Contingency MPC for Quadrotor Collision Avoidance

Jason Anderson, Nick Goodson, Adyasha Mohanty

Presentation Video Link

https://youtu.be/SKewirfz8KA

I. Introduction

Autonomous Unmanned Aerial Vehicles (UAVs) are becoming increasingly prevalent in our skies, a trend that will further accelerate as delivery-rotorcraft and urban air transport become ubiquitous. Ample literature exists on Model Predictive Control (MPC) applied to UAV systems; however, in this work, we focus on the challenging problem of controller robustness. Specifically, how do we ensure collision free trajectories of a quadrotor in an urban environment where there is a high risk of encountering obstacles?

Previously, controller robustness in MPC has been addressed with Robust MPC (RMPC) or Stochastic MPC (SMPC). RMPC preserves controller stability and performance for each possible realization of uncertain parameters. Although RMPC has shown encouraging results on systems that cannot transgress stability constraints, it is overly conservative and computationally complex. It accounts for the worst-case system parameters, even if highly improbable. These limitations impede RMPC's application to UAVs given the constraints on battery life, flight time, and computational power. Unlike RMPC, SMPC models uncertainty parameters with probabilistic distributions and minimizes an expected cost. Relaxing hard constraints into probabilistic constraints makes SMPC less conservative than RMPC; however, knowledge of the disturbance model a priori may be challenging to obtain.

Contingency MPC (CMPC) overcomes the limitations of RMPC and SMPC by anticipating potential emergencies rather than reacting when they occur [1]. CMPC separates nominal and contingency planning into separate horizons, each with a unique cost function and constraints. The nominal trajectory tracks a desired path and includes considerations for smoothness and efficiency. The additional contingency trajectory is considered to handle potential dangerous events. Alsterda, *et al.*, apply CMPC to an autonomous car that must safely navigate a potentially icy road. The controller selects the optimal control input accounting for both the nominal and the icy road conditions, thereby ensuring a trajectory that works for both situations simultaneously.

In this work, we apply CMPC to autonomous quadrotors that must safely avoid collision with other quadrotors. The contingency horizon enforces obstacle avoidance constraints given an assumed behavior of the other quadrotors. Our contributions include the following.

1) We use CMPC to enact collision avoidance for a quadrotor through dynamic constraints;
2) We build a CMPC simulator in Matlab with YALMIP [2];
3) We propose and implement constraint horizon scaling to ensure the optimization problem remains feasible between time steps; and,
4) We experimentally verify our framework for different plausible adversarial scenarios. We consider cases where the obstacle collides with our agent given no trajectory modification or when the obstacle is actively seeking to collide.

II. Related Work

Learning-based MPC was explored by Bouffard *et al.*, wherein physically based updates were used to improve transient response [3]. However, the controller was designed for a specific task of catching a ball with a known trajectory. Hu *et al.*, implemented tube MPC, a variant of RMPC for a 10-state quadrotor model [4]. Along similar lines, a tube based MPC was used in [5] to stabilize quadrotor's horizontal dynamics. Although the model accounts for additive disturbances in terms of bounded sets, it is not experimentally verified with static or dynamic obstacles. Obstacle avoidance in an MPC framework was explored in the work of Garimella *et al.*, [6]. The authors combine an online model for parameter identification with an ellipsoidal penetration to demarcate the safety region around obstacles. The work shows robust behaviour for obstacle avoidance but only considers static obstacles, all modeled as cylinders. Static obstacles were also considered in Nascimento's *et al.*, nonlinear MPC formulation for a quad rotor in a 3D unknown environment [7]. A sensor obtains the relative obstacle position and a constraint generator computes polyhedras by adding safety margin to the obstacles.

The main limitation of these approaches is that the robust MPC controllers either do not include obstacle avoidance or have been experimentally verified with only static obstacles. We now discuss several works that have accounted for moving obstacles either through an added cost function or through explicit dynamic constraints. Gros *et al.*, represented obstacles as balls at pre-defined positions with known radii [8] and modelled non-convex avoidance constraints with slack variables. In [9], a collision avoidance cost is designed based on the squared distance between the quad rotor and the obstacle. Along similar lines, in [10] a potential term is constructed as the summation of the inverse distance between the quad rotor and multiple nearest obstacles. A limitation of cost-function based approaches is that they are often excessively conservative due to over approximating the unsafe region. Furthermore significant effort may be required in hyper-parameter tuning to balance obstacle avoidance with goal seeking. This negates a key advantage of CMPC, namely the separation of nominal trajectory considerations from emergency maneuvers. Hence, we opt to use a constraints-based approach as described in the following section.

