Safe Formation Control with Constrained Linear Model Predictive Control

Conference Paper · January 2019

DOI: 10.5220/0007918100700081

1 author:

Philippe Feyel
SAFRAN A.S. group
45 PUBLICATIONS **50** CITATIONS

SEE PROFILE

Safe Formation Control with Constrained Linear Model Predictive Control

Philippe Feyel
Safran Electronics & Defense, Massy, France
philippe.feyel@safrangroup.com

Keywords: Linear Model Predictive Control, Formation, Virtual Leader, Agents, Obstacles, Collision Avoidance, Swarm.

Abstract: This paper proposes a safe formation control technic based on the virtual leader-follower principle employed to build agents' trajectory. Each agent tracks its own trajectory using exclusively constrained linear Model Predictive Control (MPC). To ensure non-collision during formation control and transient phases such as formation reconfigurations, and thanks to the flexibility of MPC, a simple and flexible method for establishing collision and obstacle avoidance linear constraints is proposed. The efficiency of the approach is illustrated by the safe formation control and reconfiguration of a swarm of quadrotor helicopters.

1 INTRODUCTION

Since few years, Model Predictive Control (MPC) has become one of the most popular intelligent control technics due to several decisive features: its efficiency to take into account constraints in real time, its capability to control simply multivariable and nonlinear plants and its ability to be easily reconfigurable in case of fault detection. The reader should have a good introduction to this control technic in (Rawlings et al, 2017). Moreover some powerful works dealing with stability (Mayne et al, 2000), (Dutta, 2014) and robustness (Bahadir Saltik et al, 2018), (Mayne et al, 2007) have made MPC be one of the major ingredients in autonomous activities especially for guidance where volatile constraints (for instance depending on external environment such as collision and obstacle avoidance) have to be considered simultaneously with other more classical ones such as actuators limitations.

A safe formation control technic requires two essential ingredients:

- a formation control mechanism characterized by a nominal distance between agents,
- a collision and obstacle avoidance scheme enforcing a safety distance between agents and obstacles.

Indeed, using only a collision avoidance scheme doesn't prevent each agent from moving freely locally (while following globally the swarm trajectory) provided that distance between each agent is upper than a safety one. And similarly, using only a formation mechanism cannot prevent each agent from colliding to each other (for instance in case of formation reconfiguration or in presence of unpredicted disturbances). Of course the nominal distance has to be higher than the safety one.

Numerous works dealing with safe formation control using the MPC framework can be found in literature. Most of them are considering MPC in its non-linear formulation - denoted NMPC - mainly due to the complexity of the proposed obstacle avoidance and formation mechanisms (Garimella et al, 2017), (Xi et al, 2007), (Nikou et al, 2017), which makes their implementation difficult although several advances have been realized in embedded non-linear optimization since last few years (Gros et al, 2016), (Ferreau et al, 2017). That's why in this work only linear MPC will be considered due to its facility of implementation.

Formation control consists in modifying the reference trajectory of each agent depending on the required geometrical formation the swarm has to design or the behaviour it has to model. Thus in (Iskandarani et al, 2014), (Hafez et al, 2014) one finds works based on the leader-follower principle where the reference trajectory of each agent

(denoted as follower) is directly obtained from the trajectory realized by a leader, but locally modified by mean of relative bias describing the figure to be designed by the swarm. Each agent is guided using a linear MPC. In spite of its simplicity, the strategy becomes inefficient in case of a defective leader unless another agent becomes the leader with the problem of transient reconfiguration. However this technic allows a swarm to be guided by sending the trajectory only to the leader of the formation. Based on the same principle, the virtual leader-follower formation technic (Antonelli, 2013) designates the global planned swarm trajectory as the (virtual) leader. On the contrary to the precedent case, planned trajectory has to be sent to all agents but the stability of the formation doesn't depend on the state of the leader any more. Those two formation technics, although quite performing from the formation point of view, don't prevent agents from colliding with each other's in case of unpredicted disturbance or transient phases such as formation reconfiguration for instance, which justifies working on collision and obstacle avoidance schemes to ensure safety in formation control.

