Model Predictive Control for Aerial Collision Avoidance in dynamic environments

Manuel Castillo-Lopez, Seyed Amin Sajadi-Alamdari, Jose Luis Sanchez-Lopez,
Miguel A. Olivares-Mendez, Holger Voos[1]

***Abstract*— Autonomous navigation in unknown environments populated by humans and other robots is one of the main challenges when working with mobile robots. In this paper, we present a new approach to dynamic collision avoidance for multi-rotor unmanned aerial vehicles (UAVs). A new nonlinear model predictive control (NMPC) approach is proposed to safely navigate in a workspace populated by static and/or moving obstacles. The uniqueness of our approach lies in its ability to anticipate the dynamics of multiple obstacles, avoiding them in real-time. Exploiting active set algorithms, only the obstacles that affect to the UAV during the prediction horizon are considered at each sample time. We also improve the fluency of avoidance maneuvers by reformulating the obstacles as orientable ellipsoids, being less prone to local minima and allowing the definition of a preferred avoidance direction. Finally, we present two real-time implementations based on simulation. The former demonstrates that our approach outperforms its analog static formulation in terms of safety and efficiency. The latter shows its capability to avoid multiple dynamic obstacles.**

I. Introduction

Unmanned aerial vehicles (UAVs), commonly known as drones, are now populating natural and industrial environments due to their multiple applications such as package delivery, traffic-surveillance or search-and-rescue operations [1]. These applications often demand the use of mobile robots in presence of humans and other robots, making autonomous navigation one of the most active topics in robotics research [2]. This problem is usually addressed by a hierarchical combination of planning and control but, the lack of proper coordination between these agents, often leads to inefficient or dangerous situations [3]. In this paper, we use nonlinear model predictive control (MPC) to integrate motion planning and control in scenarios with multiple dynamic obstacles as shown in Fig. 1.

Most prior work on moving obstacles are based on the concept of velocity obstacles (VOs) [4] to compute the set of velocities of the robot that will cause a collision. This technique has been used and enhanced by many authors, taking into account nonlinear dynamics [5], uncertainty [6], reciprocal behavior [7] or its use for computing control objectives in convex optimization [8].

Unlike these methods, nonlinear model predictive control (NMPC) is able to natively consider non-convex constraints such as obstacles to generate feasible trajectories, being safer and less prone to local minima than other hierarchical approaches [3]. Even though NMPC is known to be computationally expensive, recent advances on non-linear solvers [9], [10] has triggered its use in fast real-time applications such as collision avoidance for small UAVs.

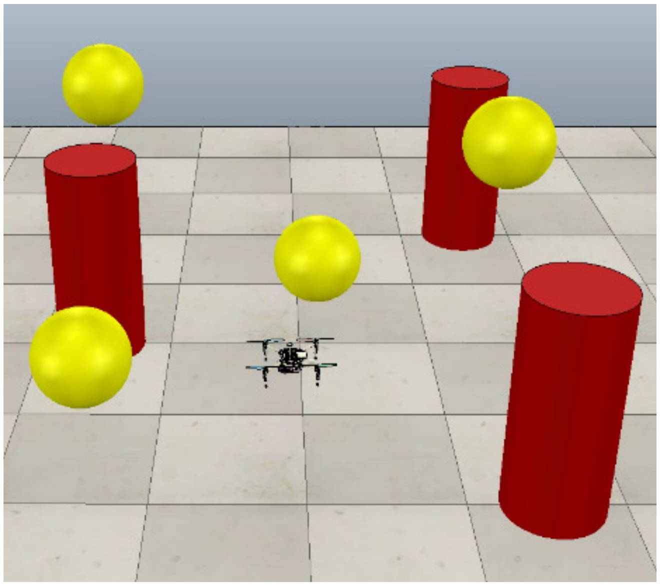


Fig. 1: UAV crossing the street avoiding multiple dynamic obstacles. See video in https://rebrand.ly/castillo2018mpc

Although great effort has been done in this area, dealing with multiple three-dimensional obstacles remains difficult. Recent work using NMPC often simplifies this problem by considering only the closest obstacle [11] or reducing the obstacles as bi-dimensional static [12], [13] or dynamic [14] constraints.

