Online Collision-free Formation Flight under Dynamic and Static Obstacles

Ben Moore
Aerospace Department
Stanford University
Stanford, USA
benmoore@stanford.edu

Wouter Van Gijseghem
Aerospace Department
Stanford University
Stanford, USA
wvg@stanford.edu

***Abstract*—The goal of this paper is to present an online control algorithm based on model predictive control (MPC) that allows for collision-free formation flight of a number of unmanned aerial vehicles (UAVs). The UAVs will follow a set of pre-determined way-points while avoiding both static and dynamics obstacles and inter-vehicle collisions. No prior or global knowledge of the map is assumed as with MPC, the obstacles will be sensed by the UAVs when they are within the time-horizon. To enable formation flight, a virtual reference point method is used and compared to the more common consensus-style formation control. The objective function of the MPC consists of three distinctive parts: the tracking error, an obstacle avoidance penalty term and an inter-vehicle collision avoidance penalty term. The problem is also separately formulated with an additional constraint by under approximating the feasible space as the union of polyhedrons which enforces collision avoidance as a hard constraint. This method allows for guaranteed obstacle avoidance in contrast with the penalty methods and has the advantage that no non-convex constraints have to be included. The simulation results show that the algorithm presents feasible, collision free trajectories while maintaining formation. The controller is proven to be stable and asymptotically convergent to an equilibrium point using Lyapunov's direct method when formulated as a dual mode controller. It is shown that the algorithm remains computationally tractable even when including many drones and obstacles and is with the exception of the inter-vehicle obstacle avoidance fully distributed.**

***Index Terms*—MPC, formation flight, collision avoidance**

I. Introduction

The significance of unmanned aerial vehicles (UAVs) has been steadily increasing over the last decade. UAVs can be used in numerous important applications, such as goods delivery, maintenance, inspection and surveillance. These techniques are particularly pertinent at this time as large scale surveillance drone teams are currently operated manually to monitor cities to enforce social distancing during virus outbreaks, which could be performed much more effectively using a centralized algorithm similar to the one demonstrated in this paper. Extensive research has been conducted in multi-agent control of distributed UAV systems for collaborative operations. [1] Formation flight has proven to be one of the more challenging aspects of multi-agent control as it involves the challenge of achieving and maintaining a desired formation, minimizing control effort and avoiding dynamic and static obstacles. Integrating formation flight into modern day airspace requires that all these goals are satisfactorily met before it can provide value in real world scenarios.

The challenge of formation flight under static and dynamic obstacle avoidance has been been previously formulated using a number of various techniques; Sequential Convex Programming (SCP) [2], [3], online optimization [4]–[6], game theoretic [7], potential field [8] and heuristic geometric [9], [10] methods.

A hierarchical model predictive control (MPC) leader follower approach is proposed in [4] as a method for the stabilization and formation flight problem, given obstacle avoidance and dynamics constraints. The upper level linear time variant (LTV) MPCs generate desired positions to reach a target while avoiding collisions. These desired positions are then tracked in real time by the lower level linear time invariant (LTI) MPCs. The non-convex feasible space for navigation is underappoximated by a convex polyhedron in order to impose collision avoidance constraints as linear constraints. In order to maintain formation, a decentralized leader follower approach is implemented whereby the leader tracks a desired target position and each unmanned aerial vehicle (UAV) tracks a desired relative distance from the leader.

In [3] the problem is formulated as a constrained optimization problem which is solved using sequential convex programming (SCP) that plans locally. The robots have the added ability to adjust the size and orientation of the formation in order to achieve the target under obstacle avoidance constraints. A global alternative is proposed by sampling convex regions in free space and connecting them if a transition in formation is possible, which is found via a constrained optimization problem. The path with the lowest cost in terms of deviation from the formations goal state, size and rotation is then selected via graph search.

A dual mode control strategy is proposed in [6], which includes "safe" and "danger" modes to operate in obstacle free spaces and potential collision scenarios respectively. The safe mode controller is formulated using an upper layer infinite time horizon linear quadratic regulator (LQR) in order to track the desired reference trajectory based on a relative state. A lower layer MPC then drives each UAV individually. The safe mode controller achieves global optimization as all participating UAV dynamics are considered. The danger mode controller uses a Grossberg neural network (GNN) for collision avoidance. A buffer zone is created around the obstacles and neurons are placed at the vertices of this zone. Neurons are only connected to other neurons if there is a direct line of sight between them, these are known as neighbors. Each neuron only receives excitation from its neighbors and therefore the optimal obstacle free path is generated by moving from one vertex to the next vertex that has the highest excitation.