III. Problem Formulation

A. CMPC Mathematical Formulation

CMPC uses at least two parallel prediction horizons within the standard MPC problem. The different prediction horizons account for each contingency plan; however, the first control input of each plan is constrained to be same.

Consider the rudimentary scenario where a controller must plan for a nominal scenario and one contingency scenario. Let n designate the nominal plan, and let c designate the contingency plan. As with the typical MPC nomenclature, let $j_t(\cdot,\cdot)$ denote the stage cost at time t; x_t the state at time t; u_t, the control input at time t, $f(\cdot,\cdot)$, the one-step dynamics function; $\mathcal{X}$, the permissible state-space set; $\mathcal{U}$, the permissible control set; and x_0, the current estimate of the state. Equations (1) through (6) provide the CMPC formulation. Note Equation (6), which constrains the control input of both scenarios to be equal.

$$J^*(x_0) = \min_{u_{1:N}, x_{1:N+1}} \sum_{t=0}^{N} j_t^n(x_t^n, u_t^n) + j_t^c(x_t^c, u_t^c) \quad (1)$$

$$\text{s.t} \quad x_{t+1}^n = f^n(x_t^n, u_t^n);\; x_{t+1}^c = f^c(x_t^c, u_t^c) \quad (2)$$

$$x_t^n, x_t^c \in \mathcal{X};\; x_t^n \in \mathcal{X}^n;\; x_t^c \in \mathcal{X}^c \quad (3)$$

$$u_t^n, u_t^c \in \mathcal{U} \quad (4)$$

$$x_0^n = x_0^c = x_0 \quad (5)$$

$$u_0^n = u_0^c \quad (6)$$

Like standard MPC, the first control input is applied to the system and the problem is then re-solved.

B. Objective Function and Constraints

We assume a fixed goal which is included in the cost function at every time step. The goal is specified as a desired state and control. The objective function is a standard quadratic cost formulation. Constraints include upper and lower bounds on the state and the control input, the dynamics as given by Equation (2), and collision avoidance constraints described in the next section.

C. Collision Avoidance Formulation

We leave advanced geometric formulations such as polytopes and zonotopes [11][12] for future work.

Obstacles are modelled as spheres with known constant radii. A safety region envelopes the obstacle to provide an avoidance margin. The obstacle-avoidance constraint is formulated in Equation (7), where $p(\cdot)$ denotes the obstacle o 's position at time t; t is the time within the MPC problem; r_t is the obstacle size and s is the safety margin at time t.

$$||p(x_t) - p(x_t^o)||_1 \geq r_t + s \quad (7)$$

Including collision avoidance constraints makes the MPC problem non-convex. Such problems can be handled by using a branch-and-bound optimizer that calls a quadratic program solver.

During simulation, the collision avoidance constraints are frequently active since the controller greedily minimizes the cost. This leads to the quadrotor occasionally entering the avoidance regions, due to plant-controller-model mismatch, making an infeasible problem mid-experiment. We refrain from using soft constraints as the MPC problem would no longer be a branch-and-bound quadratic program, leading to long computation times. Therefore, in our formulation, we allow the radius r_t to grow within the MPC time horizon so that the quadrotor perceives a smaller obstacle at the next MPC step. We let the radius vary as $r_t = \gamma^t\; r_0$ where r_0 is the initial obstacle radius and $\gamma = 1.1$. We found that γ ensured that the quadrotor never entered an infeasible region between time steps in our experiments. To find the minimum constraint-satisfaction safety, we suggest one enforce that the one-step control reachability set from the current position be constrained to lie outside the obstacle avoidance region.

D. Quadrotor Plant and Control Dynamics

To model quadrotor dynamics as a plant, we implemented the second-order accurate model as described here [13]. Within the controllers, we used the linearized-about-hover dynamics as described here [14].

IV. Experimental Results

A. Setup

We investigate the performance of CMPC applied for collision avoidance through numerical experiments in Matlab. The control problems are assembled with YALMIP [2], while IMB's CPLEX 12.1 branch-and-bound quadratic program solver performs the optimizations. The controller and simulation run at 10 Hz.

