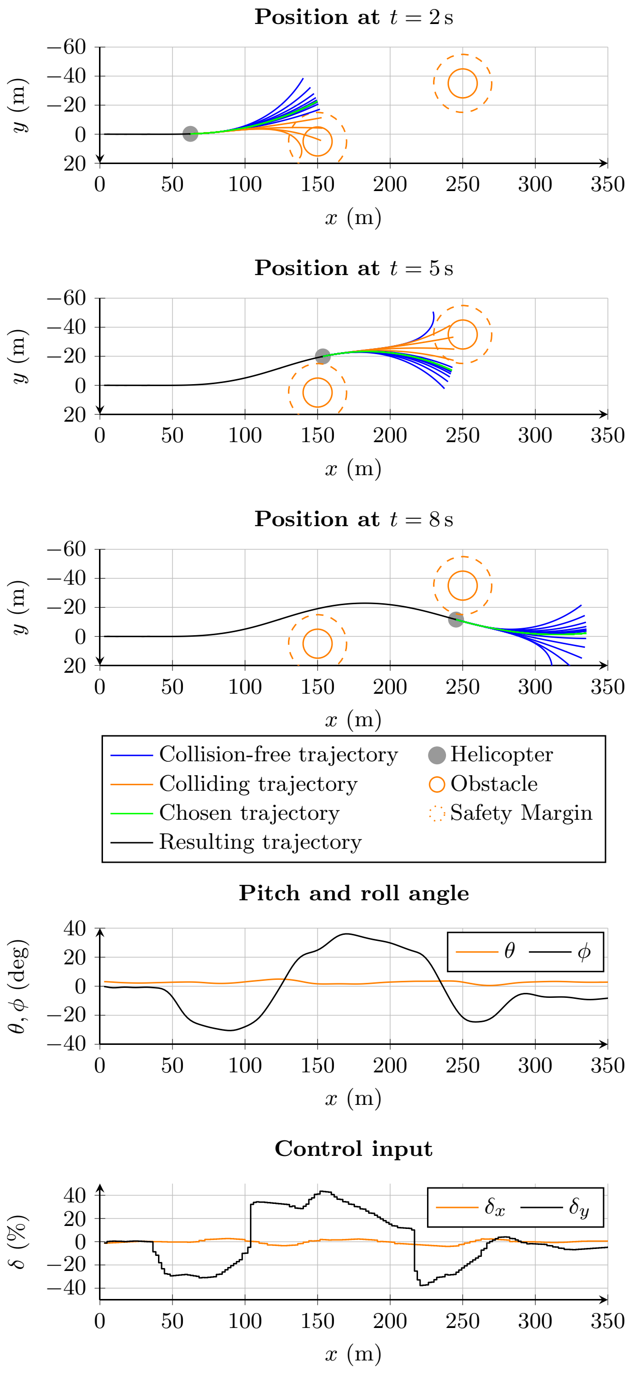


Fig. 5 Evaluation results for closed-loop simulation

4 Real-time simulation

In the previous section, the capability of the MPCA algorithm to successfully avoid collisions has been shown using closed-loop simulations. To evaluate the system performance under real-time conditions, the algorithm is implemented on an experimental flight control computer and tested in hardware-in-the-loop simulations with DLR's ground based flight simulation facility, the Air Vehicle Simulator (AVES, [29]). In the AVES, the flight control computer is set up in a control loop with a non-linear physics based model of the ACT/FHS flight dynamics together with supplementary simulation models such as actuator dynamics and sensor simulation.

The flight control computer is an exact replication of the experimental hardware of the ACT/FHS, which offers a platform for in-flight evaluation of flight control algorithms. To achieve this, the research helicopter is equipped with a high authority experimental system with fly-by-wire connection to electro-hydraulical actuators. The flight control computer generates actuator commands based on pilot inputs and the implemented control algorithms and sends them to the actuators. Further details on the system architecture can be found in [27] and [28].