To provide safe navigation in cluttered dynamic scenarios we exploit active set algorithms to consider only the constraints that affect to our problem at each sample time. In this paper, we extend our previous work [15] by including the dynamics of ellipsoidal obstacles without additional cost. We use parametrized soft constraints to specify the sensitiveness of the avoidance maneuvers with the guarantee of finding a locally-optimal solution even in highly constrained scenarios as shown in Fig. 1. Successful experiments in real-time validates our approach in scenarios with multiple dynamic obstacles, outperforming its analog formulation in terms of safety and efficiency.

[1] Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg {manuel.castillo, holger.voos}@uni.lu

* This work was supported by the Fonds National de la Recherche (FNR), Luxembourg, under the project C15/15/10484117 (BEST-RPAS).

This paper is organized as follows: In section II we develop the model for a multi-rotor UAV, including the experimental identification of a real platform. Section III formulates the NMPC control policy regarding trajectory tracking an collision avoidance. Finally, we validate our approach in two realistic scenarios in section IV, drawing the conclusions in section V.

II. UAV Model

A multi-rotor UAV is usually modeled using the Newton-Euler equations of a rigid body to stabilize and control the platform [16]. However, the typical configuration of an UAV includes an autopilot that controls its stability while following velocity commands given by an external pilot (automated or human). The control input $\mathbf{u} = [u_x\ u_y\ u_z\ u_\psi]^T$ is divided in forward, sidewards, upward, and heading velocity references. This commands are specified in a pitch/roll invariant body frame named as the hovering frame H (see Fig. 2). The world frame W, is defined as a standard North-East-Up fixed reference frame.

Recent work on quadrotor modeling describe it as a differentially flat system [16], meaning that its full state can be represented by a combination of its flat outputs and their derivatives. In this work we choose the cartesian coordinates of the center of mass $\mathbf{r} = [x\ y\ z]$ and the yaw angle ψ to build the reduced state vector:

$$\mathbf{x} = [x\ y\ z\ \psi\ v_x\ v_y\ v_z\ v_\psi]^T \tag{1}$$

being $\mathbf{v} = [v_x\ \ v_y\ \ v_z]$ the UAV linear velocity in the hovering frame and the heading angular velocity v_ψ. Then, we propose a nonlinear model $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ defined by the set of equations:

$$\dot{x} = v_x cos(\psi) - v_y sin(\psi) \tag{2a}$$

$$\dot{y} = v_x sin(\psi) + v_y cos(\psi) \tag{2b}$$

$$\dot{z} = v_z \tag{2c}$$

$$\dot{\psi} = v_\psi \tag{2d}$$

$$\dot{v_i} = (-v_i + k_i u_i)/\tau_i, \quad i \in \{x, y, z, \psi\} \tag{2e}$$

where (2e) models the velocity response of the UAV as a first order model of gain k_i and constant time τ_i.

In this work, we use a motion capture system[1] to perform a precise model identification of a DJI-M100[2] quadrotor (see Fig. 2). The model parameters are obtained by a classical step response tangent method [15] defined by:

$$k_i = \frac{v_i(\infty)}{u_i(\infty)} \qquad \tau_i = \frac{3}{2}(t_{63} - t_{28}) \tag{3}$$

where t_{63} and t_{28} are the times, from the start of the step, when the velocity reaches the 63% and the 28% of its final value respectively. The resulting parameters are shown in Table I.

This lightweight formulation allow us to solve the UAV dynamics faster without the knowledge of its physical design, being applicable on any multi-rotor platform. Besides, placing the controller outside the inner control loop relaxes its real-time requirements, allowing us to increase the prediction horizon of the MPC approach to target high-level control policies.