To focus on collision avoidance performance and isolate tuning effects, we maintain identical MPC horizons in all experiments and only modify the constraints regarding collision avoidance. Each controller tested uses the following common parameters within its trajectory optimizations: 30 0.1 second steps, identity tuning, identical linearized quadrotor dynamic constraints, total thrust limited to four times the total weight, and identical non-active state constraints. Therefore, the only difference between the MPC controller and the CMPC controller is that the CMPC controller includes an additional horizon with the corresponding obstacle related constraints.

In the following sections, each scenario involves two quadrotors: (1) a test quadrotor whose controller performance is evaluated and (2) an adversarial quadrotor on a potential collision course. The test quadrotor controller models the other quadrotor as a moving obstacle with assumed behavior. The controller can make one of the two following assumptions about the other quadrotor's behavior. The first is that the other quadrotor maintains a constant velocity within the MPC horizon. At each time step, the test quadrotor controller observes the other quadrotor's current position and velocity. The second is that the other quadrotor is ballistically attempting to collide with the test quadrotor at a maximum speed.

B. Constant-Velocity Intercept Contingency

We create two test scenarios. In the first scenario, the test quadrotor moves from $(0, 0, 0)$ to $(2, 2, 0)$ while the other quadrotor moves from $(2, 2, 0)$ to $(0, 0, 0)$. Without an avoidance strategy, the two will collide. In the second scenario, the other quadrotor diverts to $(0, 0, -1)$ at time $t = 1$. Without an avoidance strategy, the two will not collide. For both scenarios, we compare the performance of two controllers: (1) an MPC controller that assumes the other quadrotor will maintain a constant velocity through the MPC horizon and (2) a CMPC controller with two contingency horizons: (A) one assuming no other quadrotor present and (B) one assuming the other quadrotor will maintain a constant velocity through the MPC horizon. The CMPC controller continges the presence of the other quadrotor, as opposed to assuming the presence of the other quadrotor; hence, we expect to observe better control performance both when the other quadrotor follows a collision-course and when it diverts.

Figure 1 provides the complete 4-second trajectory of both controllers in the presence of a collision-course drone. Figure 2 provides the complete 4-second trajectory of both controllers in the presence of a collision-course drone that diverts at 1 second. The CMPC controller provides a less-perturbed trajectory for both scenarios. Moreover, the CMPC controller causes the test quadrotor to arrive at its goal state faster, as illustrated by the evenly-temporally-spaced dots.

In these tests, we claim that CMPC exploits two cost-reducing measures to provide the superior performance observed. First, CMPC defers acting on the less-ideal (cost increasing) contingency until the last possible moment it can still safely be managed. The difference between the nominal and collision avoidance trajectories encodes uncertainty about whether the other drone will divert. This uncertainty enables the controller to greedily act on the more fortuitous contingency (that it will divert), until it is no longer safe to do so. This is observed as a delay in the diversion of the CMPC controlled quadrotor from the direct path to its goal. Secondly, because CMPC expressively incorporates the nominal trajectory, the controller is ready to exploit the nominal case when the contingency does not happen.

C. Ballistic Adversarial quadrotor Contingency

We create two additional scenarios, where the other quadrotor is potentially ballistic. In both scenarios, the test quadrotor attempts to maintain the position $(1, 1, 0)$. In the first case, the adversary quadrotor moves from $(2, 0, 0)$ to $(0, 0, 0)$ whereas in the second scenario, it tries to collide with the test quadrotor.