One can find several works dealing with obstacle and collision avoidance using the linear MPC framework. Some of them propose to modify the MPC cost with an additional term whose value is increasing while distance between agents and/or obstacles is decreasing (i.e. when the collision risk is increasing). Several papers succeed in expressing this additional term by using convex quadratic programming formalism typical of linear MPC problems (Jiang et al, 2016), (Gao et al, 2010), (Rasekhipour et al, 2017): the main drawback of such strategies deals with the tuning of the MPC cost which may appear difficult. Moreover, because the non-collision strategy is not formulated as constraints, one cannot certify if agents will collide with each other's or not; each agent can just try to avoid others depending on a trade-off between guidance and non-collision. To bypass those drawbacks, a common way is to work with obstacle avoidance constraint.

Writing collision avoidance constraints using linear formulations is a hard task due to non-convex property of a spatial area avoidance constraint. In (Boucek, 2016) one finds an UAV guidance approach with linear MPC accompanied by linear obstacle avoidance constraints. The main idea consists in defining a forbidden circular area around the obstacle to avoid and then finding the conditions preventing the guided UAV from moving inside the plane tangent to this area, belonging to the same side and normal to the motion. The drawback of this approach is that the obstacle avoidance constraint is 2-dimensional limited and its 3-dimensional extension is complex to formulate. Furthermore its flexibility is limited so that special avoidance rules can't be taken into account easily. In (Papen et al, 2017), (Richards et al, 2002), one can find other approaches using binary logical variables within Mixed-Integer Linear Programming (MILP) to encode non-convex avoidance constraints using linear formulation. Although efficient, those approaches are restricted to linear MPC criteria such as the Minimum Time Trajectory criterion, and are not adapted for more complex criteria such as the usual MPC quadratic criterion.

This paper is composed as follows. In section 2 we recall some features of stable linear MPC for reference tracking subject to constraints. A slightly focus is made on the MPC weights setting task and the quadratic programming formulation. In section 3, the formation control based on the virtual leader-follower principle is reminded and coupled with agents' MPC controllers, similar to the one presented in section 2 with slightly modifications related to safe formation control. Indeed, thanks to the MPC flexibility, we propose to ensure safety through a new simple and flexible linear formulation of collision avoidance constraints. Same strategy is retained for obstacle avoidance. Finally the section 4 proposes an illustrating example based on the safe formation control of a swarm of quadrotor helicopters with formation reconfigurations in presence of obstacles.

2 MPC FOR REFERENCE TRACKING

2.1 MPC Problem Formulation

At discrete-time *k*, we denote as $Y_k = \left(x_k, y_k, z_k\right)^T \in \Re^3$ the Cartesian coordinates of an agent modelled by the linear discrete state-space equation (1).

$$\begin{cases} X_{k+1} = AX_k + BU_k \\ Y_k = CX_k \end{cases} \quad (1)$$

For simplicity state-matrices are assumed to be time invariant. $X_k \in \Re^n$ and $U_k \in \Re^m$ denote respectively the state-vector and the input control vector of the agent. The state-vector X_k is assumed

to be fully measured or fully estimated. We denote as Y_k^{path} the planned trajectory to be tracked by the agent at time *k*. To this end, the agent is controlled using a linear model predictive controller which consists in on-line solving the constraint optimization problem (2) at each sampling time *k*. All predicted data are noted with a hat and *N* denotes the size of the prediction horizon. Due to actuators limitations, control signal is bounded. Note that using control variations formalism in the cost function aims to provide the controller with an integral action, allowing thus offset-free tracking (Maciejowski et al, 2002).

$$\begin{aligned}
&\min_{\Delta U=\left[\Delta u_0^T,\ldots,\Delta u_{N-1}^T\right]} J\left(X_k,U_{k-1}\right)\\
&\text{subject to}\\
&\begin{cases}
J\left(X_k,U_{k-1}\right) := \sum_{j=0}^{N-2}\hat{\varepsilon}_{j+1}^T Q\hat{\varepsilon}_{j+1}^T + \sum_{j=0}^{N-1}\Delta u_j^T R\Delta u_j + J_N\\
J_N := \hat{\varepsilon}_N^T P\hat{\varepsilon}_N\\
\hat{\varepsilon}_j = Y_{k+j}^{path} - \hat{Y}_j\\
\Delta u_j = u_j - u_{j-1},\ u_{-1} = U_{k-1}\\
\hat{X}_{j+1} = A\hat{X}_j + Bu_j,\ \hat{X}_0 = X_k\\
\hat{Y}_j = C\hat{X}_j\\
U^{\min} \le u_j \le U^{\max}
\end{cases}
\end{aligned} \tag{2}$$