A very novel and interesting approach is to cast the problem as a non-convex optimization which is then iteratively solved using SCP which approximates non-convex collision avoidance constraints and minimises thrusts required for trajectories. [2] The non-convex collision avoidance constraint, represented by the ℓ_2 norm of the relative positions of two UAVs is approximated using a first order Taylor expansion. The centralized SCP algorithm guarantees feasibility of the trajectory by setting limits on the thrust input and jerk of the UAVs. However in this paper, the focus is more on the static and dynamic collision avoidance and formation flight is not addressed.

A more restrictive and uncommon way to approach the formation flight problem is presented in [9] as a heuristic obstacle avoidance strategy and control for a UAV leader. When the formation encounters an obstacle the formation is split and then regroups after passing the obstacle. A simple PID controller is implemented to maintain the desired formation.

In [10], a more general geometric approach is presented. The authors present a Lyapunov function based on the flight path angle and the heading angle of the line-of-sight vector. The line-of-sight vector in this paper is defined as the vector from the vehicle tangent to the closest constraint. Using this Lyapunov function, the authors characterize the stability of the proposed guidance law for collision avoidance. Then additional dynamic and detection constraints are introduced to better characterize the envelope for collision avoidance. This envelope guarantees successful collision avoidance and formation keeping and is a sense-and-avoid method which does not require a-priori or global knowledge of the environment.

A distributed non-linear MPC formation flight controller is proposed in [5], which is designed based on the tracking error to a reference trajectory. The controller solves an optimal control problem online by minimizing the formation's deviation from an assigned relative position and heading to a virtual reference points which tracks the reference trajectory. A priority tagging level is assigned to each UAV based on the relative position of the drones to each other at each sample time. This strategy avoids unnecessary chain manoeuvres. The non-linear optimization problem is solved using the filter SCP method.

In [11] a formation control scheme is presented based on a potential force to obtain a desired formation with collision avoidance. An artificial potential function is built between the leader robot and its destination and the same function is inserted into the interactive potentials between all neighbors of the leader. This means that as the leader is being attracted to the target, the other robots are attracted to the leader while maintaining formation. The angle of the force generated by the potential field represents the desired orientation for the heading direction of the robot. In order to achieve obstacle avoidance the current robot state is projected on to the obstacle geometry in sight and an avoidance velocity is calculated which in turn force the robots to avoid the obstacle. The Lyapunov stability theorem is used to construct a controller with smooth continuous feedback for which stability is proved for multiple robots.

The problem can also be modelled as a differential game between multiple agents, whereby formation control is formulated as a linear-quadratic Nash differential game through the use of graph theory. [7] By building the Nash equilibria all robots find a self-enforcing controller, which guarantees the existence of the formation control, which is stable in terms of their individual cost functions. The controller performs in a distributed manner where individual robots only communicate with neighbors were the leader only tracks the desired trajectory and followers only follow the leader.

The paper is organized as follows. Sections II-A, II-B, II-D and II-C describe the flight dynamics, provide the problem formulation of formation flight, formulate the MPC optimization problem and outline the various collision avoidance strategies implemented respectively. Section III uses Lyapunov's direct method to prove asymptotic stability of the system. Section IV presents the simulation results. Finally, Section V includes concluding remarks and future work.

II. Problem Formulation

A. Dynamics

An UAV can be modelled as an underactuated six degree of freedom system with four control inputs. However, for this problem, the dynamics of the UAV are approximated using a linear dynamical model, shown in Equation 1, with the degrees of freedom being the position and velocity in three dimensions and the control inputs are taken to be the accelerations in the three dimensions. It is shown in [2], that this dynamical model results in feasible trajectories when the bounds on jerk and accelerations are properly set and the time step is sufficiently small to admit a linearization of the system.

$$\begin{aligned} x(t+1) = I_N \otimes \left(\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \otimes I_3 \right) x(t) \\ + I_N \otimes \left(\begin{bmatrix} \Delta t^2/2 \\ 1 \end{bmatrix} \otimes I_3 \right) u(t) \end{aligned} \tag{1}$$

In Equation 1, $x(t) \in \mathbb{R}^{6N}$ and $u(t) \in \mathbb{R}^{3N}$ represent the concatenated state of N robots, where $x_i = \begin{bmatrix} p_x & p_y & p_z & v_x & v_y & v_z \end{bmatrix}^T$ and $u_i = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$, $\forall i \in \{1, .., N\}$. Δt is set to 1 for the purpose of this paper. It is clear that changing Δt results in a scaling of the magnitude of the velocity and control input without altering the behavior of the system. These dynamics can be easily interchanged with different linear dynamical systems and the performance of the algorithm will be the same.