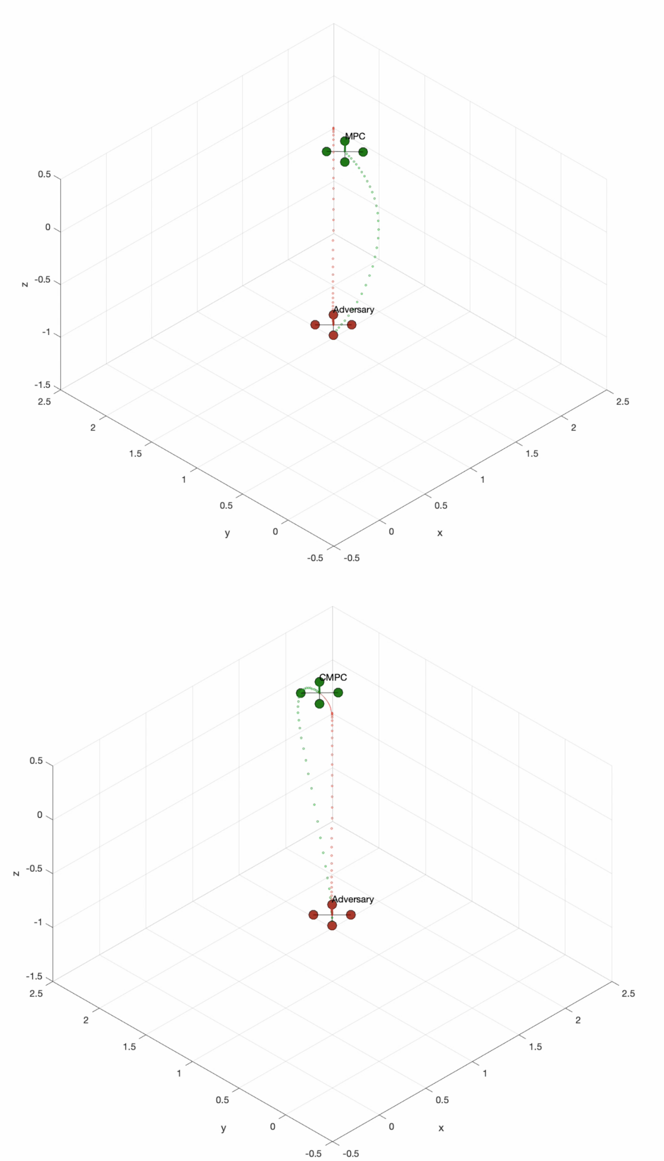

Fig. 1. Comparison results of vanilla MPC (top) and CMPC (bottom) from Section IV-B for the first 4 seconds in the presence of a quadrotor that does not divert. Each of the dots represents 0.1 seconds of movement. Notice that the CMPC trajectory is less perturbed and arrives at the goal state earlier.

For each scenario, we compare the performance of two controllers for the test quadrotor: (1) an MPC controller that assumes the adversary quadrotor is attempting to collide and (2) a CMPC controller with two contingency horizons: (A) one assuming no other quadrotor present and (B) one that assumes an adversarial quadrotor is attempting to collide with the test quadrotor. The CMPC controller continges on a ballistic adversarial quadrotor, as opposed to assuming a ballistic adversarial quadrotor; hence, we expect better performance of the CMPC controller in the presence of both a non-ballistic and a ballistic adversary.

Figure 3 provides the complete 20-second trajectory of both controllers in the presence of a non-ballistic quadrotor. Clearly,

