2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Dynamic Obstacle Avoidance for UAVs using MPC and GP-Based Motion Forecast

Ertug Olcay, Henri Meeß, Gordon Elger

Abstract— **Dynamic obstacle avoidance is an essential function for Unmanned Aerial Vehicles (UAVs) to ensure the safe and reliable operations of drones in real-world environments. It allows drones to navigate and react to environmental changes in real time, preventing collisions and maintaining their flight paths. Dynamic obstacle avoidance also improves the success rate of the drone's mission by reducing the need for manual control. In this study, we propose a model predictive control (MPC) concept to generate high-level control commands for drones to avoid dynamic obstacles by integrating Gaussian process regression to forecast the motion of the moving obstacle based on noisy observations. Additionally, we also investigated the applicability of the Kalman filter as an alternative approach in this context. Our tests demonstrate promising results for multi-rotor drones in physics-based simulations.**

Index Terms— **Automatic guidance, UAV, motion control, applications.**

I. INTRODUCTION

With the advancement of aircraft technologies and the rising demand for efficient mobility, Unmanned Aerial Vehicles (UAVs) are considered a key element with a large variety of applications in the future. Besides monitoring [1], search and rescue purposes [2], drone technologies have been intensively developed for logistics applications. These are roughly divided into several areas: retail and E-commerce, mail delivery, food delivery, and healthcare as well as emergency services [3]. Especially in dense operation regions, where multiple UAVs are operated, the interaction of drones with the dynamic environment is crucial.

The interaction can usually be either bidirectional or unidirectional. Cooperative scenarios represent the bidirectional type of interaction, where information flow from both directions is provided. In unidirectional or non-cooperative cases, the drone requires sensors for perception and algorithms to track the object and grasp its motion. In this way, appropriate avoidance maneuvers can be executed.

Most of the research in air mobility has been focusing on the aircraft technologies and individual features of drones, e.g., flight stability, remote sensing, and path planning. The interaction with the environment is mostly handled in the context of path and trajectory planning. However, motion planning in an environment with dynamic obstacles is barely addressed.

A significant amount of effort has been put into various approaches to generate reliable trajectories in unknown, dynamic environments and perform collision avoidance with moving obstacles. Different methods have been studied for the autonomous navigation of UAVs. Some of them are based on heuristic assumptions and simple obstacle representations. In [4], artificial potential fields were used to avoid moving obstacles. However, potential field-based approaches usually suffer from local minimum problems. [5] presents a further study using potential functions, where the authors additionally used fuzzy logic to overcome the local minimum problem. A recent study [6] utilized the combination of Dynamic Window Approach (DWA) and RRT* to plan a collision-free path and react to unexpected obstacles. To navigate in an environment with unknown obstacles, Chen et al. [7] presented a real-time path update scheme based on A* algorithm that does not take moving obstacles into account in a predictive fashion.

Alvarez et al. [8] presented the ACAS sXu scheme, which is a decision-making framework that helps autonomous drones fly safely in the air by detecting and avoiding potential collisions for large airspaces.

In recent years, deep learning-based approaches have received considerable attention. Such approaches are usually applied to solve different parts of the obstacle avoidance problem, such as vision-based localization or planning. In [9], a methodology for deep learning-based scene understanding by using monocular depth and ego-motion estimation with an optical flow model is presented to deal with obstacles.

Reinforcement learning (RL) is a different learning technique that has been intensively studied because of the advantage that it can provide intuitive solutions for complex tasks. In [10], with only the information on distance obtained through Lidar, an obstacle avoidance policy for a mobile ground-based robot was derived. The prediction and avoidance of the movements of dynamic obstacles relied on the information of the lidar distance of the present and the past steps. A theoretical RL-based attempt for decision-making to avoid dynamic obstacles in 3D space was presented in [11].

Behavior cloning is another machine learning approach where an agent learns to mimic the behavior of a demonstrated expert to accomplish a task. In [12], a learning-based, perception-aware trajectory planner was developed to perform collision avoidance with dynamic obstacles.

Another popular type of method, which is based on optimization, is MPC. Study [13] presented an MPC-based motion planning framework for a static environment by defining flyable spaces within the prediction time horizon as optimization constraints. In [14], [15], another MPC and optimal control concepts in a multi-agent context for a static

Authors are with Fraunhofer Institute for Transportation and Infrastructure Systems IVI, Ingolstadt, Germany. email: {ertug.olcay, henri.meess, gordon.elger}@ivi.fraunhofer.de

environment are proposed. However, only static obstacles are addressed. An MPC-based approach was also applied to flight motion planning of a UAV in the presence of moving obstacles [16]. The authors predefined certain motion classes and identified them based on some measurements and presumptions. The localization was emulated with an external tracking system, and the approach was experimentally demonstrated. In a further study [17], which is relevant to our work, collision avoidance with moving obstacles was achieved using MPC and Kalman filter, which uses a predefined motion model of the moving obstacle for making predictions about its future state.

However, despite the promising results achieved by various studies, there still exists a research gap when it comes to robustly predicting the motion of arbitrary moving objects in dynamic environments. In real-world scenarios, the motion of objects can be highly unpredictable and prone to noise in detection. This makes it difficult to accurately model their behavior. This lack of robustness can result in a higher risk of collisions and accidents. Therefore, there is a need for more research in this area to develop robust and accurate methods for detecting and predicting the motion of moving objects in dynamic environments.