B. Formation Control

Two different types of formation control are compared. The first type is the more common relative distance formation control used for consensus analysis and coverage optimization. For every formation, a Laplacian matrix $\mathcal{L}$ is defined in which for each node n_i:

$$\mathcal{L}_{ij} = \begin{cases} deg(n_i) & \text{if} \quad i = j \\ -1 & \text{if} \quad j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Together with a relative distance matrix $\mathcal{D}$ in which for ever agent i, $\mathcal{D}_{ij} = x_j^d - x_i^d$ with $j \in \mathcal{N}_i$. x_i^d is the desired state of agent i. It is assumed that $\mathcal{D}_{ij}$ is consistent ($\mathcal{D}_{ij} = -\mathcal{D}_{ji}$) and physically realizable.

The second formation control method implemented is the virtual reference point method. In this method, a virtual point is defined following a predetermined trajectory. The other UAVs try to maintain a a-priori known relative distance to this virtual reference point. This is very similar to the first type of formation control presented, however there is no error propagation between the drones in the formation which leads to more robust MPC optimization results. The desired state of each drone in the formation can be determined using Equation 3.

$$\begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \end{bmatrix} + R_y(-\gamma) R_z(-\psi) \begin{bmatrix} x_i^{dr} \\ y_i^{dr} \end{bmatrix} \tag{3}$$

In Equation 3, R_y and R_z are the standard rotation matrices around the y-axis and the z-axis. γ and ψ represent the flight path angle and the heading angle of the UAV respectively. Superscript ^{r} indicates the position of the virtual reference point, while the superscript ^{dr} indicates the desired relative position from the virtual reference point for agent i.

C. MPC Formulation

The formation flight problem can be formulated as follows:

$$x_i(k) - x_i^d(k) = 0 \quad \forall i \in 1, .., N \tag{4}$$

Equation 5 presents the standard MPC formulation for the collision avoidance problem.

$$\begin{aligned} \min_u \quad & f_0(x, u) \quad \forall s \in \{1, .., C\} \\ \text{s.t.} \quad & u_{min} \leq u_i(k+s|k) \leq u_{max} \quad \forall i \in \{1, ..N\} \\ & x_{min} \leq x_i(k+s|k) \leq x_{max} \quad \forall i \in \{1, ..N\} \\ & x(k+s+1|k) = A \cdot x(k+s|k) + B \cdot u(k+s|k) \\ & \|u(k+s+1|k) - u(k+s|K)\| \leq \text{jerk}_{\max} \\ & u(k+s|k) = u(k+C|k) \quad \forall s \in \{C+1, .., K\} \end{aligned} \tag{5}$$

In Equation 5, C is the control horizon, K is the prediction horizon. $f_0(x, u)$ is the objective function which will be elaborated below. $\text{jerk}_{\max}$ is the maximum jerk constraint of the UAV to guarantee feasible trajectories. The minimization of Equation 5 results in a sequence of optimal $u^\star$ and corresponding $x^\star$. Clearly MPC is an online algorithm where $x(k+1|k) = Ax(k) + Bu^\star(k)$ after which MPC is solved for the next optimal control input. Note that the constraints for obstacle avoidance are not yet included Equation 5 as they are elaborated in Section II-D.

Two different cost functions are investigated in this paper for both formation control methods. Equation 6 presents the cost function used for the virtual reference point method, where the cost is proportional to the error to the reference trajectory for every drone and the control input. Equation 7 presents the cost function for the relative distance formation control where the error depends on the deviation of the distance between the drones and the predetermined relative distance.

$$\begin{aligned} f_0(x, u) = {} & e(k+K|k)^T Q_n e(k+K|k) + \\ & \sum_{s=1}^{K} e(k+s|k)^T Q e(k+s|k) + u^T(k+s|k) R u(k+s|k) \end{aligned} \tag{6}$$

$$\begin{aligned} f_0(x, u) = {} & e(k+K|k)^T Q_n e(k+K|k) + \\ & \sum_{s=1}^{K} \Big(u^T(k+s|k) R u(k+s|k) + \\ & \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(x_i(k+s|k) - x_j(k+s|k) - \mathcal{D}_{ij} \right)^T Q \\ & \left(x_i(k+s|k) - x_j(k+s|k) - \mathcal{D}_{ij} \right) \Big) \end{aligned} \tag{7}$$

In the Equation 6 and 7, $e(k+s|k) = x(k+s|k) - x^d(k+s)$, Q, $Q_n \in \mathbb{S}_{++}^{6N}$ and $R \in \mathbb{S}_{++}^{3N}$.

D. Collision Avoidance

As mentioned previously, this paper compares multiple obstacle avoidance methods and compares their performances. In the section, all methods implemented in the algorithm are presented and discussed.