Criterion matrices $Q \in \Re^{3\times 3}$ and $R \in \Re^{m\times m}$ are respectively semi-positive and positive definite; given the prediction horizon *N*, a difficulty of MPC controllers is related to *Q* and *R* settings depending on the trade-off between performance and robustness/energy. One can find several studies dealing with this problem (Garriga et al, 2010). An interesting approach coming from the Generalized Predictive Control (GPC) framework can be found in (Boucher et al, 1996) which we propose to extend to the MPC proposed here. *Q* and *R* are set once for all using the following rules providing a maximum phase margin and a good gain margin (in the case of a non-constrained problem).

$$\begin{aligned}
&Q = I_3,\ R = diag\left(r_1,\ldots,r_m\right)\\
&\text{for } p = 1:m, \text{ do}\\
&\quad u_p = \left[u_1 = 0,\ldots,u_p = 1,\ldots,u_m = 0\right]^T\\
&\quad \left.\begin{aligned}\hat{X}_{j+1} &= A\hat{X}_j + Bu_p\\ \hat{Y}_j &= C\hat{X}_j\end{aligned}\right\} j = 1,...,N,\ \hat{X}_1 = 0\\
&\quad r_p = \frac{1}{N}\sum_{j=1}^{N}\hat{Y}_j^T\hat{Y}_j\\
&\text{end for}
\end{aligned} \tag{3}$$

Note that the previous general rules can also be used in non-linear case.

For nominal close-loop stability (Mayne et al, 2000), a terminal quadratic cost has been defined with a positive definite matrix *P* solution of the Discrete Algebraic Riccati Equation (4).

$$\begin{aligned}
&\left(A+BK\right)^T P\left(A+BK\right) + K^T RK + Q = 0\\
&K = -\left(R + B^T PB\right)^{-1} B^T PA
\end{aligned} \tag{4}$$

Note that the state-feedback *K* can also be used to extend the MPC previous controller in the tube-based robust MPC framework (Mayne et al, 2007) to enhance performance robustness in case of plant's variations.

Once (2) is solved at each time *k* and according to the receding horizon principle, the control signal applied at time *k* to the agent is written in (5).

$$U_k = \left[I_m \underbrace{0_m \ldots 0_m}_{N-1 times}\right]\Delta U^* + U_{k-1} \tag{5}$$

where ΔU^* is the solution of problem (2).

2.2 Quadratic Programming (QP) Formulation

The QP formulation consists in writing problem (2) under the form (6) to be usable with commercial solvers such as the well-known *quadprog* routine (Wang, 2009).

$$\begin{aligned}
&\min_{\Delta U} \Delta U^T \Theta \Delta U + \theta\Delta U\\
&\text{subject to}\\
&A_{iq}\Delta U \le b_{iq}
\end{aligned} \tag{6}$$

2.2.1 Main Cost QP Formulation

It is straightforward to show that:

$$\begin{aligned}
\Delta\hat{X}_{j+1} &= \hat{X}_{j+1} - \hat{X}_j\\
&= A\Delta\hat{X}_j + B\Delta u_j
\end{aligned} \tag{7}$$

Moreover:

$$\begin{aligned}
\Delta\hat{Y}_{j+1} &= \hat{Y}_{j+1} - \hat{Y}_j\\
&= CA\Delta\hat{X}_j + CB\Delta u_j
\end{aligned} \tag{8}$$

Then:

$$\hat{Y}_{j+1} = \hat{Y}_j + CA\Delta\hat{X}_j + CB\Delta u_j \tag{9}$$

Collecting last equations into an augmented state-space, one obtains:

$$\begin{aligned}\tilde{X}_{j+1} &= \tilde{A}\tilde{X}_j + \tilde{B}\Delta u_j \\ \hat{Y}_j &= \tilde{C}\tilde{X}_j\end{aligned} \tag{10}$$

With:

$$\begin{aligned}\tilde{X}_j &= \begin{bmatrix}\Delta\hat{X}_j \\ \hat{Y}_j\end{bmatrix},\ \tilde{A} = \begin{bmatrix}A & 0 \\ CA & I_3\end{bmatrix} \\ \tilde{B} &= \begin{bmatrix}B \\ CB\end{bmatrix},\ \tilde{C} = \begin{bmatrix}0 & I_3\end{bmatrix}\end{aligned} \tag{11}$$