The experimental computer has been specifically designed to execute flight control algorithms. Therefore, it implements hard real-time constraints, i.e., a fixed cycle time of 8 ms. The implemented flight control algorithms should not exceed 6 ms per cycle to retain sufficient computation time for other processes. The flight controller MBC requires approximately 2 ms each cycle, thus the MPCA may use a maximum of 4 ms. However, for the implementation with one active control axis utilizing a set of 15 predicted trajectories, an execution time on the experimental system hardware of approximately 30 ms is needed. To meet the real-time requirements, the MPCA algorithm's calculations are distributed by spreading the model prediction and cost calculation over eight cycles of the flight control system execution. As a result, the algorithm generates updated control inputs to the MBC each 64 ms.

For the evaluation of the MPCA in the real-time simulation, a low radio tower is defined as obstacle, which is displayed in Fig. 6. As the implementation of the MPCA system does not yet utilize sensor information for the detection, the tower top is resembled by a sphere with radius $R_{\mathrm{obs}} = 5\,\mathrm{m}$ at predefined coordinates. The safety margin to the obstacle again has been set to $d_s = 10\,\mathrm{m}$. Due to the mentioned limitations in execution time, evasive maneuvers with only lateral cyclic control input with 15 predicted trajectories are evaluated. The remaining parameters of the algorithm are set according to the evaluation setup in the previous section with $t_p = 3\,\mathrm{s}$, $h = 30\,\mathrm{ms}$ and $W_1 = 10$, $W_2 = 0.1$, $W_3 = 1$. The initial state of the helicopter is again steady flight with $u_v = 30\,\mathrm{m/s}$ and $\psi = 0$. The initial position is set to achieve an initial flight path colliding with the radio tower. The resulting trajectory of the maneuver as well as roll angle and control input is shown in Fig. 7. The upper three plots show the trajectory of the helicopter in the x–y-plane at different points in time. At each point in time the predicted trajectories are shown. The coloring separates collision-free from colliding trajectories and highlights the chosen trajectory.

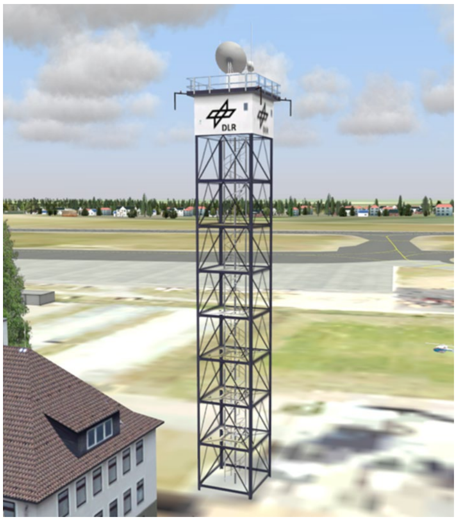


Fig. 6 Obstacle in real-time simulation: radio tower

The simulation starts with the helicopter approaching the tower top. At a position $x \approx 50\,\mathrm{m}$ a collision is predicted and the helicopter introduces a right turn. While passing the obstacle, the roll angle is reduced to conduct a left turn to capture the reference track angle $\chi_{\mathrm{ref}} = 0$. After the helicopter has passed the tower, the reference track angle is reestablished. A collision with the tower is successfully avoided, yet the resulting trajectory lies close to the obstacle to minimize the roll rate during the maneuver. Compared to the evaluation described in the previous section, the necessary control input amplitude and, thus, the resulting roll angle amplitude is lower. This is due to the smaller diameter of the obstacle in the real-time scenario. Figure 8 displays a sequence of images of the maneuver. Figure 8a shows the helicopter conducting a right turn shortly after a collision with the tower is predicted. In Fig. 8b the helicopter is approaching the obstacle while maintaining a sufficiently large lateral distance. Finally, in Fig. 8c the helicopter introduces a left turn to reestablish the reference track angle after passing the tower.

The simulation results show the real-time capability of the MPCA method for lateral evasive maneuvers. Utilizing a set of predicted trajectories with constant control input yields a promising solution to the collision avoidance task. However, implementing the algorithm into the experimental system of the ACT/FHS shows that the proposed method is computationally expensive and can easily exceed available flight control hardware capabilities. For the present setup, control discretization and model prediction can only be achieved for one control axis at a time.