1) Explicit non-convex obstacle constraints: The most trivial obstacle avoidance constraint is to explicitly state that the distance between drones must be bigger than the minimum distance to avoid collisions and the distance between the obstacles and the drones has to be larger than the radius of the obstacle added to the radius of the drone. This can be trivially translated in Equation 8.

$$\begin{aligned} & \|x_i(t) - x_{obs}^o\| \geq R_{ons}^o + R_{drone} \quad \forall i \in \{1, ..N\}, \forall o \in \{1, ..., M\} \\ & \|x_i(t) - x_j(t)\| \geq 2R_{drone} \quad \forall i, j \in \{1, ..N\} \end{aligned} \tag{8}$$

In Equation 8 it is assumed that all drones are circular with radius R_{drone} and the static obstacles are infinite cylinders in the z-direction with radius R_{obs}^o at location x_{obs}^o for obstacle o. The constraints are non-convex and the optimization is not guaranteed to have a unique global minimizer which means optimality of the trajectory is not guaranteed. However due to the explicit obstacle avoidance constraints, if a feasible solution is found, it is guaranteed to be collision-free.

2) Penalty Method: Since non-convex constraints are inherently intractable, they can be replaced by a penalty term in the cost function, similar to the interior points method. One has to note that by removing the explicit constraints, no collision-free trajectories can be guaranteed and although using penalty terms speeds up the algorithm significantly, the optimization remains non-convex and is not guaranteed to have a global unique minimizer or to be solvable in polynomial time.

A dangerous distance is defined about an obstacle along with an allowable distance. Equation 9 demonstrates how the two regions are used to determine whether a cost term has to be added to prevent a collision. As depicted in Figure 1, when the shortest distance between drone and obstacle is less than the dangerous distance l_D, the position and velocity orientation of the drone are used to predict if the shortest distance between them is less than the minimal allowable distance l_M over the prediction horizon. If the drone is predicted to become closer than l_M to the obstacle, a cost term is added causing the drone to start avoiding the obstacle. Two different cost terms are implemented and compare: log barrier and linear. In Equation 9, the linear penalty term is presented. In Equation 10, the log barrier penalty term is shown. The formulation applies to both static and dynamic obstacles where the dynamic obstacle centers are updated at each time step.

$$L_o(x_i,k) = \begin{cases} \sum_{s=1}^{K} -a(l^s_{i,o}(k+s|k) - l_M) & \text{if } l^s_{i,o}(k) > l_D, \\ & l^{sp}_{i,o}(k) < l_M \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$L_o(x_i,k) = -a\log(l^s_{i,o}(k)) \tag{10}$$

In Equation 9 and 10, a, l_D and $l_m \in \mathbb{R}$ and the distances are defined as follows, in which x^o_{obs} represents the position of the obstacle o:

$$l^s_{i,o}(k) = \|x_i(k) - x^o_{obs}\| - R^o_{obs} - R_{drone} \tag{11}$$

$$l^s_{i,o}(k+s|k) = \|x_i(k+s|k) - x^o_{obs}\| - R^o_{obs} - R_{drone} \tag{12}$$

$$l^{sp}_{i,o}(k) = \frac{|\cot(\psi_i(k))\,\|x^o_{obs} - x_i(k)\|^2\,|}{\sqrt{(\cot(\psi_i(k)))^2+1}} \tag{13}$$

Finally to get the full penalty term, the sum is taken over time and over the obstacles as in Equation 14

$$\text{penalty} = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{o=1}^{M} L_o(x_i,k) \tag{14}$$

M is the number of obstacles.

3) Convex Approximations for Obstacle Avoidance: The obstacles and free space can be approximated by convex polyhedrons, as a collection of half spaces defined by a number of linear inequalities. The polyhedron $P = \{p \in \mathbb{R}^d | A_c p \leq b_c\}$, formed using Equation 15, contains x_i in its interior and does not contain any of the points x^o_{obs}, the obstacle centers, in its interior,$\forall o \in \{1,\ldots,M\}$. Once formulated as linear inequalities, the optimization problem described in Equation 16 is solved for each robot with respect to each obstacle. From this additional optimization problem, the constraints $P = \{p \in \mathbb{R}^d | A_c p \leq b_c + g\}$ can be added to the optimization problem and ensure obstacle avoidance if a feasible solution exists. W_o represents a convex polyhedron of an obstacle o. By taking into account all time steps in the prediction horizon, the union of all convex feasible regions over this time period provides an approximation of the non-convex feasible space available to navigate over that time horizon.