Data-driven system identification has been a popular topic in engineering, and natural sciences to model dynamic systems based on measurement data [18], [19]. Due to high flexibility in terms of handling different types of data, ease of hyperparameter tuning, and purely data-driven nature (no need for model assumptions), Gaussian process (GP) regression has gained increasing interest to quantify model uncertainties, especially in motion planning, in combination with MPC framework [20]–[22].

In this paper, we focus on the motion control of a transport drone. Since the crowded airspace may include other drones that are uncooperative or less environmentally aware in the future, appropriate motion planning in the presence of moving obstacles is crucial to prevent accidents. Due to the power of MPC in handling nonlinear systems, constraints, and adaptability to unforeseen events, we propose a modular motion planning scheme centralized around MPC. This includes the prediction of the motion of an obstacle with measurable positions, and the shaping of optimization constraints to integrate the future behavior of the detected obstacle to the predictive planner. MPC relies on mathematical models of the system, which are often easier to develop and validate in scenarios where prior knowledge of the system dynamics is available. The key contributions of the paper are listed below:

- An MPC scheme that can also be easily adaptable to fixed-wing VTOL drones was implemented to generate high-level control commands.
- Gaussian process-based observations are integrated into the MPC scheme as inequality constraints in a recursive way. GP can deal with delayed position measurements and still provide a reasonable confidence interval, especially if it is used in a recursive fashion.
- Modular design in terms of replaceable individual modules, e.g., type of GP, flight controller, obstacle detection/localization approach.
- The proposed scheme is demonstrated in a realistic, physics-based simulation environment together with an open-source flight controller.

The paper is structured as follows: Section II briefly describes essential preliminaries of the proposed framework. In Section III, we explain the MPC scheme along with GP-based shaping of the optimization constraints to perform avoidance maneuvers. In Section IV, the application and evaluation of the concept on a UAV are shown in realistic simulation scenarios using the real-world communication framework (ROS2). Furthermore, we discus the use Kalman filter as potential alternative for our approach. Finally, Section V concludes the paper and provides future research directions.

II. PRELIMINARIES

This section briefly introduces the basics of MPC and the frameworks used for our guidance scheme implementation.

A. Model Predictive Control

As the name implies, MPC requires a dynamic model of the controlled system on an appropriate abstraction level. The basic idea of MPC is to handle the multivariable, constrained control problem by considering a specified finite time horizon to predict the system's future behavior. In this way, instead of calculating the whole set of controls from the current state to the final state, the optimization problem is repeatedly solved over a finite horizon during the operation.

Along a specified time horizon, a previously defined cost function is minimized under the consideration of the given state and input constraints in each time instant. Subsequently, only a particular part of the calculated optimal input is applied to the plant.

State, control variables, and reference state over the prediction horizon $N \in \mathbb{N}$ are usually the essential elements of the cost function for the optimization step of MPC. Constant factors weight the variable terms in the cost function to define the priority of certain state variables within the optimization problem. The optimization problem in MPC is formulated as follows:

$$\min_{\mathbf{u}} \sum_{k=0}^{N-1} \ell(\mathbf{x}(k), \mathbf{u}(k)) + V(\mathbf{x}(N)) \tag{1}$$

subject to

$$\mathbf{x}(k+1) = f(\mathbf{u}(k), \mathbf{x}(k)), \quad \forall k \in \{0, N-1\} \tag{2}$$

$$\mathbf{g}(\mathbf{x}(k)) = \mathbf{0}, \tag{3}$$

$$\mathbf{h}(\mathbf{x}(k), \mathbf{u}) \leq 0, \quad \forall k \in \{0, N-1\}, \tag{4}$$

with a stage cost $(\sum_{k=0}^{N-1} \ell(\mathbf{x}(k), \mathbf{u}(k)))$. The so-called Mayer term V represents terminal cost acting only at the end of a prediction horizon. In addition, $\mathbf{u}$ is the control input of the system (2) with the state $\mathbf{x}$. The initial condition is described by $\mathbf{x}_0$ at time t_0. The end conditions $\mathbf{g}(\cdot)$ usually describe a target state $\mathbf{x}_{\text{ref}}$ to be reached at the end of the prediction horizon, e.g., $\mathbf{g}(\mathbf{x}(N_C)) = \mathbf{x}(N_C) - \mathbf{x}_{\text{ref}}(N_C)$. Eqn. (4) denotes inequality constraints, such as the restriction of a control or state variable.

The optimal control problem (OCP) can be solved either analytically or numerically by translating it into nonlinear programming (NLP) problem. For the numerical solution, there are different external solvers. In this work, we used CasADI [23] with IPOPT solver to iteratively solve the OCP in the MPC. Within this framework, symbolic expressions for the various equations (1)-(4) are defined, and their derivatives are evaluated by using algorithmic differentiation.

B. Gaussian Process-based Prediction

Gaussian Processes are stochastic processes that are completely specified by their mean and covariance functions. They are frequently applied for non-parametric function approximation. Based on a finite set of n observations $\{(x_1, y_1), \dots (x_n, y_n)\}$, GP makes it possible to formulate a non-linear function $y = f(x) : \mathbb{R}^d \mapsto \mathbb{R}$. In a GP, the mean function defines the expected value of the process at any point, and the covariance function defines the similarity between any two points. The covariance function, also known as the kernel, encodes the assumptions about the smoothness and periodicity of the modeled function. For fundamental information, the reader is referred to [24].

GPs can be used for non-parametric Bayesian inference, where the number of parameters is not fixed and is determined by the data. This makes GPs particularly well-suited for systems where the underlying function is not or only partially known in advance.