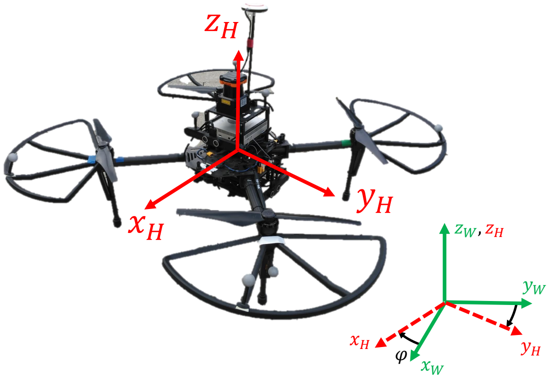


Fig. 2: Hovering frame of the DJI-M100 platform.

TABLE I: First order model parameters of the DJI-M100

	k_i	τ_i (s)
x	1.0000	0.8355
y	1.0000	0.7701
z	1.0000	0.5013
ψ	$\pi/180$	0.5142

III. Controller Design

Model predictive control obtains the control input of the system by solving an Optimal Control Problem (OCP) each sample time. That involves minimizing a given cost function over a defined prediction horizon subjected to states and input constraints. In this work, the OCP is designed to minimize the effort required for an UAV to track a desired state or trajectory while avoiding dynamic obstacles.

A. Trajectory tracking

Trajectory tracking is formulated as the cost term J^t defined in (4). The difference between the predicted state $\mathbf{x}_i$ and its reference $\mathbf{x}_i^*$ at each step i is penalized over the prediction horizon N as follows:

$$J^t = \frac{1}{2} \sum_{i=0}^{N-1} \|\mathbf{x}_i - \mathbf{x}_i^*\|_P^2 + \|\mathbf{x}_N - \mathbf{x}_N^*\|_Q^2 \tag{4}$$

where P and Q are weighting matrices of each quadratic penalty. For stability and energy efficiency, an analog cost J^c is defined for the control inputs as:

$$J^c = \frac{1}{2} \sum_{i=0}^{N-1} \|\mathbf{u}_i\|_R^2 \tag{5}$$

B. Obstacle avoidance

The use of three-dimensional constraints to model obstacles in a 3D environment is often a good practice. However, the use of hard constraints to guarantee the generation of safe trajectories leads to non-feasible problems and unpredictable results, making its viable use only with few obstacles [11].

[1] Optitrack Motion Capture System: http://optitrack.com/
[2] DJI Matrice 100: https://www.dji.com/matrice100

In this work, we propose to use parametrized soft constraints to model each ellipsoidal obstacles as follows:

$$\xi^2(\mathbf{r}_i, \mathbf{r}_i^o) + \theta^\xi s_i^\xi \geq 1 \tag{6}$$

where s_i^ξ is an extra control input, known as slack variable, that relaxes the constraint with sensitivity θ^ξ to guarantee feasible solutions in tight situations. The ellipsoidal term $\xi^2(\mathbf{r}_i, \mathbf{r}_i^o)$ is based on the distance function:

$$\xi(\mathbf{r}_i, \mathbf{r}_i^o) = \sqrt{(\mathbf{r}_i - \mathbf{r}_i^o)^T Q_i (\mathbf{r}_i - \mathbf{r}_i^o)} \tag{7}$$

where $\mathbf{r}_i^o$ is the center position of the ellipsoid in world frame at time step i. Q_i is the metric induced by the ellipsoid dimensions $M = diag(r_x^{-2}, r_y^{-2}, r_z^{-2})$ rotated by ${}^O R_W$ to the world frame as:

$$Q_i = {}^O R_W^T \; M_i \; {}^O R_W \tag{8}$$

This formulation generalizes the euclidean obstacle model present in the literature, allowing to move, orientate and resize the obstacles along the prediction horizon at each sample time. Thus, these parameters can be manipulated to include the dynamics of every obstacle without the need of additional variables and extra computation. In this work, we propagate the position of the obstacle at constant velocity as:

$$\mathbf{r}_i^o = \mathbf{r}_{i-1}^o + \dot{\mathbf{r}}_{i-1}^o \cdot \Delta t \tag{9}$$