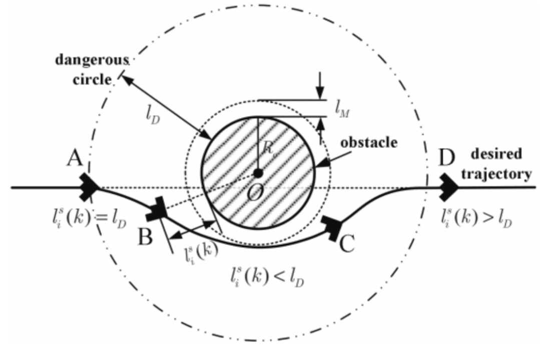


Fig. 1: Collision Avoidance [5]

$$A_c = \begin{bmatrix} (x^1_{obs} - x_i)' \\ \vdots \\ (x^M_{obs} - x_i)' \end{bmatrix}, \quad b_c = \begin{bmatrix} (x^1_{obs} - x_i)' x^1_{obs} \\ \vdots \\ (x^M_{obs} - x_i) x^M_{obs} \end{bmatrix} \tag{15}$$

$$\begin{aligned} g^j = \min_{w \in R^d} \quad & A^j_c w \\ \text{s.t.} \quad & w \in W_j \end{aligned} \tag{16}$$

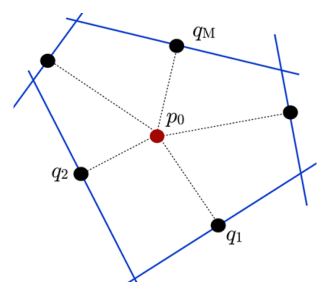


Fig. 2: Convex polyhedron about a point p_0 [4]

III. MPC Stability

One of the disadvantages of MPC is that over a finite cost horizon, there is no guarantee that the control law will achieve optimal performance or even result in a stable and feasible trajectory. There are many other controllers which utilise an infinite cost horizon to address this problem however in the case of MPC this would result in an infinite number of optimization variables, thus making the problem impractical. A dual mode control paradigm is used to reduce the optimization problem to a finite number of variables. [12] In this scenario the MPC calculates the control explicitly up to a fixed horizon as before and the cost is calculated up to the last step of the horizon. [13], [14] The final state is considered separately however and an infinite horizon LQR is used to determine the terminal cost.

Lyapunov's direct method is used to prove asymptotic stability of the controller. In order to do this the following two statements must be shown:

- $V(x) \succ 0$
- $V(x_{k+1}) - V(x_k) \preceq -l(x) \preceq 0$

where $l(x)$ converges to zero as $k \to \infty$. The cost function is given by Equation 17.

$$V(x_k) = \sum_{i=1}^{C}(x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) + x_{k+C+1}^T P x_{k+C+1} \tag{17}$$

By definition $Q, R \in \mathbb{S}_{++}$ which ensures that $V(x_k)$ is a positive definite function. The terminal weight P is chosen such that $V(x_k)$ is an infinite horizon cost. This is achieved by assuming a fixed feedback control law in the form of an infinite horizon LQR and solving the Lyapunov equation [12] in Equation 18.

$$P - (A + BK)^T P (A + BK) = Q + K^T R K \tag{18}$$

were K is a fixed feedback control law and A and B represent the vehicle dynamics from Equation 1. Assuming a feasible input sequence then the optimal predicted cost is nonincreasing and satisfies Equation 19 as the optimal cost at current time k must be at least as small or smaller than the cost of the previous sample.

$$V(x_{k+1}) - V(x_k) \leq -l(x) = -(x_k^T Q x_k + u_k^T R u_k) \tag{19}$$

The following $\mathbf{u_{k+1}}$ is defined:

$$\mathbf{u_{k+1}} = [u_{k+1}, u_{k+1}, \ldots, u_{k+C-1}, K u_{k+C}]^T \tag{20}$$

$\tilde{V}(x_{k+1})$ is the optimal cost associated with $\mathbf{u_{k+1}}$ calculated at time step k.

$$\begin{aligned} \tilde{V}(x_{k+1}) &= \sum_{i=1}^{C}(x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) + \\ &\quad x_{k+N+1}^T P x_{k+N+1} \\ \tilde{V}(x_{k+1}) &= \sum_{i=1}^{\infty}(x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) \\ \tilde{V}(x_{k+1}) &= \sum_{i=0}^{\infty}(x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) - \\ &\quad (x_k^T Q x_k + u_k^T R u_k) \\ \tilde{V}(x_{k+1}) &= V(x_k) - (x_k^T Q x_k + u_k^T R u_k) \end{aligned} \tag{21}$$

K is the control feedback law calculated using the infinite horizon LQR. Since $\tilde{V}(x_{k+1})$ is suboptimal as it is based on the optimal predictions from the previous time step k, the optimal value $\tilde{V}(x_{k+1})$ at time step $k+1$ satisfies Equation 19.