Assuming $\tilde{X}_0 = \begin{bmatrix}\Delta X_k{}^T & Y_k{}^T\end{bmatrix}^T$ is measured or estimated at time *k*, the state prediction along the horizon is recursively written as follows:

$$\begin{aligned}\tilde{X}_1 &= \tilde{A}\tilde{X}_0 + \tilde{B}\Delta u_0 \\ \tilde{X}_2 &= \tilde{A}^2\tilde{X}_0 + \tilde{A}\tilde{B}\Delta u_0 + \tilde{B}\Delta u_1 \\ &\vdots \\ \tilde{X}_N &= \tilde{A}^N\tilde{X}_0 + \tilde{A}^{N-1}\tilde{B}\Delta u_0 + \\ &\quad \tilde{A}^{N-2}\tilde{B}\Delta u_2 + \ldots + \tilde{B}\Delta u_{N-1}\end{aligned} \tag{12}$$

And then the output prediction:

$$\begin{aligned}\hat{Y}_1 &= \tilde{C}\tilde{A}\tilde{X}_0 + \tilde{C}\tilde{B}\Delta u_0 \\ \hat{Y}_2 &= \tilde{C}\tilde{A}^2\tilde{X}_0 + \tilde{C}\tilde{A}\tilde{B}\Delta u_0 + \tilde{C}\tilde{B}\Delta u_1 \\ &\vdots \\ \hat{Y}_N &= \tilde{C}\tilde{A}^N\tilde{X}_0 + \tilde{C}\tilde{A}^{N-1}\tilde{B}\Delta u_0 \\ &\quad + \tilde{C}\tilde{A}^{N-2}\tilde{B}\Delta u_2 + \ldots + \tilde{C}\tilde{B}\Delta u_{N-1}\end{aligned} \tag{13}$$

This can be written in the matrix form:

$$\begin{aligned}&\hat{Y} = W\tilde{X}_0 + Z\Delta U \\ &\hat{Y} = \begin{bmatrix}\hat{Y}_1 \\ \vdots \\ \hat{Y}_N\end{bmatrix},\ \Delta U = \begin{bmatrix}\Delta u_0 \\ \vdots \\ \Delta u_{N-1}\end{bmatrix},\ W = \begin{bmatrix}\tilde{C}\tilde{A} \\ \vdots \\ \tilde{C}\tilde{A}^N\end{bmatrix} \\ &Z = \begin{bmatrix}\tilde{C}\tilde{B} & & & \\ \tilde{C}\tilde{A}\tilde{B} & \tilde{C}\tilde{B} & & \\ \vdots & \vdots & \ddots & \\ \tilde{C}\tilde{A}^{N-2}\tilde{B} & \cdots & \cdots & \tilde{C}\tilde{B}\end{bmatrix}\end{aligned} \tag{14}$$

The MPC cost is then QP formulated as bellow:

$$\begin{aligned}&J := \left(Y^{path} - \hat{Y}\right)^T \mathbb{Q}\left(Y^{path} - \hat{Y}\right) + \Delta U^T \mathbb{R}\Delta U \\ &Y^{path} = \begin{bmatrix}Y_k^{path} \\ \vdots \\ Y_{k+N}^{path}\end{bmatrix},\ \mathbb{Q} = \begin{bmatrix}Q & & \\ & \ddots & \\ & & P\end{bmatrix},\ \mathbb{R} = \begin{bmatrix}R & & \\ & \ddots & \\ & & R\end{bmatrix}\end{aligned} \tag{15}$$

Developing (15), we arrive quickly at a quadratic expression with respect to ΔU :

$$\begin{aligned}&J := \frac{1}{2}\left[\varepsilon^T\mathbb{Q}\varepsilon - 2\varepsilon^T\mathbb{Q}Z\Delta U + \Delta U^T\left(Z^T\mathbb{Q}Z + \mathbb{R}\right)\Delta U\right] \\ &\varepsilon = Y^{path} - W\tilde{X}_0\end{aligned} \tag{16}$$

Note that minimizing *J* or J_2 (17) with respect to ΔU is equivalent.