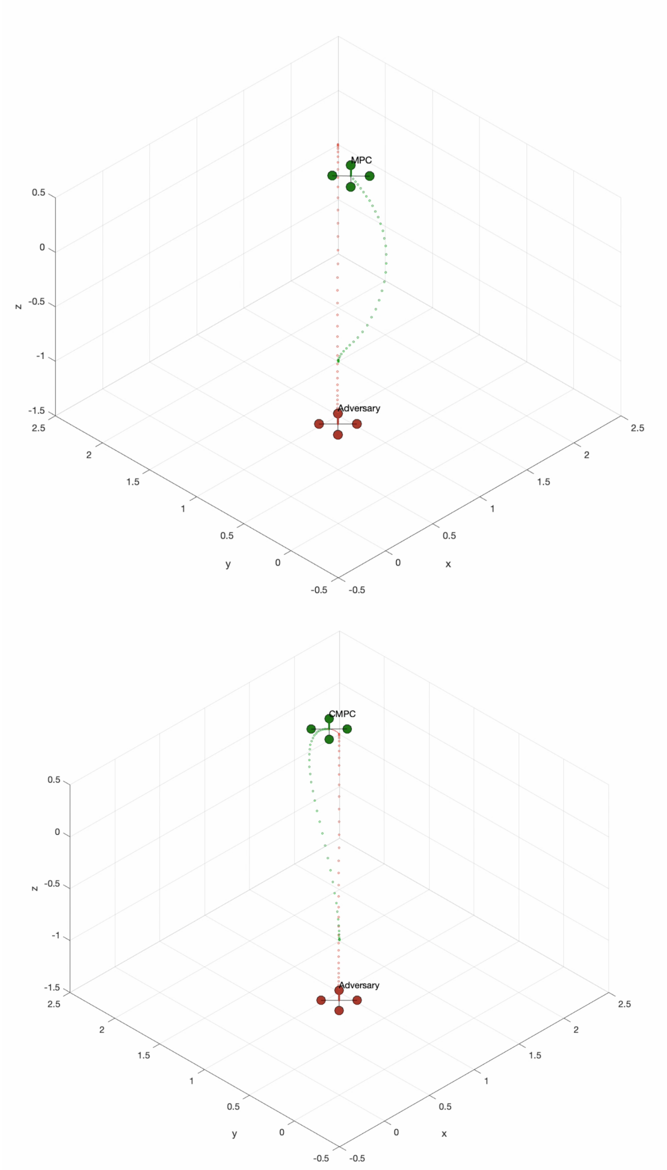

Fig. 2. Comparison results of vanilla MPC (top) and CMPC (bottom) from Section IV-B for the first 4 seconds in the presence of a quadrotor that diverts at 1 second. Each of the dots represents 0.1 seconds of movement. Notice that the CMPC trajectory is less perturbed and arrives at the goal state earlier.

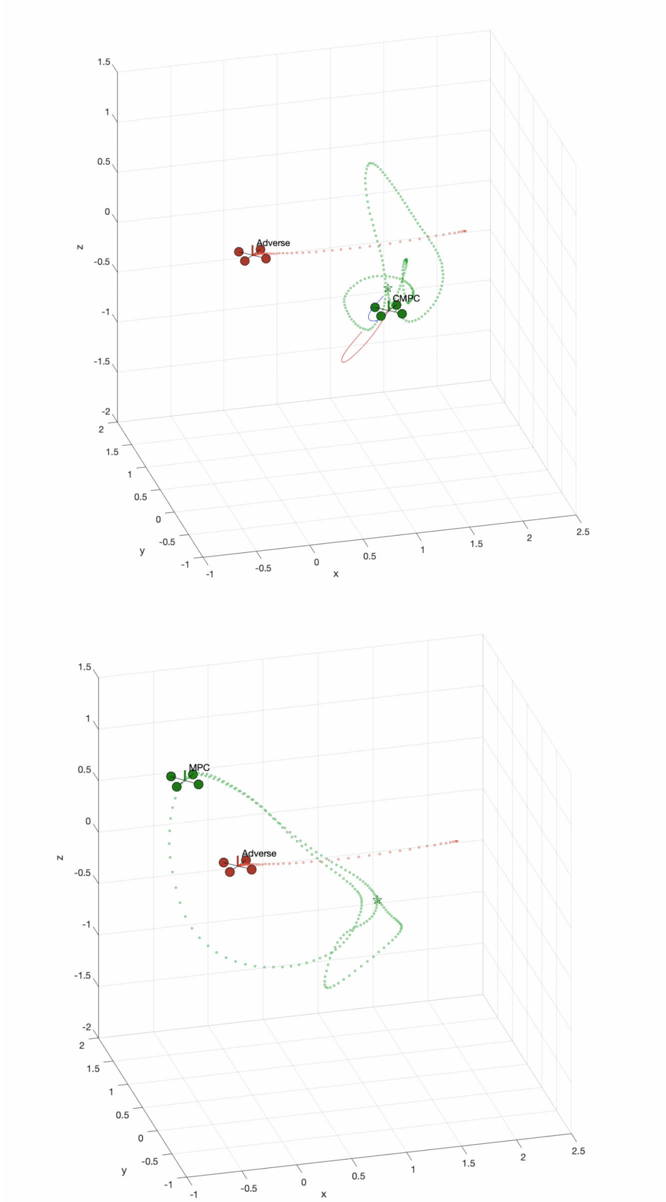

Fig. 3. Comparison results of MPC (top) and CMPC (bottom) from Section IV-C for the first 20 seconds in the presence of a non-ballistic quadrotor. The CMPC controller is better able to keep the quadrotor at its goal position without the presence of the ballistic quadrotor contigency scenario.

the CMPC controller better handles the situation without the contingency occurring. Figure 4 provides the complete 20-second trajectory of both controllers in the presence of a ballistic quadrotor. The CMPC controller outperforms MPC as it keeps the test quadrotor closer to the goal, even in the presence of a ballistic quadroor. Moreover, the CMPC trajectory's average deviation from the goal is smaller and the test quadrotor manages to cross its goal state multiple times.