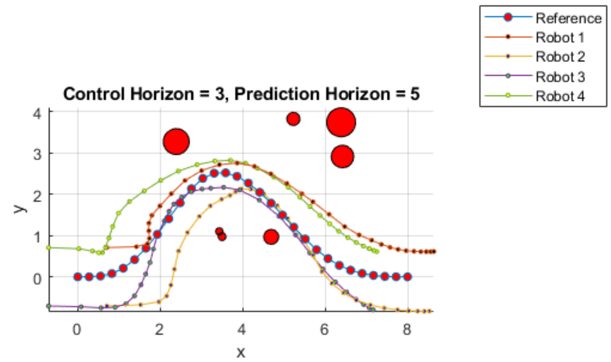


Fig. 3: *Case 1*, with explicit obstacle avoidance constraints with relative distance formation control

$$V(x_{k+1}) \leq \tilde{V}(x_{k+1}) = V(x_k) - (x_k^T Q x_k + u_k^T R u_k) \tag{22}$$

Therefore it has been proved that formulating the optimization problem over an infinite horizon satisfies Lyapunov's direct method criteria and asymptotically converges to $x_{eq} = 0$. Note that by considering the problem in a dual control mode format allows the terminal weight P to be determined using an infinite horizon LQR at each time step.

IV. Results and Discussion

Multiple simulations are conducted for each obstacle avoidance strategy described in Section II-D, under different scenarios, varying number of drones and obstacles as well as varying control horizon length. Two different reference trajectories are implemented. One where the leader follows a Gaussian-like curve and the reference velocity is equal to the derivative of this curve: *Case 1*, and one where the leader first goes horizontally and then vertically at a constant speed: *Case 2*. In both cases, four drones are in a square formation, thus distributed at at 45deg angle from the virtual reference point. In Figure 3 and Figure 4, the two cases are compared. The formation is maintained during the flight and is only broken when the drones have to avoid each other or the obstacles. The algorithm is robust against varying input trajectories, cost matrices and changing both the control horizon and prediction horizon. Note that in *Case 1*, the consensus style formation control is used, while in *Case 2*, the virtual reference point method is used. No considerable difference between the two formation controls is observed except for a higher control cost for the consensus style formation control. This is attributed to the fact that the control has to counteract the relative errors introduced by this method.

In Figure 6, the different obstacle avoidance methods from Section II-D2 are compared for *Case 2*. It is clear that all methods results in collision-free and feasible trajectories. While

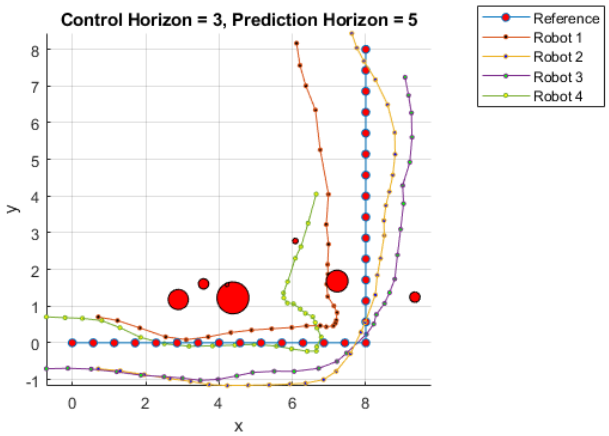


Fig. 4: *Case 2*, with explicit obstacle avoidance constraints with virtual reference point formation control

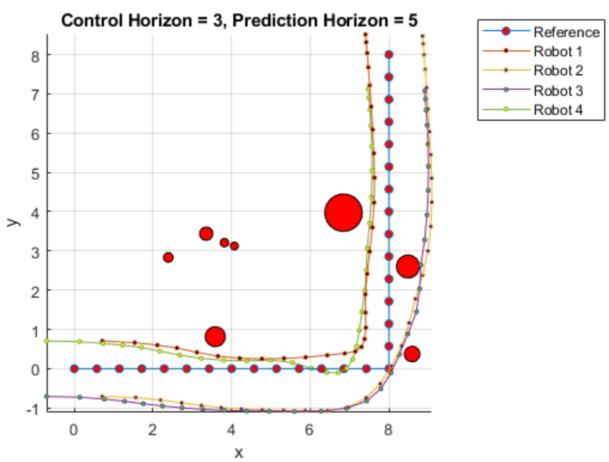


Fig. 5: *Case 2*, with under approximated non-convex feasible region using polyhedrons

the runtime of both penalty methods are similar, including the explicit obstacle avoidance constraints increases the run time significantly. It has to be noted that including the explicit obstacle avoidance constraints results in the lowest control cost over the complete trajectory.

It is observed through the simulation that obstacle avoidance via the addition of a penalty term to the cost function is very sensitive to the weighting of the additional cost. This sensitivity would result in either a collision occurring or the drones would avoid the final state entirely in order to ensure a collision does not occur. In order to avoid these two events careful tuning of the weighting of the cost term is required depending on the scenario. This is a disadvantage when using the algorithm in a real world scenario and requires sufficient testing of the system.