One of the main advantages of GPs is that they quantify the uncertainty in the predictions made by the model. This is done by computing the posterior distribution of the function values at any given point, given the observed data. This allows the model to make probabilistic predictions, which can be useful in many control and decision-making tasks.With the GP-based predictions about confidence levels, trajectory corridors can be generated to be used as boundary conditions for the MPC.

III. Methodology

In this problem set, we consider UAVs that can localize themselves in the 3D space using a GPS receiver and measure their velocities using an IMU.

A. Motion Planner

The motion of a wide range of vehicle classes, such as quadrotors, holonomic ground vehicles, or robots, can be simply modeled by using double-integrator dynamics. Since we use an external flight controller that uses acceleration setpoints, for the high-level motion planner, we consider the UAV in quadcopter mode to be a simplified point mass. The state of the UAV is defined by $(\mathbf{p}, \mathbf{v}) \in \mathbb{R}^3 \times \mathbb{R}^3$, which represent the position and velocity of the UAV in 3-dimensional space, respectively. The dynamics for the time-discrete implementation are given as follows:

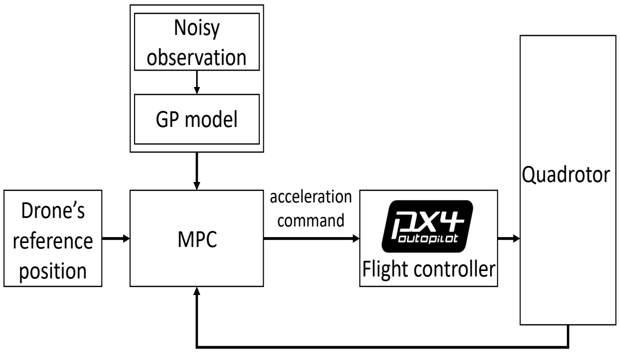

Fig. 1: System architecture of the proposed motion control concept.

$$\begin{aligned} \mathbf{p}(t_k + 1) &= \mathbf{p}(t_k) + \mathbf{v}(t_k) \cdot \Delta t, \\ \mathbf{v}(t_k + 1) &= \mathbf{v}(t_k) + \mathbf{u}(t_k) \cdot \Delta t, \end{aligned} \tag{5}$$

where Δt is a finite step size, t_k is the k-th time step and $\mathbf{u} \in \mathbb{R}^3$ is the high-level control input.

In order to generate predictive motion, we design the MPC, which makes use of a point mass model (5) and minimizes a simple cost function:

$$\min_{\mathbf{p}_{0:N-1}, \mathbf{u}_{0:N-1}} J(\mathbf{p}, \mathbf{u}) = \sum_{k=0}^{N-1} \Big(||\mathbf{p}(k) - \mathbf{p}_r||^2_{\mathbf{Q}_1} + ||\mathbf{v}(k) - \mathbf{v}_r||^2_{\mathbf{Q}_2} + ||\mathbf{u}(k)||^2_{\mathbf{R}} \Big) \tag{6}$$

$$\begin{aligned} \text{subject to } \mathbf{p}(k+1) &= \mathbf{p}(k) + \mathbf{v} \cdot \Delta t, \\ \mathbf{v}(k+1) &= \mathbf{v}(k) + \mathbf{u} \cdot \Delta t, \quad \forall k \in [0, N-1] \end{aligned} \tag{7}$$

where $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{R}$ are weighting-matrices[1] of the appropriate dimensions and can be seen as design parameters, which can be defined empirically and based on the application. The notation $\mathbf{x}_{0:N-1} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}]^T$ represents sequences. The index k denotes the current time instant. The initial states are defined as follows:

$$\mathbf{p}_0 = \mathbf{p}(k = 0), \quad \mathbf{v}_0 = \mathbf{v}(k = 0).$$

The first term of (6) considers the minimization of the distance between the UAV and the given reference position $\mathbf{p}_r$. The second term is to minimize the velocity difference between the UAV and reference velocity $\mathbf{v}_r$. The third term aims at minimum control effort. Given position setpoints, we can generate high-level control inputs, the acceleration commands, by repeatedly solving eqn. (6) using CasADi. Fig. 1 depicts the system's overall architecture. To convert the high-level control commands to thrust setpoints, we used an open-source flight controller, PX4.

[1] $||\mathbf{x}(k)||^2_{\mathbf{M}}$ is a short notation for $\mathbf{x}(k)^{\mathsf{T}}\mathbf{M}\,\mathbf{x}(k)$.

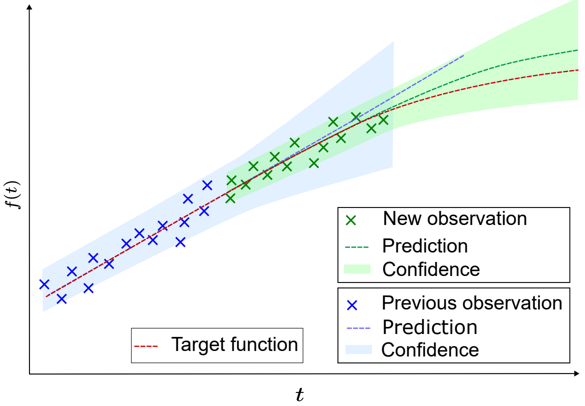

Fig. 2: Illustration of the recursive observation and GP regression scheme.

B. Constraint Shaping with GP

In many MPC implementations, static obstacles are usually integrated into the optimization scheme using safety distance r and obstacle position $\mathbf{O} \in \mathbb{R}^3$. To avoid collision between the UAV and the static obstacles, the following inequality constraint can easily be used:

$$h(\mathbf{p}, \mathbf{u}, k) = -||\mathbf{p}(k) - \mathbf{O}|| + r + d_O \leq 0, \quad d_O > 0 \tag{8}$$

where d_O denotes the predefined, additional safety distance in case of uncertainty in position measurement.