$$\begin{aligned}&J_2 := \Delta U^T\left(Z^T\mathbb{Q}Z + \mathbb{R}\right)\Delta U - 2\varepsilon^T\mathbb{Q}Z\Delta U \\ &\varepsilon = Y^{path} - W\tilde{X}_0\end{aligned} \tag{17}$$

2.2.2 Control Constraints QP Formulation

According to definition (14) and initializing the recursion with $u_{-1} = U_{k-1}$, it is easy to show that:

$$\begin{aligned}u_0 &= U_{k-1} + \Delta u_0 = U_{k-1} + \begin{bmatrix}I_m & 0_m & \ldots & 0_m\end{bmatrix}\Delta U \\ u_1 &= u_0 + \Delta u_1 = U_{k-1} + \Delta u_0 + \Delta u_1 \\ &= U_{k-1} + \begin{bmatrix}I_m & I_m & \ldots & 0_m\end{bmatrix}\Delta U \\ &\vdots\end{aligned} \tag{18}$$

Or equivalently:

$$\begin{aligned}&U = EU_{k-1} + H\Delta U \\ &E = \begin{bmatrix}I_m \\ I_m \\ \vdots \\ I_m\end{bmatrix},\ H = \begin{bmatrix}I_m & 0_m & \cdots & 0_m \\ I_m & I_m & \ddots & 0_m \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \cdots & I_m\end{bmatrix},\ U = \begin{bmatrix}u_0 \\ u_1 \\ \vdots \\ u_{N-1}\end{bmatrix}\end{aligned} \tag{19}$$

Thus the control constraint is written with respect to ΔU :

$$\begin{aligned}&\begin{bmatrix}-H \\ H\end{bmatrix}\Delta U \le \begin{bmatrix}-\underline{U} + EU_{k-1} \\ \overline{U} - EU_{k-1}\end{bmatrix} \\ &\underline{U} = \begin{bmatrix}U^{\min} \\ \vdots \\ U^{\min}\end{bmatrix},\ \overline{U} = \begin{bmatrix}U^{\max} \\ \vdots \\ U^{\max}\end{bmatrix}\end{aligned} \tag{20}$$

3 SAFE FORMATION CONTROL

Now we consider a swarm of *M* agents. The global planned trajectory of the swarm is the Y^{path} defined previously. At time *k*, we denote as $Y_k^i = \left(x_k^i, y_k^i, z_k^i\right)^T \in \Re^3$ the Cartesian coordinates of the i^{th} agent of the swarm. Each agent is controlled using the linear MPC controller introduced in section 2 but with slightly modifications dealing with formation control and additional collision avoidance constraints which we describe below.

3.1 Formation Control based on the Virtual Leader-follower Principle

According to the virtual-leader follower principle (Antonelli, 2013), the i^{th} agent follows its own trajectory obtained by δ^i-shifting the swarm trajectory. Thus Y^{path} stands for the virtual leader trajectory and each agent stands for a follower. The i^{th} agent is controlled by the MPC (2) slightly modified by substitution (21).

$$Y_{k+j}^{path} \to Y_{k+j}^{path} - \delta_{k+j}^{i},\ j=1,\ldots,N \tag{21}$$

The M bias vectors $\delta_k^i \in \Re^3$, $i=1,\ldots,M$ are set according to the formation agents have to design and/or the behaviour the swarm has to model. Although the bias may be time-dependent, it will be considered (for practical reasons) constant throughout the prediction, i.e.:

$$Y_{k+j}^{path} \to Y_{k+j}^{path} - \delta_{k}^{i},\ j=1,\ldots,N \tag{22}$$

Basically the bias shifting main idea consists in designing the formation structure by specifying directly agents' coordinates inside the swarm in relation to the virtual leader such as in (Iskandarani et al, 2014), (Hafez et al, 2014). A trivial line formation example is depicted in Figure 1 in which the coordinates of agents are chosen so that nominal distance between each of them is upper than a specified one d_{nom}. Here the agent n°1 is arbitrary chosen as the virtual leader.

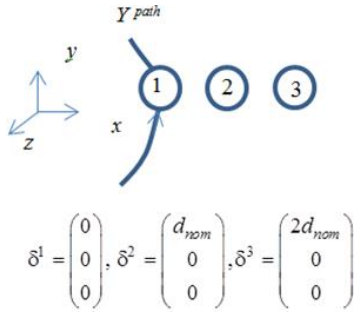

Figure 1: A basic line formation strategy.