To get around the expensive nature of the explicit obstacle avoidance constraints or the loss of guaranteed collision-free trajectories using the penalty terms, the obstacle avoidance constraints are formulated as the linear inequality constraints presented in Section II-D3. The assumption is made that the initial position is feasible. In Figure 5, the result is shown for *Case 2*. A feasible trajectory is obtained and the constraints have no sensitivity issues as was the case for the penalty term method. The authors also note that using the linear inequality constraints results often in smoother trajectories, however the cost is significantly higher than using the other methods of collision avoidance.

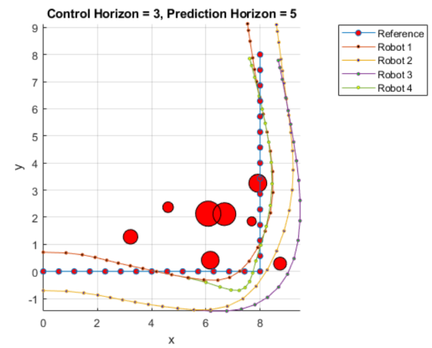
In Figure 7, the runtime analysis results of the algorithm are presented. The MPC algorithm is run on *Case 2* with the penalty term obstacle avoidance. From this, it is concluded that the MPC algorithm scales with $\mathcal{O}(N^2)$, where N is the number of drones, behaves like $\mathcal{O}(C)$, where C is the control horizon and scales with $\mathcal{O}(K)$, in which K is the number of obstacles. Therefore it is shown that the algorithm presented in this paper, although not being convex, is computationally tractable and can be run in real-time on-board of drones.

V. Conclusion and Future Work

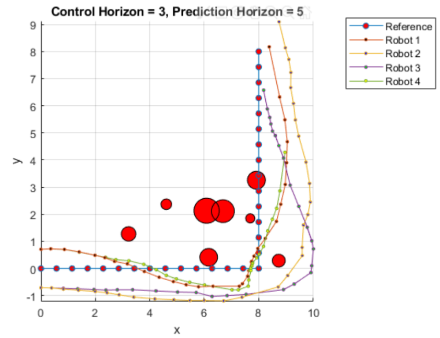
In this paper the problem of formation flight has been formulated and implemented as an MPC controller using both hard and soft constraints for collision avoidance. The lack of stability guarantees commonly associated with MPC controllers has been addressed by formulating the controller as a dual mode controller. The stability of the system has been shown to be asymptotically convergent to an equilibrium point using Lyapunov's direct method. Simulations of obstacle avoidance methods including: explicit non-convex constraints, penalty method and convex approximations for obstacle avoidance have been used to demonstrate that the MPC algorithm is able to find optimal collision-free trajectories while maintaining formation. The runtime of these simulations has been analysed and the algorithm is shown to run in polynomial time indicating it can be scaled to larger system and implemented on hardware to run in realtime.

Obstacle avoidance has always been a difficult non-convex problem to solve. The non-convex nature mandates that there is no global unique minimizer to the optimal control problem and no computationally tractable way to guarantee an optimal can be found. Relaxing the non-convex explicit obstacle avoidance constraints to penalty terms in the cost function, allows one to considerably speed up the MPC optimization, however one loses the guarantee of collision-free trajectories. This is solved by approximating the non-convex feasible region as a convex polyhedron. Linear inequality constraints can be added to the problem to guarantee obstacle avoidance, however the feasible region is considerably smaller due the under-approximation.

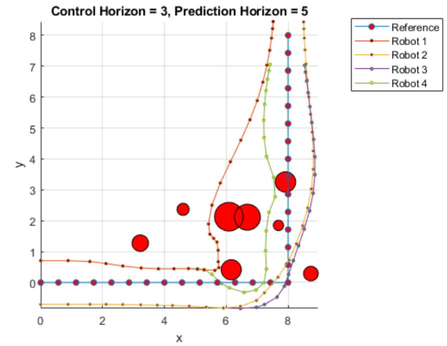
Perfect sensing of obstacles and drones was assumed as well as perfect communication between a centralized control unit and all drones. In order to further explore the practical implementation of this controller further work should consider the robustness of the controller to time delays, external noise on sensors, vehicle dynamics and control inputs. [15] A robust