For dynamic obstacles, however, inequality (8) should be updated repeatedly over the horizon in a predictive manner. To this end, we recursively apply a GP regression to a set of recently collected observation data measured via localization with image sensors or radar. In the recursive observation scheme, we forecast the short-term future behavior of the detected obstacle based on a specific set of *previous* observations (see Fig. 2). Before each optimization step of the MPC, the GP regression is conducted based on a predefined number of recently collected observations.

The observation data do not only consist of the 3D position observations but also the corresponding timestamp of each measurement point, e.g., $[t, x(t), y(t), z(t)]$. After the application of the GP regression to the data, we obtain a model that can represent the 3D spatio-temporal behavior of the obstacle. With this, the new inequality constraint is formulated in a time-variant way, similar to ellipsoid equations, as follows:

$$\begin{aligned} h(\mathbf{p}, \mathbf{u}, k) = &- \frac{\Big(p_x(k) - C_x(k)\Big)^2}{(a(k) + d_O)^2} - \frac{\Big(p_y(k) - C_y(k)\Big)^2}{(b(k) + d_O)^2} \\ &- \frac{\Big(p_z(k) - C_z(k)\Big)^2}{(c(k) + d_O)^2} + 1 \leq 0, \quad d_O > 0 \end{aligned} \tag{9}$$

where $\mathbf{p}(k) = (p_x, p_y, p_z)$ are the coordinates of any point on the ellipsoid and $\mathbf{C}(k) = (C_x, C_y, C_z)$ are the coordinates of the mean value of the GP model at the discretization point k. The paramaters a, b, and c are the semi-axes lengths along the x, y, and z-axes, respectively. Semi-axes lengths are defined based on the confidence interval of the GP model in each dimension at the corresponding discretization point. The constant parameter d_O denotes the predefined safety distance to expand the ellipsoidal constraints to provide additional safety. In this way, the MPC constraints are shaped actively in each optimization step (6) over the prediction horizon (see Fig. 3).

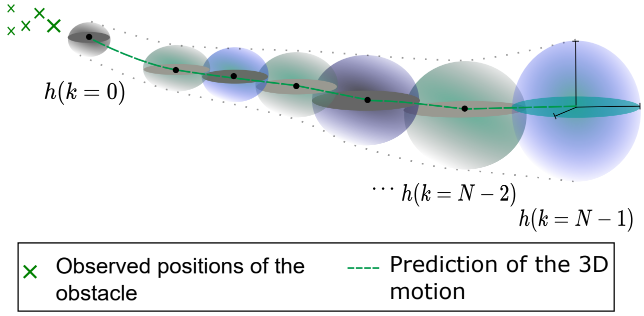
$h(k=0)$
$\cdots h(k=N-2)$
$h(k=N-1)$
Observed positions of the obstacle
Prediction of the 3D motion

Fig. 3: Visualization of the GP-based constraint shaping.

IV. RESULTS

This section presents simulation results of the proposed dynamic obstacle avoidance scheme using a communication framework based on ROS2 and a physics engine[2] (Fig. 4).

A. Simulation Experiments

For this study, we used a physics model of a quadrotor with parameters taken from AirSim [25], and the dynamics modeling is based on [26]. The UAV is equipped with a GPS receiver to determine the location of the UAV and IMU for the velocities. In addition, a sensor is required to localize the obstacle in the proximity and collect observation data. Since object tracking and localization is not within the scope of this work, this part was emulated by publishing noisy position messages with a standard deviation of 0.3 in ROS2 with slight time delays ranging from a few milliseconds up to $20\,\text{ms}$. All simulations are conducted on a computer with Intel i9-11950H CPU @ 2.6 GHz. One optimization step of the MPC with our parameter and problem setting takes $0.15\,\text{s}$ on average. Training of a GP model on 50 data points, that include 3D position information and observation timestamps, takes $\approx 88.9\,\text{ms}$ in average.

The open-source flight controller PX4 is used in Offboard mode for our experiments. PX4 Offboard mode is a control mode that is used in drone flight control systems based on the PX4 Autopilot software. In this mode, control commands are sent from an external control source, in our case, from the motion planner to the drone's flight controller over a communication link. The flight controller then executes these commands to control the drone's movement. This allows more advanced and flexible motion control of the drone in contrast to providing only position setpoints. The yaw angle of the UAV is controlled by the PX4 autopilot, which adjusts

[2] Simulations are done using OCTANE, a simulation platform developed by Fraunhofer IOSB. https://www.octane.org.

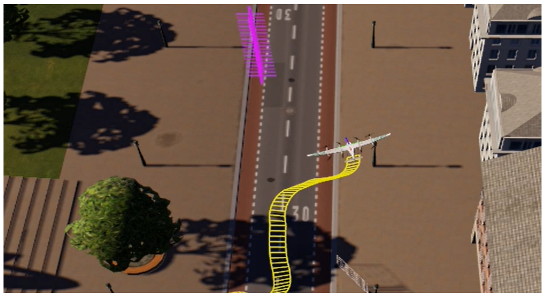

Fig. 4: A scene from the simulation, where the drone is flying in multicopter mode.

the drone's orientation to align with the targeted reference position.