Soft constraints has the main drawback of increasing the computational cost because of the slack variables. To mitigate this effect when dealing with multiple obstacles, we define a shared slack variable s_i^ξ for all the ellipsoidal obstacles, with the cost:

$$J^\xi = \frac{1}{2} \sum_{i=0}^{N-1} \|\mathbf{s}_i^\xi\|_S^2 \tag{10}$$

Boundary obstacles, such as walls or the floor, are also considered analogously as soft planar constraints based on its position $\mathbf{r}_i^o$ and normal vector $\mathbf{n}_i$ as shown in (11) and (12).

$$\pi(\mathbf{r}_i) + \theta^\pi s_i^\pi \geq 0 \tag{11}$$

$$J^\pi = \frac{1}{2} \sum_{i=0}^{N-1} \|\mathbf{s}_i^\pi\|_T^2 \tag{12}$$

where $\pi(\mathbf{r}_i) = \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{r}_i^o)$

C. Optimal Control Problem

Integrating the previous definitions, the optimal control problem is formulated in the set of equations (13), taking the form of a discrete non-linear program (NLP).

$$\begin{array}{lllr}
\underset{X,U}{\text{minimize}} & J = J^t + J^c + J^\xi + J^\pi & & (13a) \\
\text{subject to:} & \mathbf{x}_0 = \bar{x}_0 & & (13b) \\
 & \mathbf{x}_{i+1} = \Phi_i(\mathbf{x}_i, \mathbf{u}_i) & i = 0, \ldots, N-1 & (13c) \\
 & \xi(\mathbf{x}_i) + \theta^\xi s_i^\xi \geq 1 & i = 0, \ldots, N-1 & (13d) \\
 & \pi(\mathbf{x}_i) + \theta^\pi s_i^\pi \geq 0 & i = 0, \ldots, N-1 & (13e) \\
 & |\mathbf{u}_i| \leq \mathbf{u}_{max} & i = 0, \ldots, N-1 & (13f)
\end{array}$$

In (13a) all the cost terms are merged in a single objective function to find the best trade-off between trajectory tracking, efficiency and collision avoidance. In (13b) the feedback state $\bar{x}_0$ is set as the initial state in the prediction horizon. The (13c) introduces the discretized form of the UAV model $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ as an equality constraint, including obstacles and maximum control inputs in (13d)-(13e) and (13f) respectively.

IV. Experiments

The aim of our approach is to provide safe navigation for UAVs in complex dynamic environments. To validate it, we set a simple scenario in subsection IV-C to show its performance compared with its analog approach, which uses a static obstacle formulation. Then, we set a challenging scenario in section IV-D to analyze its capability to deal with multiple moving obstacles.

A. Risk evaluation

In this work, the risk of collision is evaluated by the distance to obstacle d and the inverse time to collision (TTC^{-1}) [17], which is defined as:

$$TTC^{-1} = \frac{\dot{d}}{d} \tag{14}$$

This rate indicates the risk of collision between two agents. Negative values correspond to a potential collision, while positive values indicate that the agents are moving away from each other. The safest situation is around zero, corresponding to high distances and small relative velocity.

B. Implementation details

To implement the optimal control problem defined in (13), ACADO Toolkit for C++ has been used to generate a fast explicit solver for the NMPC controller. The NMPC controller has been implemented in C++ programming language, building the communications and interfaces using the ROS Kinetic framework [18]. As a simulation environment V-REP [19] is used to run software-in-the-loop experiments.

The implementation parameters of the MPC algorithm are shown in Table II. A long prediction horizon is chosen to promote a long-term optimal control policy, which is highly sensitive to obstacles. More sensitivity is given to ellipsoidal obstacles, which are meant to be dynamic.

TABLE II: Model Predictive Controller implementation parameters.

Prediction horizon	4s
Discretization steps	20
Integrator type	Runge-Kutta 4
Maximum controls	1 m/s
Control Rate	20 Hz
Ellipsoidal obstacles sensitivity	0.15
Planar constraints sensitivity	0.25

An uniform weight distribution is chosen to provide a balanced trade-off between efficiency, safety and tracking. As shown in Table III, orientation penalty is an exception that has been set aggressively to compensate the low gain given by the model.