In this test, like the test from Section IV-B, the CMPC controller is immediately ready to exploit the nominal case cost if the contingency does not occur; hence, the observed performance increase when the other quadrotor is not ballistic. Moreover, the CMPC controller performs better in the ballistic scenario given the extra emphasis on following the nominal scenario.

D. Comparing the Obstacle Behavior Modeling

Figure 5 compares how the CMPC controller accounts for the two obstacle behaviors modelled. In this experiment, the test quadrotor attempts to maintain a position of $(1, 0, 0)$. The other quadrotor moves from $(2, 0, 0)$ to $(0, 0, 0)$. With a constant velocity contingency, the test quadrotor returns to a stationary position after moving itself out of the path of the other quadrotor. With a ballistic contingency, the quadrotor returns to its original position, but does not consider a

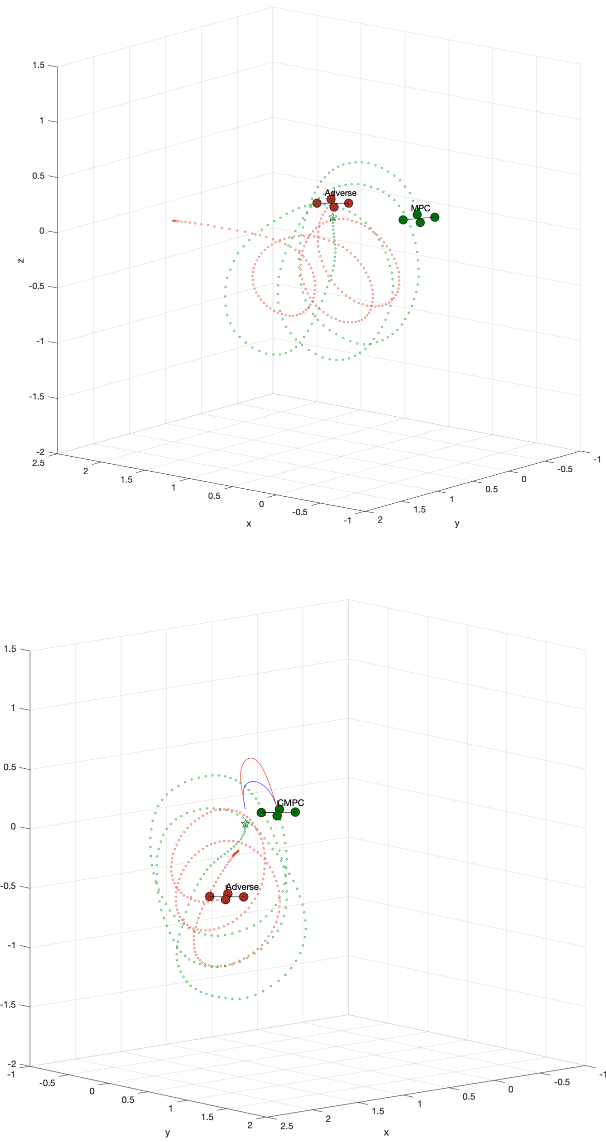

Fig. 4. Comparison results of MPC (top) and CMPC (bottom) from Section IV-C for the first 20 seconds in the presence of a ballistic quadrotor. The quadrotor trajectories fall within a plane, so the view angle is adjusted to be perpendicular to that plane to show maximum trajectory spread. The CMPC controller trajectory spread is smaller, and its quadrotor is able to cross its goal point multiple times.

stationary hover safe, so it completes loops that contain its original position. In this looping trajectory, the test quadrotor maintains a safe state from which it can react if the other quadrotor becomes ballistic.