For the training of the GP model, we utilized the radial basis function (RBF) kernel initialized with a length scale of 130 and white kernel initialized with a noise level of 0.3 are utilized, where the white kernel is employed to estimate the noise level within the data, while the RBF kernel is employed to capture the non-linear relationship between the data and the target variable. As optimizer, L-BFGS-B from scipy was used and the kernel hyperparameters are evaluated with the log-marginal likelihood.

The values of the parameters for the motion planner were empirically determined and are given in Table II[3]. All simulations were performed with these parameter values using the cost function from eqn. (6) and in a working area with identical dimensions. To prevent significant deviation from the straight line connecting two target positions, we introduced additional inequality constraints to ensure that the trajectory remains within an ellipsoidal space between two consecutive reference positions, following an approach similar to the first step of the Safe Flight Corridor Construction outlined in [27].

To evaluate the effectiveness of the proposed approach, we identified several test scenarios in which the obstacle's speed and direction of motion are varied. In a subset of test scenarios, the obstacle executes 3D linear motion with different constant speed values $(2, 3, 4\,m/s)$ beginning from different altitudes $(28, 34\,m)$. We set the obstacle's motion so that its path intersects with the preplanned trajectory of the UAV at potential collision point $(34, 30, 28)$, marked with a red cross in Fig. 5. We adjusted the initial position of the obstacle in each test scenario, exploring a range of collision directions around the intersection (collision) point. In addition, we also investigated obstacles with a circular motion applying different angular velocities $(0.05 - 0.15\,rad/s)$ and intersecting the flight trajectory of the UAV.

Fig. 5 depicts the UAV's reference positions and paths and moving obstacle from a selected test scenario. The UAV takes off from position $(0, 0, 0)$ and flies first to the closest reference position. Fig. 6 shows the predicted position of the obstacle at each MPC iteration from the same test scenario. The obstacle was detected first at ca. $29\,s$, and after collecting observation data for ca. 3 seconds, the forecasting begins. When more than 50 data points are available, the prediction is made based on the 50 most recent collected observations. Fig. 7 visualizes the mean values over the prediction horizon in each GP model prediction. The prediction obviously tends to better represent the path as more observations are utilized for the GP regression. Since the ground truth and the confidence of the prediction are partially, relatively close to each other, the additional distance d_O provides extra safety and also compensates faulty forecasting.

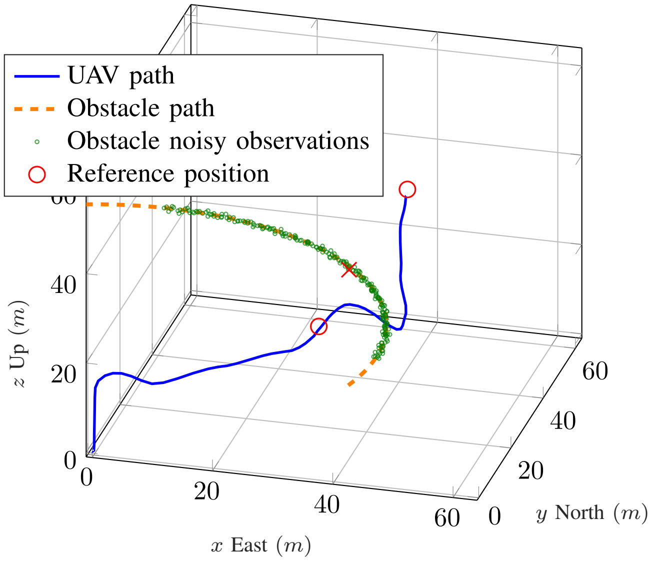

Fig. 5: Paths of UAV and the dynamic obstacle from an example test scenario with the take-off position $(0, 0, 0)$. The obstacle flies from the North towards the South-East.

B. Discussion

Since the multi-step position prediction over the horizon is an important component of the proposed scheme, we compared the extended Kalman filter (EKF) as an alternative method to GP regression. Since the Kalman filters are usually applied for one-step state estimation, we modified the EKF accordingly to enable multi-step estimation that can be used in the MPC framework. In alternative prediction approaches, we consider multi-step EKF based on a linear motion model and constant turn-rate (CTR) model with a turn rate of 0.05 rad/s. Table I shows a comparison of them on the same dataset with 150 position data points based on metrics such as average multi-step prediction time and root mean square error (RMSE). Although Kalman filter is time-efficient, it is based on an assumed dynamic model and designed for one-step prediction, so in multi-step estimations, the error propagates. On the other hand, the implemented GP regression provides significantly better estimation since it is not based on a model assumption.

The minimum distance values between the UAV and the moving obstacle during all simulated scenarios for different

[3] Our main goal is to demonstrate the approach's performance across diverse scenarios, with an emphasis on generalization rather than cost function optimization.