Nevertheless, specialized hardware may exploit the structure of the algorithm. As calculations of the predicted trajectories are independent of one another, the algorithm is a suitable candidate for parallel computation. Therefore, future work will investigate the implementation of the algorithm on graphic processor based hardware.

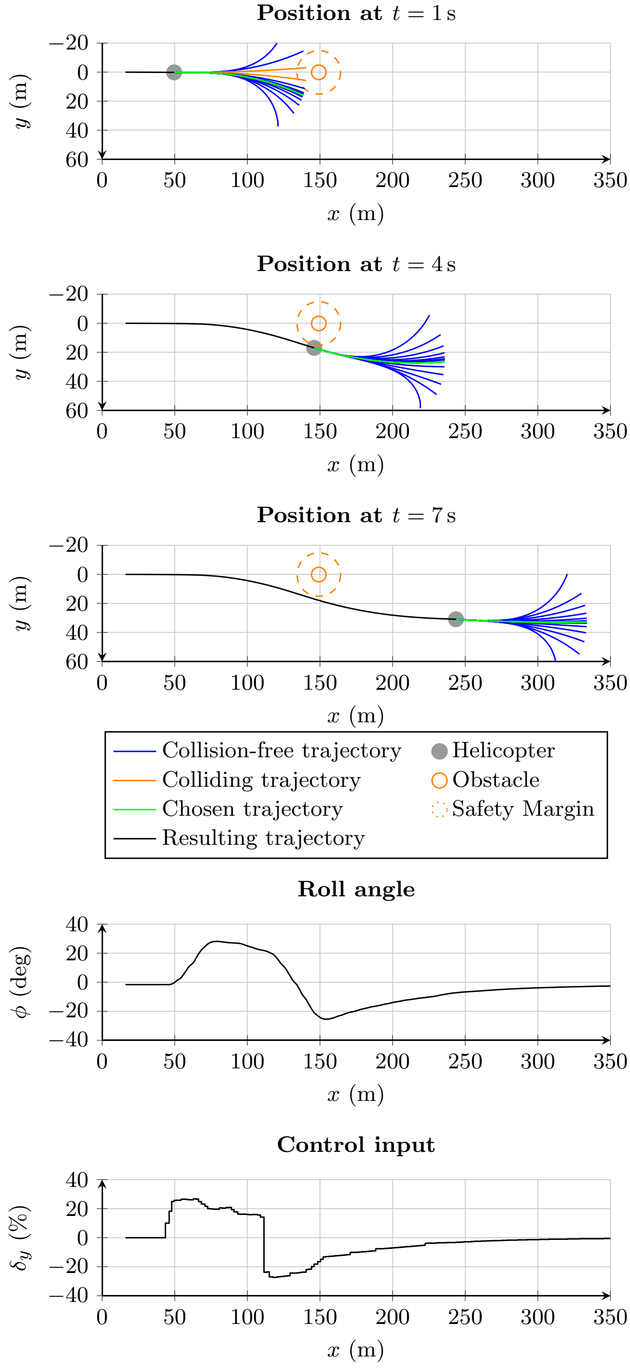
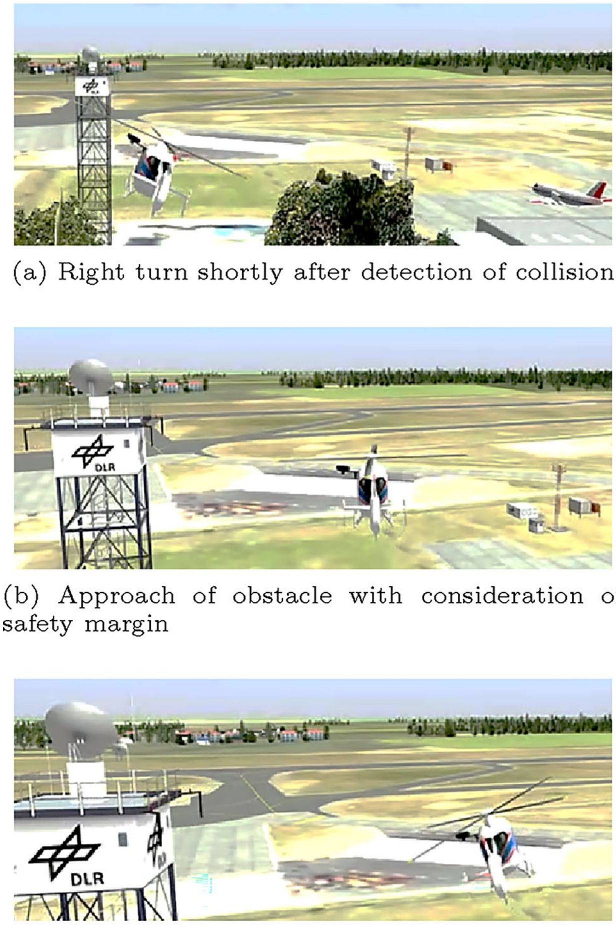