TABLE III: Model Predictive Controller weighting values.

	Weights
Position	10
Orientation	5000
Linear velocities	1
Angular velocities	1
Control inputs	10
Slack variables	10

C. *Street crossing scenario*

Lets consider a scenario where the goal of the UAV is to cross a street populated by humans, which must be avoided without flying over them. For that purpose, we model the pedestrians as ellipsoids with radius of $(r_x, r_y, r_z) = (0.5m, 0.6m, \infty)$. This generates an elliptical cylinder slightly slimmer in the approaching direction, which reduces the probability of reaching a local minima and promotes fluent maneuvers. The UAV is modeled as a sphere with radius of $0.5m$, applying a safety distance of $1m$. In this experiment, we consider three pedestrians moving alternatively at 1 m/s perpendicular to the robot's shortest path as shown in Fig. 3.

To validate the approach we compare the performance of the same controller updating the obstacles in two different ways: statically updated at each sample time and dynamically propagated based on the obstacle velocity. Fig. 7 shows the trajectory generated while performing the experiment with the static approach. Even though that the UAV manage to avoid all obstacles, the generated maneuvers interfere with the trajectories of pedestrians, leading to situations with greater risk and higher deviations from the optimal path. In contrast, our approach presents a safer control policy, finding optimal to avoid the obstacles without crossing their future trajectory as shown in Fig. 8.

Even though the soft constraints are designed to be violated with high sensitivity, this situation should be minimized. In Fig. 4 we can see that both implementations provide a collision-free navigation. However, our approach provides safer avoidance, reducing the soft constraint violations in time and magnitude. As shown in Fig. 5, our approach reduces the inverse time to collision (TTC^{-1}) peak values and, as a result, the magnitude of the risk. The frequency in which dangerous situations occur is shown in Fig. 6, where the histogram of the minimum TTC^{-1} indicates that our approach reduces the time in which the UAV is at risk of collision.

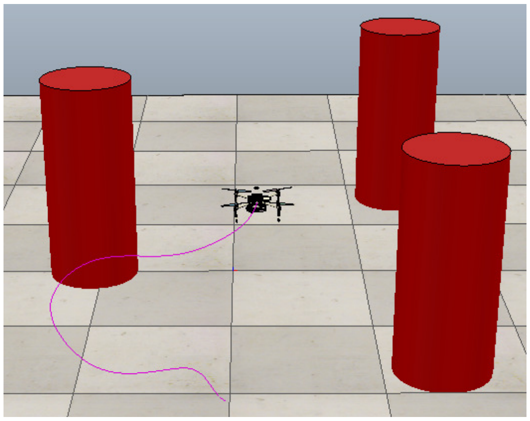
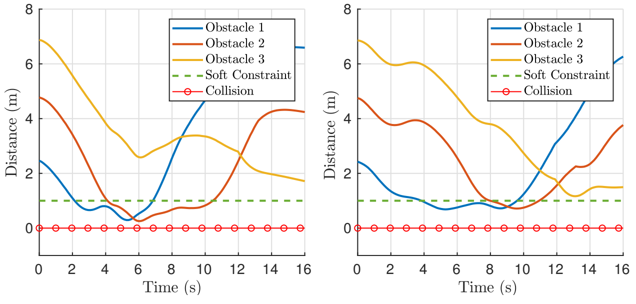
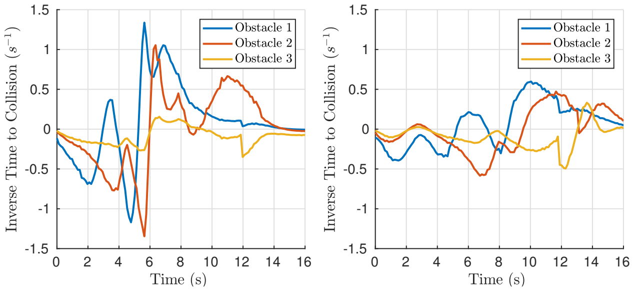