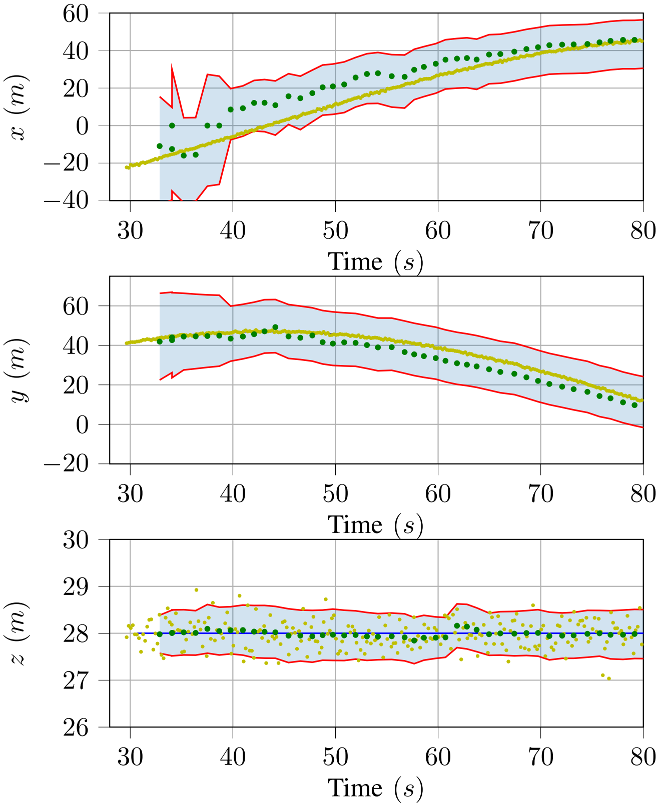

Fig. 6: Position forecasts (green points) with corresponding confidence intervals (light blue-colored region) at each MPC iteration from the selected test scenario. Light green points and the blue line represent the noisy observations and the ground truth, respectively.

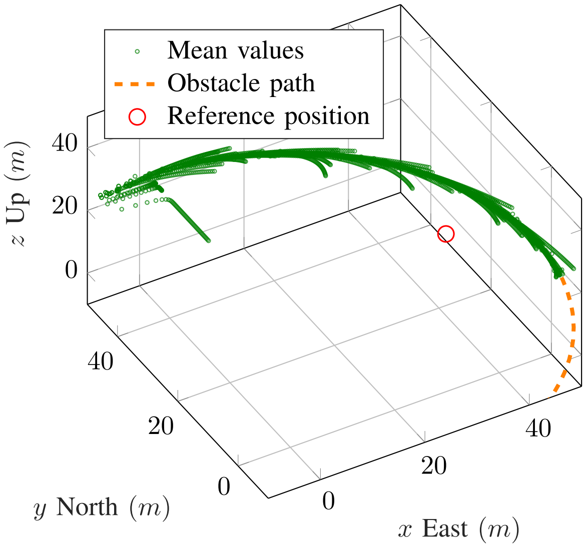

Fig. 7: Forecasting behavior over the individual prediction horizon in each MPC step from the selected test scenario (see Fig. 5).

	EKF (lin)	EKF (ctr)	GPR
RMSE	3.11	1.32	1.12
avg. time (ms)	2.8	3.4	68.1

TABLE I: Comparison of multi-step (50 time steps with $0.2\,\mathrm{s}$ step size) prediction performances of KF and GP-based estimation. The real obstacle motion was circular. It should be noted that the value of the constant turn-rate strongly influences the multi-step prediction accuracy.

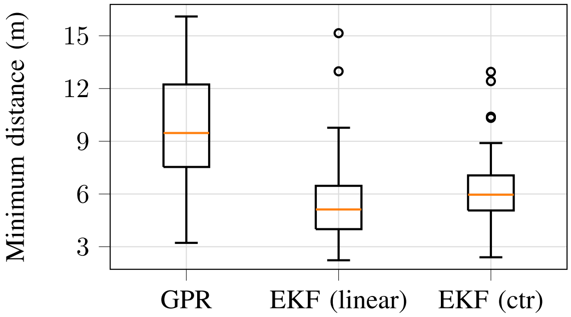

Fig. 8: Analysis of the minimum distances in all test scenarios.

multi-step prediction approaches are depicted in a boxplot in Fig. 8. The simulated test scenarios covered 48 linear and 12 circular obstacle motions with different characteristics. Based on the results, the UAV applying the GP-based method could maintain a reasonable distance from the obstacle and in more than $75\,\%$ of the test scenarios, it successfully obeyed the defined minimum distance of $6\,m$. On the other hand, however, in $75\,\%$ of the tests with EKF, the UAV could maintain a minimum distance below $6.7\,m$. In EKF basd on CTR model, $50\,\%$ of the cases, and in EKF using a linear motion model, more than $50\,\%$ of the cases, the minimum distance was below $6\,m$. The outliers, where the drone maintained a high distance, correspond to cases in which the obstacle approaches from a similar direction to that of the drone, rather than from the opposite direction. Due to the high covariance values from EKF used similarly to those in eqn. (9), the drone may have reacted at an appropriate time.

The simple GP-based forecast works with limitations and additional safety distance sufficiently good when the hyperparameters of the GP regression are adequately tuned by considering the measurement noise, observation frequency, and the number of data points used for the GP regression model fitting. Compared to the modified EKF-based attempts, the proposed scheme helps predict spatio-temporal behavior of the moving obstacle in a more generalized fashion with a good confidence. However, it is recommended for moderately dense areas to ensure safe avoidance maneuvers.

V. CONCLUSION

This paper proposes a modular and efficient approach for collision avoidance with dynamic obstacles. While utilizing the obstacle's spatio-temporal observations and data-driven motion forecast via GP, MPC constraints can be shaped

TABLE II: Parameter Setting

Cost function	$\mathbf{Q}_1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
	$\mathbf{Q}_2$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
	$\mathbf{R}$	$[1,1,1]^T$
MPC parameters	Prediction horizon N	75
	Step size Δt	$0.2\,s$
	Velocity range	$[-5,5]\ m/s$
	Acceleration range	$[-3,3]\ m/s^2$
	Additional safety distance d_O	$6\ m$

during motion planning. Thus, moving obstacles can be avoided in a predictive way by maintaining the intended flight path. The proposed scheme is tested and evaluated in realistic physics simulations. Future work will incorporate the avoidance of multiple obstacles and additional environmental constraints by making use of environmental perception. Furthermore, employing heuristics such as choosing them from a subset of the initial data points can be investigated to enhance the computational efficiency of the GP module. We also plan to investigate the UAV's reactivity in detail, focusing on improving its response times and maneuverability.