Fig. 7 Evaluation results for real-time hardware-in-the-loop simulation



(a) Right turn shortly after detection of collision

(b) Approach of obstacle with consideration of safety margin

(c) Left turn to reestablish reference track angle

Fig. 8 Visualization of evasive maneuver in real-time simulation

5 Concluding remarks

This paper proposes a model predictive approach, previously used in automotive driver assistance systems, for short-term collision avoidance for helicopters. The method solves an optimal control problem by evaluating the output of a set of trajectories which are generated by predicting a model of the system dynamics with roughly discretized control inputs over a short prediction horizon.

First, the method is adapted to be suitable for rotorcraft applications and to serve as an extension of the model-based flight controller of DLR's research helicopter. The three main components of the algorithm, control input range discretization, prediction model and cost function, are developed. While the adaptive discretization technique is slightly modified, the prediction model is derived from the command model of the underlying flight controller. This six degree-of-freedom model resembles the desired closed-loop dynamics of the helicopter. The cost function is designed to provide collision-free trajectories, while at the same time account for reasonable maneuvers.

The resulting model predictive collision avoidance (MPCA) system is implemented into the flight control tool chain of the research helicopter and evaluated in closed-loop simulations with a non-linear helicopter model. The implementation supports longitudinal and lateral cyclic inputs. The evaluations show the capability of the system to guide the helicopter along collision-free trajectories while keeping control intervention to a minimum. Some shortcomings of the current implementation, which need further improvement, are identified, i.e., insufficiencies of the cost function and deviations of the dynamics of the prediction model from the helicopter simulation model. To investigate the real-time capability of the MPCA, it is additionally evaluated in real-time hardware-in-the-loop simulations at DLR's flight simulation facility. Limitations of computational resources of the experimental flight control system allow only for prediction in one control axis at a time, constraining the application to lateral evasive maneuvers. The real-time simulations show the capability of the algorithm to successfully conduct evasive maneuvers under real-time constraints.

The results in this paper provide a proof of concept for the MPCA as a short-term collision avoidance system for helicopters. Future work will include the enhancement of the system to extend the possible maneuvers for obstacle avoidance, i.e., by ascending or descending. Also, the current implementation does not account for prediction uncertainties which may be caused by model uncertainties and disturbances, for example the influence of wind. Therefore, the system will be extended to consider prediction errors that arise from tracking errors of the underlying flight controller. Furthermore, the design and parameterization of the cost function will be refined to explicitly consider the flight envelope and the control value history. Another focus of future work will be set on the implementation of the MPCA method on specialized hardware. The structure of the algorithm allows for parallel computation, as the predicted trajectories are being calculated independent of each other. Implementation of the system on graphic processor based hardware is expected to strongly decrease the required computation time while at the same time allow for a larger number of predicted trajectories.

Acknowledgements The approach proposed in this paper was previously presented at the Vertical Flight Society (VFS) 77th Annual Forum & Technology Display [30]. The paper at hand describes refinements of the method and provides updated results.