Fig. 3: UAV avoiding human-size dynamic obstacles. See video in https://rebrand.ly/castillo2018mpc



(a) Static approach (b) Dynamic approach

Fig. 4: Distance to obstacles in street crossing scenario



(a) Static approach (b) Dynamic approach

Fig. 5: TTC^{-1} in street crossing scenario

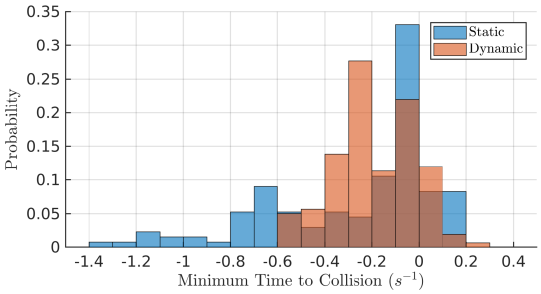


Fig. 6: Histogram of minimum TTC^{-1} in street crossing scenario

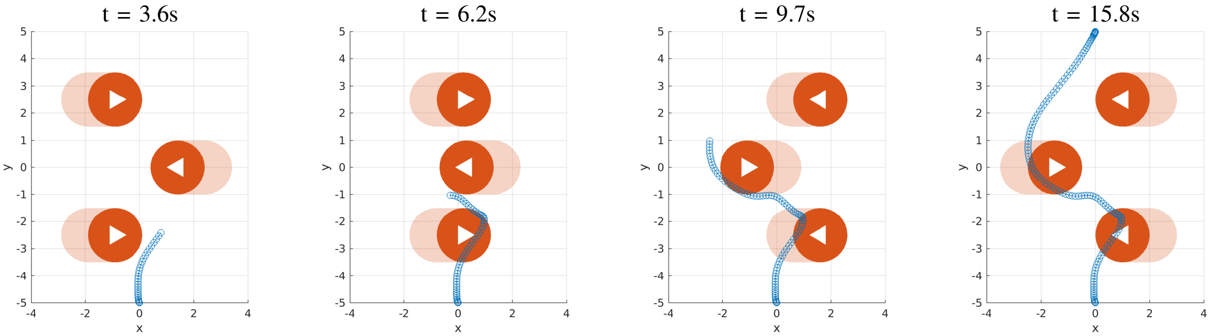


Fig. 7: UAV trajectory in street crossing scenario. Static approach.

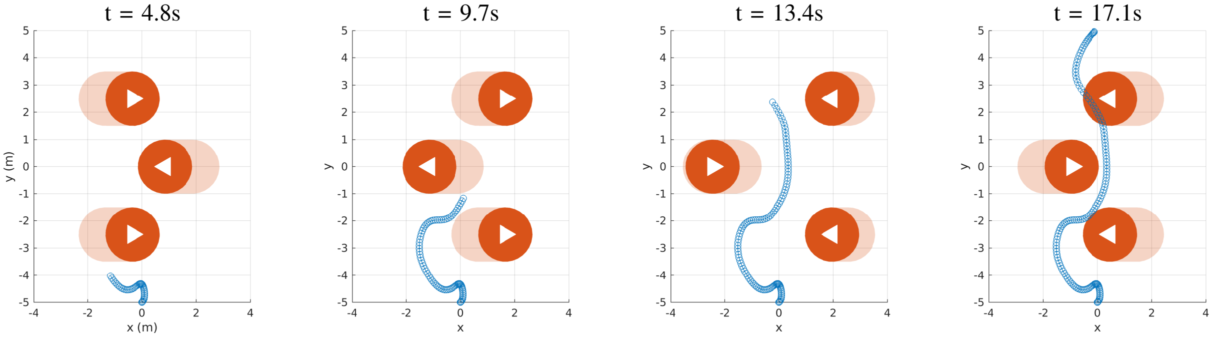
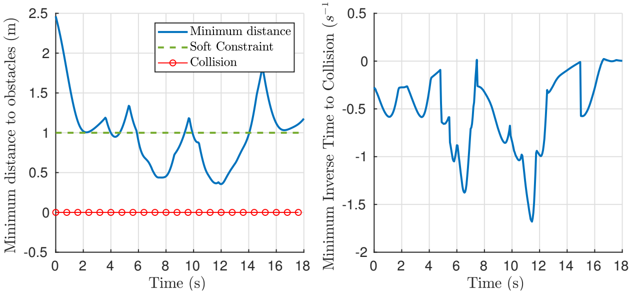