(a) Log-barrier penalty term, CPU time: 22 sec

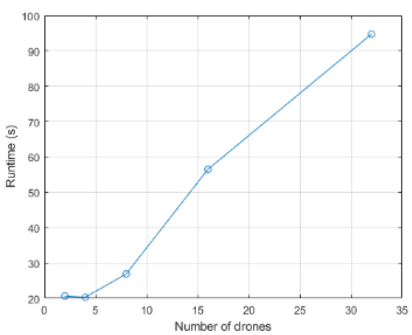


(b) Explicit obstacle avoidance constraint, CPU time: 30 sec

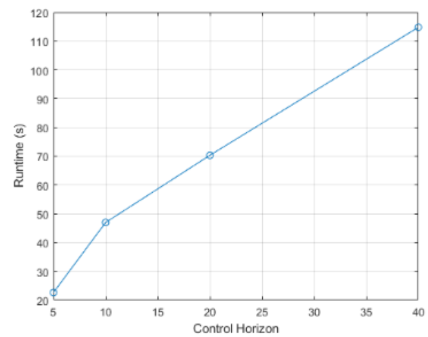


(c) Linear penalty term, CPU time: 23 sec

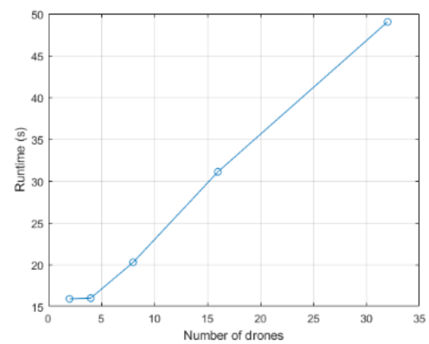
Fig. 6: Comparison of the explicit constraints and the penalty methods



(a) Runtime dependence on the number of drones



(b) Runtime dependence on the control horizon length



(c) Runtime dependence on the number of obstacles

Fig. 7: Runtime analysis of the MPC algorithm for the penalty term obstacle avoidance

controller analysis should be conducted to determine the magnitude disturbances and delays the system can sustain while still maintaining acceptable performance. A current limiting factor is the centralized nature of the algorithm which requires global information for inter-vehicle collision avoidance, sustained communication network to communicate control commands and to receive state information form each drone. To enable a fully distributed algorithm, an approach as in [16] should be implemented to decentralize all computation. This problem uses an assumed set of control inputs of other agents over the prediction horizon which may deviate wildly from the real inputs depending on the scenario. Further research should also consider a time varying network topology and the effects on the ability of the network to maintain formation.

References

[1] Z. Lian and A. Deshmukh, "Performance prediction of an unmanned airborne vehicle multi-agent system," *European Journal of Operational Research*, vol. 172, no. 2, pp. 680 – 695, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221704008161

[2] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1917–1922.

[3] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017. [Online]. Available: https://doi.org/10.1177/0278364917719333

[4] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec 2011, pp. 7488–7493.

[5] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang, "Collision-free uav formation flight control based on nonlinear mpc," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, Sep. 2011, pp. 1951–1956.

[6] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative uav formation flying with obstacle/collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 672–679, July 2007.

[7] D. Gu, "A differential game approach to formation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 85–93, Jan 2008.

[8] Y. Zhao, L. Jiao, R. Zhou, and J. Zhang, "Uav formation control with obstacle avoidance using improved artificial potential fields," in *2017 36th Chinese Control Conference (CCC)*, July 2017, pp. 6219–6224.

[9] D. Luo, T. Zhou, and S. Wu, "Obstacle avoidance and formation regrouping strategy and control for uav formation flight," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, June 2013, pp. 1921–1926.

[10] J. Seo, Y. Kim, S. Kim, and A. Tsourdos, "Collision avoidance strategies for unmanned aerial vehicles in formation flight," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2718–2734, Dec 2017.

[11] Yi Liang and Ho-Hoon Lee, "Decentralized formation control and

obstacle avoidance for multiple robots with nonholonomic constraints," in *2006 American Control Conference*, June 2006, pp. 6 pp.–.

[12] G. Takács and B. Rohal'-Ilkiv, *Stability and Feasibility of MPC*. London: Springer London, 2012, pp. 253–285. [Online]. Available: https://doi.org/10.1007/978-1-4471-2333-0_7

[13] H. CHEN and F. ALLGöWER, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stabilitythis paper was not presented at any ifac meeting. this paper was accepted for publication in revised form by associate editor w. bequette under the direction of editor prof. s. skogestad." *Automatica*, vol. 34, no. 10, pp. 1205 – 1217, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109898000739

[14] S. S. Keerthi and E. G. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *Journal of Optimization Theory and Applications*, vol. 57, no. 2, pp. 265–293, 1988. [Online]. Available: https://doi.org/10.1007/BF00938540

[15] Y. Zhu and U. Ozguner, "Robustness analysis on constrained model predictive control for nonholonomic vehicle regulation," in *2009 American Control Conference*, June 2009, pp. 3896–3901.

[16] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068 – 2085, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016003216304926