Funding Open Access funding enabled and organized by Projekt DEAL. The author has no relevant financial or non-financial interests to disclose.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. European Aviation Safety Agency.: Annual Safety Review 2020. Publications Office, LU (2020). https://doi.org/10.2822/147804. Accessed 16 Feb 2021
2. Scherer, S., Singh, S., Chamberlain, L., Saripalli, S.: Flying Fast and Low Among Obstacles. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 2023–2029. IEEE, Rome, Italy (2007). https://doi.org/10.1109/ROBOT.2007.363619. http://ieeexplore.ieee.org/document/4209383/. Accessed 30 Mar 2021
3. Scherer, S., Chamberlain, L., Singh, S.: First results in autonomous landing and obstacle avoidance by a full-scale helicopter. In: 2012 IEEE International Conference on Robotics and Automation, pp. 951–956. IEEE, St Paul, MN, USA (2012). https://doi.org/10.1109/ICRA.2012.6225215. Accessed 9 Jan 2022
4. Goerzen, C., Whalley, M.: Minimal risk motion planning: a new planner for autonomous UAVs in uncertain environments. In: AHS International Specialists' Meeting on Unmanned Rotorcraft, Tempe, Arizona, USA, p. 21 (2011)
5. Whalley, M.S., Takahashi, M.D., Fletcher, J.W., Moralez, E., Ott, L.C.R., Olmstead, L.M.G., Savage, J.C., Goerzen, C.L., Schulein, G.J., Burns, H.N., Conrad, B.: Autonomous black hawk in flight: obstacle field navigation and landing-site selection on the RASCAL JUH-60A: autonomous black hawk in flight. J. Field Robot. **31**(4), 591–616 (2014). https://doi.org/10.1002/rob.21511
6. Goerzen, C., Takahashi, M., Whalley, M., Cleary, M.M., Cox, J.: Kinematic velocity commander for obstacle field navigation. In: Vertical Flight Society Autonomous VTOL Technical Meeting and Electric VTOL Symposium, Mesa, Arizona, USA, p. 12 (2019)
7. Suzuki, S., Ishii, T., Aida, Y., Fujisawa, Y., Iizuka, K., Kawamura, T.: Collision-free guidance control of small unmanned helicopter using nonlinear model predictive control. SICE J Control Meas Syst Integr **7**(6), 347–355 (2014). https://doi.org/10.9746/jcmsi.7.347
8. Williams, G., Aldrich, A., Theodorou, E.A.: Model predictive path integral control: from theory to parallel computation. J Guid Control Dyn **40**(2), 344–357 (2017). https://doi.org/10.2514/1.G001921
9. Prasad, J.V.R., Comandur, V., Walters, R., Guerrero, D.: Model predictive path integral approach for trajectory guidance of rotorcraft shipboard landing. In: AHS International 74th Annual Forum, Phoenix, Arizona, USA, p. 16 (2018)
10. Comandur, V., Prasad, J.: Rotorcraft shipboard landing guidance using MPPI trajectory optimization. In: 44th European Rotorcraft Forum, Delft, The Netherlands, p. 14 (2018)
11. Comandur, V., Walters, R., Prasad, J.V.R.: MPPI parallel trajectory optimization for guidance in rotorcraft shipboard landing. In: AHS International 75th Annual Forum, p. 17 (2019)
12. Pravitra, J., Theodorou, E., Johnson, E.N.: Flying complex maneuvers with model predictive path integral control. In: AIAA Scitech 2021 Forum. American Institute of Aeronautics and Astronautics, VIRTUAL EVENT (2021). https://doi.org/10.2514/6.2021-1957
13. Shimizu, Y.: Nonlinear model predictive real-time control of aircraft in collision avoidance. In: AIAA Scitech 2020 Forum. American Institute of Aeronautics and Astronautics, Orlando, FL (2020). https://doi.org/10.2514/6.2020-1829
14. Keller, M., Hass, C., Seewald, A., Bertram, T.: A model predictive approach to emergency maneuvers in critical traffic situations. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 369–374. IEEE, Gran Canaria, Spain (2015). https://doi.org/10.1109/ITSC.2015.69. http://ieeexplore.ieee.org/document/7313161/. Accessed 2021-01-28
15. Homann, A., Buss, M., Keller, M., Glander, K.-H., Bertram, T.: Multi stage model predictive trajectory set approach for collision avoidance. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 945–950. IEEE, Maui, HI (2018). https://doi.org/10.1109/ITSC.2018.8569790
16. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., Plosila, J.: Unmanned aerial vehicles (UAVs): collision avoidance systems and approaches. IEEE Access **8**, 105139–105155 (2020). https://doi.org/10.1109/ACCESS.2020.3000064
17. Mujumdar, A., Padhi, R.: Reactive collision avoidance of using nonlinear geometric and differential geometric guidance. J Guid Control Dyn **34**(1), 303–311 (2011). https://doi.org/10.2514/1.50923
18. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings 1985 IEEE International Conference on Robotics and Automation, vol. 2. Institute of Electrical and Electronics Engineers, St. Louis, pp. 500–505 (1985). https://doi.org/10.1109/ROBOT.1985.1087247. Accessed 2022-03-08
19. Sun, J., Tang, J., Lao, S.: Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. IEEE Access **5**, 18382–18390 (2017). https://doi.org/10.1109/ACCESS.2017.2746752
20. Tang, J., Sun, J., Lu, C., Lao, S.: Optimized artificial potential field algorithm to multi-unmanned aerial vehicle coordinated trajectory planning and collision avoidance in three-dimensional environment. Proc Inst Mech Eng Part G J Aerosp Eng **233**(16), 6032–6043 (2019). https://doi.org/10.1177/0954410019844434
21. Lin, Y., Saripalli, S.: Sampling-based path planning for UAV collision avoidance. IEEE Trans Intell Transp Syst **18**(11), 3179–3192 (2017). https://doi.org/10.1109/TITS.2017.2673778
22. Lin, Y., Saripalli, S.: Path planning using 3D Dubins curve for unmanned aerial vehicles. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 296–304. IEEE, Orlando, FL, USA (2014). https://doi.org/10.1109/ICUAS.2014.6842268
23. Israelsen, J., Beall, M., Bareiss, D., Stuart, D., Keeney, E., van den Berg, J.: Automatic collision avoidance for manually tele-operated unmanned aerial vehicles. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6638–6643. IEEE, Hong Kong, China (2014). https://doi.org/10.1109/ICRA.2014.6907839
24. Bareiss, D., van den Berg, J., Leang, K.K.: Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles. In: 2015 IEEE/RSJ International Conference on Intelligent Robots