V. CONCLUSION

CMPC performed better than MPC in our experiments. However, we do not provide any provable, generalizable claims regarding performance, feasibility, and stability. Natural extensions for our work include considering a more realistic formulation of obstacle avoidance using state-estimation, co-

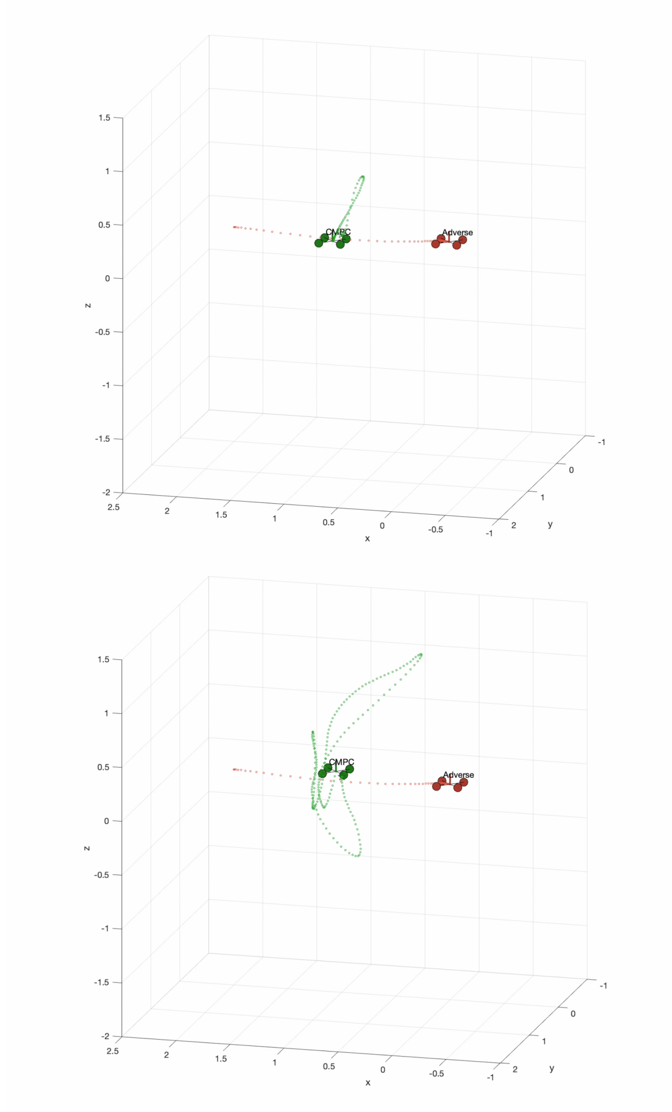

Fig. 5. Comparison results of CMPC controller assuming a constant-velocity quadrotor obstacle (top) and assuming a ballistic quadrotor obstacle (bottom).

operative multi-agent systems, using more than two contingency scenarios and employing different dynamics and cost functions for the nominal and contingency horizons. Furthermore, we suggest investigating scenarios where the optimal nominal and contingency horizons are opposites, potentially resulting in a net zero control. Including HJB reachability analysis in our framework is of key interest as well.

REFERENCES

[1] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *2019 American Control Conference (ACC)*, pp. 717–722, 2019.

[2] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.

[3] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *2012 IEEE International Conference on Robotics and Automation*, pp. 279–284, 2012.

[4] H. Hu, X. Feng, R. Quirynen, M. E. Villanueva, and B. Houska, "Real-time tube mpc applied to a 10-state quadrotor model," in *2018 Annual American Control Conference (ACC)*, pp. 3135–3140, 2018.

[5] N. Michel, S. Bertrand, S. Olaru, G. Valmorbida, and D. Dumur, "Design and flight experiments of a tube-based model predictive controller for the ar.drone 2.0 quadrotor," *IFAC-PapersOnLine*, vol. 52, pp. 112–117, 01 2019.

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

[7] I. B. P. Nascimento, A. Ferramosca, L. C. A. Pimenta, and G. V. Raffo, "Nmpc strategy for a quadrotor uav in a 3d unknown environment," in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 179–184, 2019.

[8] S. Gros, R. Quirynen, and M. Diehl, "Aircraft control based on fast non-linear mpc multiple-shooting," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 1142–1147, 2012.

[9] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos, "A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors," in *2016 IEEE Conference on Control Applications (CCA)*, pp. 519–525, 2016.

[10] D. Shim, H. Chung, H. J. Kim, and S. Sastry, "Autonomous exploration in unknown urban environments for unmanned aerial vehicles," *IEEE Robotics & Automation Magazine*, vol. 13, 08 2005.

[11] A. Girard, "Reachability of uncertain linear systems using zonotopes," vol. 3414, pp. 291–305, 03 2005.

[12] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," 10 2019.

[13] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, 2011.

[14] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 498–503, 2016.