The reader will find other more complex formation methods in literature for computing suitable δ_k^i, $i=1,\ldots,M$ that can be directly employed in the virtual leader-follower approach proposed here. Depending on the behaviour the swarm has to model, one can cite artificial potential field methods (De Vries et al, 2011), elliptical formation structures (Kahn et al, 2013), swarm intelligence methods or flocking-based control (Soni et al, 2018).

3.2 Safe Formation Control with Collision and Obstacle Avoidance Linear Constraints

For a safe formation control, thanks to the MPC flexibility, we propose a simple and flexible linear obstacle avoidance constraints formulation in this subsection. For that purpose we model the (firstly static) object to avoid as a classical rectangular parallelepiped with edges denoted as $\left(d_x, d_y, d_z\right)$ and centered at point $Y^o = \left(x^o, y^o, z^o\right)^T \in \Re^3$. To avoid collision between the object and the i^{th} agent (i.e. to avoid i^{th} agent entering inside the object) constraints (23) have to be verified at time k.

$$\begin{gathered}\left[\left(x_k^i > \frac{d_x}{2} + x^o\right) \text{ or } \left(x_k^i < x^o - \frac{d_x}{2}\right)\right] \\ \text{and} \\ \left[\left(y_k^i > \frac{d_y}{2} + y^o\right) \text{ or } \left(y_k^i < y^o - \frac{d_y}{2}\right)\right] \\ \text{and} \\ \left[\left(z_k^i > \frac{d_z}{2} + z^o\right) \text{ or } \left(z_k^i < z^o - \frac{d_z}{2}\right)\right]\end{gathered} \tag{23}$$

For readability, we denote the relative coordinates of the 6 corners as $\left(\overline{x}^o, \underline{x}^o, \overline{y}^o, \underline{y}^o, \overline{z}^o, \underline{z}^o\right)$ so that constraints (23) can be rewritten as (24).

$$\begin{gathered}\left[\left(x_k^i > \overline{x}^o\right) \text{ or } \left(x_k^i < \underline{x}^o\right)\right] \\ \text{and} \\ \left[\left(y_k^i > \overline{y}^o\right) \text{ or } \left(y_k^i < \underline{y}^o\right)\right] \\ \text{and} \\ \left[\left(z_k^i > \overline{z}^o\right) \text{ or } \left(z_k^i < \underline{z}^o\right)\right]\end{gathered} \tag{24}$$

In the MPC framework, constraints (24) have to be enforced all along the prediction horizon, which leads to the non-convex $3\times N$ exclusion constraints (25). Hats stand for predictions.

$$\left.\begin{gathered}\left[\left(\hat{x}_j^i > \overline{x}^o\right) \text{ or } \left(\hat{x}_j^i < \underline{x}^o\right)\right] \\ \text{and} \\ \left[\left(\hat{y}_j^i > \overline{y}^o\right) \text{ or } \left(\hat{y}_j^i < \underline{y}^o\right)\right] \\ \text{and} \\ \left[\left(\hat{z}_j^i > \overline{z}^o\right) \text{ or } \left(\hat{z}_j^i < \underline{z}^o\right)\right]\end{gathered}\right\} j=1,\ldots,N \tag{25}$$

To transform (25) into convex and linear constraints, a simple idea is to force getting out the

object when the prediction plans to get inside it. The great advantage of this approach lies in its flexibility because it becomes easy to choose the best rules to get around the object (with a high-level heuristic for instance). Thus the algorithm computing the collision avoidance constraints will have the form (26).

The bypass strategy has to be chosen. Below we give an example of arbitrary rules that can be translated into linear collision avoidance constraints. For instance one can decide to avoid the object by using at first the (x, y) plan and forcing the prediction to be on the x-axis ($\bar{x}^o$ or $\underline{x}^o$) where it tends to establish (see Figure 2).