and Systems (IROS), pp. 4818–4825. IEEE, Hamburg, Germany (2015). https://doi.org/10.1109/IROS.2015.7354054

25. Bareiss, D., Bourne, J.R., Leang, K.K.: On-board model-based automatic collision avoidance: application in remotely-piloted unmanned aerial vehicles. Autonom Robots **41**(7), 1539–1554 (2017). https://doi.org/10.1007/s10514-017-9614-4
26. Fragniére, B., Wartmann, J., Greiser, S.: Preliminary Results Towards Fidelity Evaluation of an In-flight Simulator. In: AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics, San Diego, California (2019). https://doi.org/10.2514/6.2019-0988
27. Greiser, S., Lantzsch, R., Wolfram, J., Wartmann, J., Müllhäuser, M., Lüken, T., Döhler, H.-U., Peinecke, N.: Results of the pilot assistance system"Assisted Low-Level Flight and Landing on Unprepared Landing Sites" obtained with the ACT/FHS research rotorcraft. Aerosp Sci Technol **45**, 215–227 (2015). https://doi.org/10.1016/j.ast.2015.05.017
28. Kaletka, J., Kurscheid, H., Butter, U.: FHS, the new research helicopter: ready for service. Aerosp Sci Technol **9**(5), 456–467 (2005). https://doi.org/10.1016/j.ast.2005.02.003
29. Duda, H., Advani, S.K., Potter, M.: Design of the DLR AVES Research Flight Simulator. In: AIAA Modeling and Simulation Technologies (MST) Conference. American Institute of Aeronautics and Astronautics, Boston, MA (2013). https://doi.org/10.2514/6.2013-4737
30. Dikarew, A.: Model predictive approach for short-term collision avoidance. In: Proceedings of the Vertical Flight Society 77th Annual Forum, pp. 1–10. The Vertical Flight Society, Virtual Conference (2021). https://doi.org/10.4050/F-0077-2021-16906