APPENDIX

In Fig. 9, the acceleration commands (red) generated by our motion control scheme and the IMU-based acceleration (blue) are shown. PX4 sufficiently tracks the high-level control inputs. However, the tracking behavior can be improved by adequately tuning the controller gains.

Fig. 10 visualizes the mean values over the prediction horizon in each EKF-based prediction using CTR. In Fig. 11, the prediction behavior is illustrated for the EKF based on liner motion. In addition, the test cases with linear and circular obstacle motion are analysed separately in Fig. 12. Based on the analyses, it can be concluded that GP-based prediction can outperform Kalman filter-based multi-step prediction approaches, as errors strongly propagate when the Kalman filter's update step relies solely on new observation data. The EKF can partially compensate for this by incorporating variance values from the covariance matrices in the MPC constraints. Furthermore, EKF with CTR tends to provide better results in our scenario. However, the predefined constant turn-rate value remains a limiting factor in terms of multi-step prediction accuracy.

ACKNOWLEDGMENT

This work was supported by the LuFo VI-3 program of the Federal Ministry for Economic Affairs and Climate Action as part of the NiKoLaS project under grant agreement number 20F2203A. The authors would like to express their gratitude to Andreas Greiner for providing technical support, and Harisankar Babu for setting up the simulation environment.

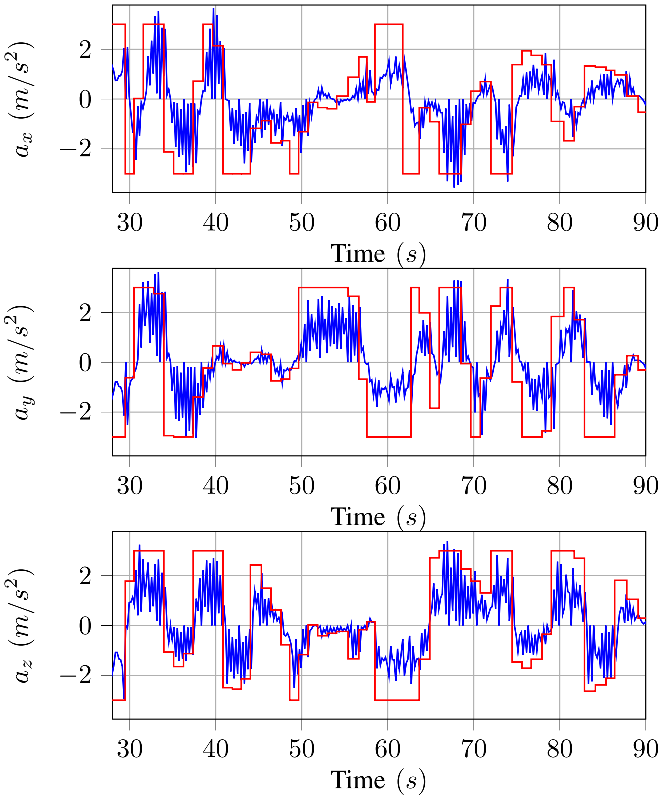

Fig. 9: High-level control inputs (red) and the IMU-based acceleration values (blue) of the UAV from the selected test scenario (see Fig. 5).

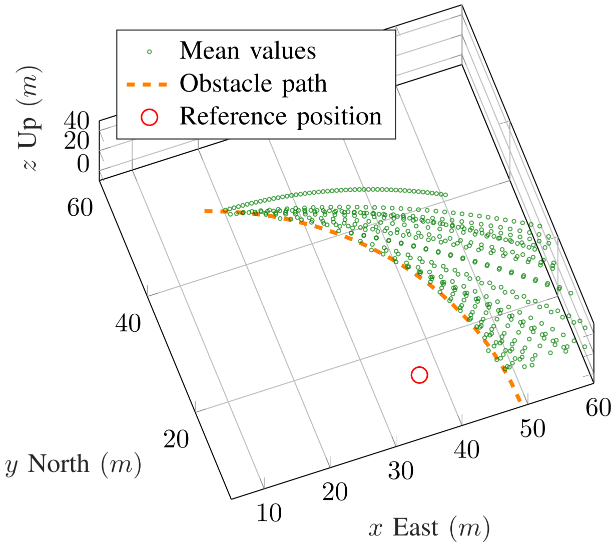

Fig. 10: Forecasting behavior of the EKF using CTR over the individual prediction horizon in each MPC step from the selected test scenario (see Fig. 5).

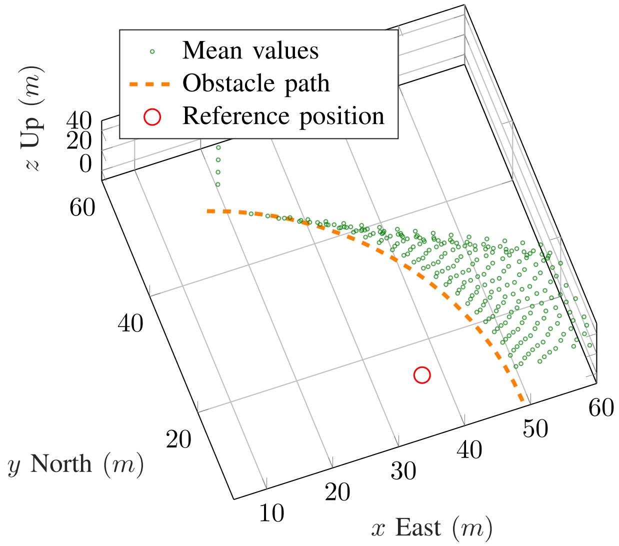

Fig. 11: Forecasting behavior of the EKF based on linear motion over the individual prediction horizon in each MPC step from the selected test scenario (see Fig. 5).