Fig. 8: UAV trajectory in street crossing scenario. Dynamic approach

D. Multiple obstacle scenario

Lets consider a new scenario where the UAV has to cross a street populated not only by humans, but also other aerial robots. In this experiment, the robot is allowed to fly over other robots, but not over pedestrians. The pedestrian model of the previous experiment is used, considering other UAVs as ellipsoids with radius of $(r_x, r_y, r_z) = (0.5m, 0.6m, 0.5m)$ and a safety distance of $1m$.

In this experiment, in addition to the pedestrians of the previous scenario, we consider four aerial robots moving alternatively at 1 m/s as shown in Fig. 1. Three of them move perpendicular to the robot's shortest path while the fourth is moving in diagonal direction. In Fig. 11 is shown that the UAV is able to avoid laterally the pedestrians while avoiding three-dimensionaly the aerial robots. Fig. 9 shows that our approach manage to avoid multiple collisions, with the increase of TTC^{-1} magnitude and soft constraint violations due to the complexity of the environment.



(a) Minimum Distance

(b) Minimum TTC^{-1}

Fig. 9: Risk variables in the multiple obstacle scenario

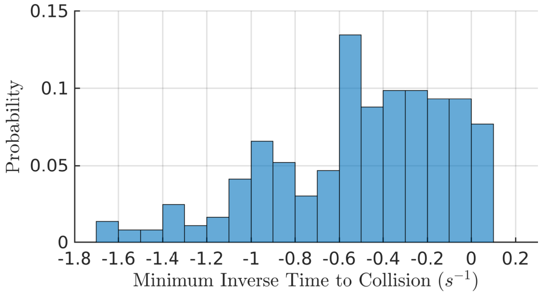


Fig. 10: Minimum inverse time to collision histogram in multiple obstacle scenario

V. Conclusions

We presented a new model predictive control approach for three-dimensional collision avoidance in scenarios with multiple dynamic obstacles. These obstacles were modeled as orientable ellipsoids by using parametrized soft constraints, which allows a flexible obstacle definition, guaranteeing feasible solutions. With this formulation, the dynamics of each obstacle can be introduced externally without additional cost, suiting the needs of each application. In this paper, we used a constant velocity model to test its collision avoidance performance in two real-time scenarios. The first experiment shows a considerable improvement over a static formulation, being safer and more efficient. The second experiment validates the approach for three-dimensional avoidance in a cluttered scenario with seven moving obstacles.

In the future, we propose to consider the implementation in a real platform, including uncertainty on the model, perception and localization systems. We also plan to increase the model accuracy and the prediction horizon length to target higher-level control policies.