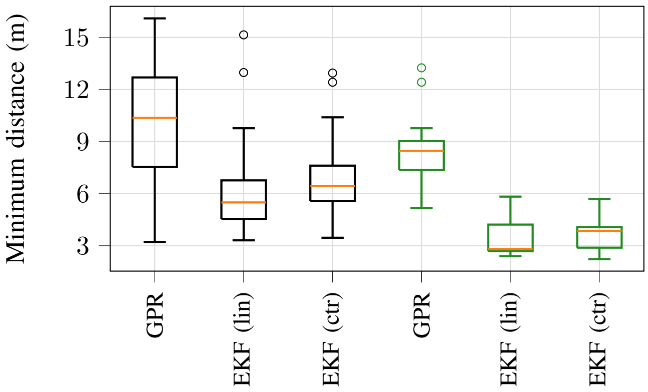

Fig. 12: Analysis of minimum distances for different obstacle motion types: black boxes for linear and green for circular trajectories.

References

[1] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, and L. A. Saidane, "Monitoring road traffic with a uav-based system," in *2018 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2018, pp. 1–6.

[2] A. Albanese, V. Sciancalepore, and X. Costa-Pérez, "Sardo: An automated search-and-rescue drone-based solution for victims localization," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3312–3325, 2021.

[3] M. Moshref-Javadi and M. Winkenbach, "Applications and research avenues for drone-based models in logistics: A classification and review," *Expert Systems with Applications*, vol. 177, p. 114854, 2021.

[4] A. Ma'Arif, W. Rahmaniar, M. A. M. Vera, A. A. Nuryono, R. Majdoubi, and A. Çakan, "Artificial potential field algorithm for obstacle avoidance in uav quadrotor for dynamic environment," in *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*. IEEE, 2021, pp. 184–189.

[5] L. Keyu, L. Yonggen, and Z. Yanchi, "Dynamic obstacle avoidance path planning of uav based on improved apf," in *2020 5th International Conference on Communication, Image and Signal Processing (CCISP)*. IEEE, 2020, pp. 159–163.

[6] T. Jia, S. Han, P. Wang, W. Zhang, and Y. Chang, "Dynamic obstacle avoidance path planning for uav," in *2020 3rd International Conference on Unmanned Systems (ICUS)*. IEEE, 2020, pp. 814–818.

[7] H. Chen, P. Lu, and C. Xiao, "Dynamic obstacle avoidance for uavs using a fast trajectory planning approach," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019.

[8] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "Acas sxu: Robust decentralized detect and avoid for small unmanned aircraft systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–9.

[9] F. Mumuni, A. Mumuni, and C. K. Amuzuvi, "Deep learning of monocular depth, optical flow and ego-motion with geometric guidance for uav navigation in dynamic environments," *Machine Learning with Applications*, vol. 10, p. 100416, 2022.

[10] J. Choi, G. Lee, and C. Lee, "Reinforcement learning-based dynamic obstacle avoidance and integration of path planning," *Intelligent Service Robotics*, vol. 14, pp. 663–677, 2021.

[11] X. Han, J. Wang, J. Xue, and Q. Zhang, "Intelligent decision-making for 3-dimensional dynamic obstacle avoidance of uav based on deep reinforcement learning," in *2019 11th International conference on wireless communications and signal processing (WCSP)*. IEEE, 2019.

[12] J. Tordesillas and J. P. How, "Deep-panther: Learning-based perception-aware trajectory planner in dynamic environments," *IEEE Robotics and Automation Letters*, 2023.

[13] H. Ahn, J. Park, H. Bang, and Y. Kim, "Model predictive control-based multirotor three-dimensional motion planning with point cloud obstacle," *Journal of Aerospace Information Systems*, vol. 19, no. 3, pp. 179–193, 2022.

[14] E. Olcay and A. Azizoglu, "Coordination of a semi-informed flocking system via model predictive control," *1st IFAC Workshop on Control of Complex Systems*, 2022.

[15] E. Olcay, K. Gabrich, and B. Lohmann, "Optimal control of a swarming multi-agent system through guidance of a leader-agent," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 1–6, 2019.

[16] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE robotics and automation letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[17] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2682–2688.

[18] P. M. Van den Hof, "System identification-data-driven modelling of dynamic systems," *Eindhoven University of Technology*, 2012.

[19] E. Olcay, C. Dengler, and B. Lohmann, "Data-driven system identification of an innovation community model," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1269–1274, 2018.

[20] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[21] Y. Liu, P. Wang, and R. Tóth, "Learning for predictive control: A dual gaussian process approach," *arXiv preprint arXiv:2211.03699*, 2022.

[22] N. Schmid, J. Gruner, H. S. Abbas, and P. Rostalski, "A real-time gp based mpc for quadcopters with unknown disturbances," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2051–2056.

[23] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[24] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[25] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[26] A. S. Sanca, P. J. Alsina, and F. C. Jés de Jesus, "Dynamic modelling of a quadrotor aerial vehicle with nonlinear inputs," in *2008 IEEE Latin American Robotic Symposium*. IEEE, 2008, pp. 143–148.

[27] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.