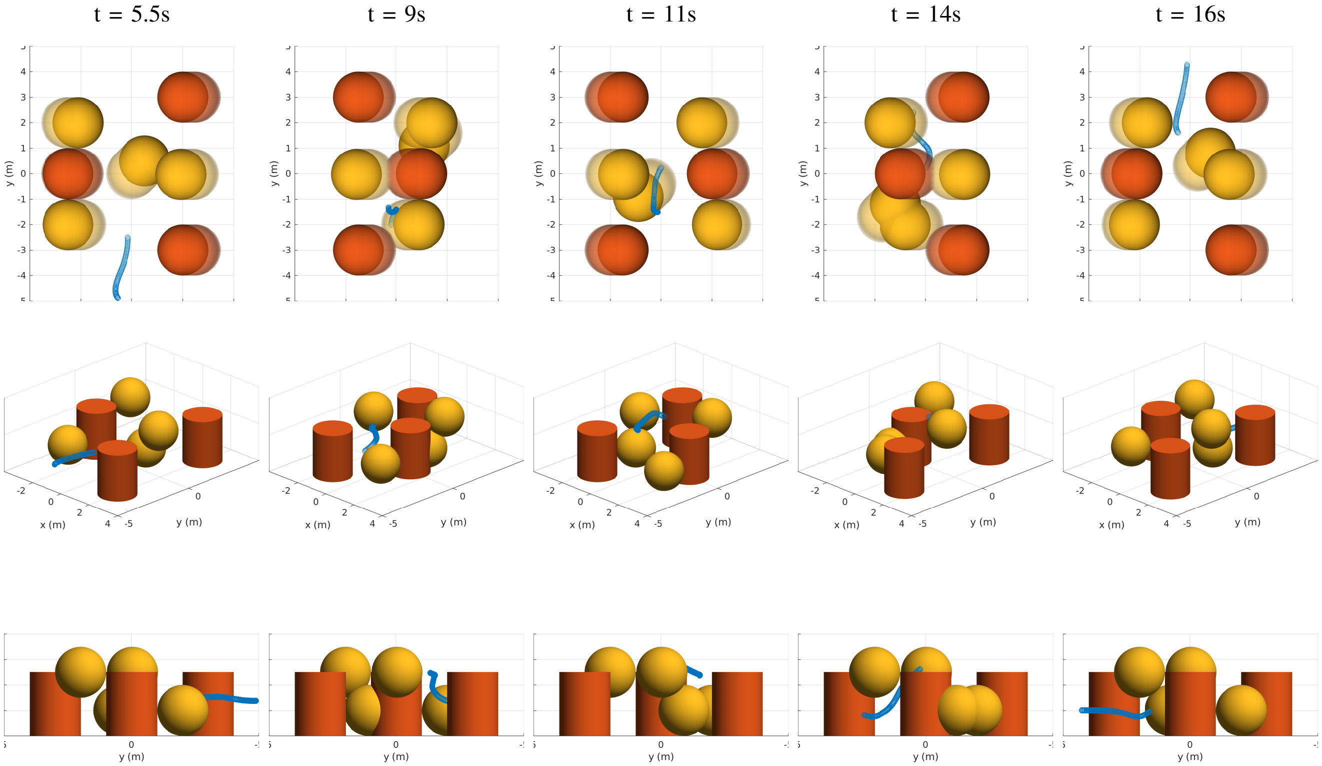


Fig. 11: UAV trajectory in the multiple obstacle scenario

References

[1] K. P. Valavanis and G. J. Vachtsevanos, "Uav applications: introduction," in *Handbook of Unmanned Aerial Vehicles*. Springer, 2015, pp. 2639–2641.

[2] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[3] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Decision and control, 2003. Proceedings. 42nd IEEE conference on*, vol. 4. IEEE, 2003, pp. 3621–3626.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[5] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 3716–3721.

[6] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1610–1616.

[7] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[8] J. Van Den Berg, D. Wilkie, S. J. Guy, M. Niethammer, and D. Manocha, "Lqg-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 346–353.

[9] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 668–674.

[10] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, "Autogenerating microsecond solvers for nonlinear mpc: a tutorial using acado integrators," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 685–704, 2015.

[11] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predkamel2017robustictive control," 2017.

[12] G. Garimella, M. Sheckells, and M. Kobilarov, "Robust obstacle avoidance for aerial platforms using adaptive model predictive control," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5876–5882.

[13] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 528–535.

[14] O. Andersson, M. Wzorek, P. Rudol, and P. Doherty, "Model-predictive control with stochastic collision avoidance using bayesian policy optimization," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4597–4604.

[15] M. Castillo-Lopez, M. A. Olivares-Mendez, and H. Voos, "Evasive maneuvering for uavs: An mpc approach," in *Iberian Robotics conference*. Springer, 2017, pp. 829–840.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[17] S. A. Sajadi-Alamdari, H. Voos, and M. Darouach, "Stochastic optimum energy management for advanced transportation network," in *Control in Transportation Systems (CTS), 2018 IFAC Symposium on*. IFAC, 2018.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[19] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1321–1326.