When the prediction plans to get inside the object and directly on the x^o -axis, the previous rule leads to an indeterminate; thus one can decide for instance to bypass selecting the y-axis ($\bar{y}^o$ or $\underline{y}^o$) using the same rules as for the previous x-axis as depicted in Figure 3.

$$
\left.
\begin{array}{l}
\text{if } \left(\hat{x}_j^i > \bar{x}^o\right) \text{ and} \left(\hat{x}_j^i < \underline{x}^o\right) \\
\quad \text{if } \left(\hat{y}_j^i > \bar{y}^o\right) \text{ and } \left(\hat{y}_j^i < \underline{y}^o\right) \\
\quad\quad \text{if } \left(\hat{z}_j^i > \bar{z}^o\right) \text{ and } \left(\hat{z}_j^i < \underline{z}^o\right) \\
\quad\quad\quad \Rightarrow \text{bypass strategy} \\
\quad\quad \text{end} \\
\quad \text{end} \\
\text{end}
\end{array}
\right\} j = 1, \ldots, N \qquad (26)
$$

Now if the prediction plans to get inside the object and directly at point (x^o, y^o) of the (x, y) plan, one can decide to bypass choosing a z-axis using the same rules as for the y and x-axis before. And finally if the prediction coincide with the object center (x^o, y^o, z^o), we can decide to bypass using the z-axis direction from above for instance.

Note that the previous rules are an arbitrary choice. Nothing prevents the reader from enforcing additional state constraints or changing bypass rules. This flexibility is clearly an advantage of the method.

At prediction time j, the previous rules can be translated into a suitable "mathematical formulation" using the algorithm (27). The corrections of the predicted positions stand all along the prediction horizon, i.e. for $j = 1, \ldots, N$.

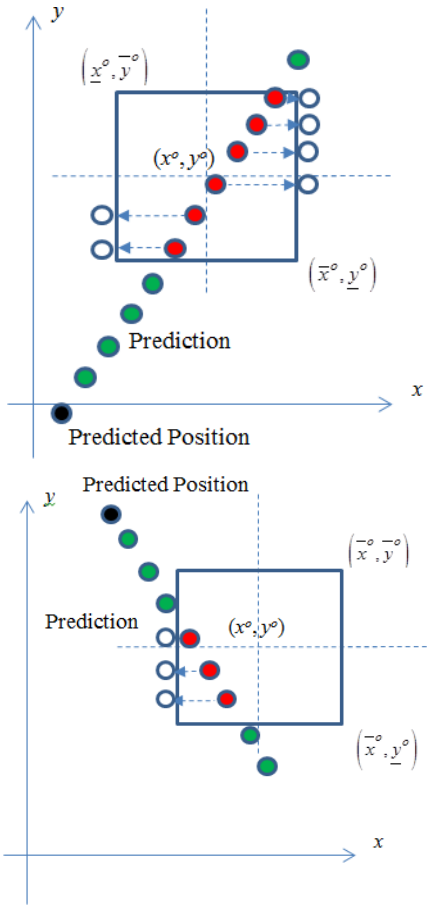

Figure 2: A (x, y) bypass strategy.

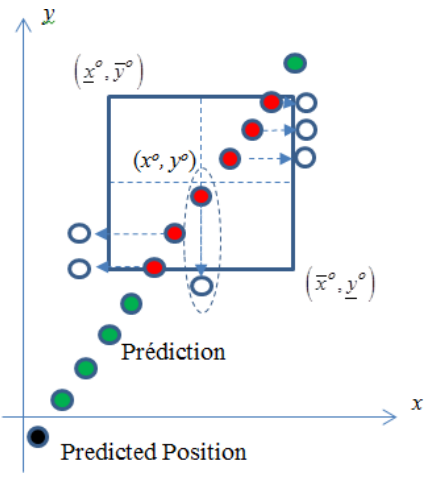

Figure 3: A (x, y) bypass strategy for indeterminate x-side.

if $\hat{x}_j^i > \underline{x}^o$ and $\hat{x}_j^i < \overline{x}^o$

if $\hat{y}_j^i > \underline{y}^o$ and $\hat{y}_j^i < \overline{y}^o$

if $\hat{z}_j^i > \underline{z}^o$ and $\hat{z}_j^i < \overline{z}^o$

if $\left|\hat{x}_j^i - \underline{x}^o\right| < \left|\hat{x}_j^i - \overline{x}^o\right|$, $\hat{x}_j^i \le \underline{x}^o$

else if $\left|\hat{x}_j^i - \underline{x}^o\right| > \left|\hat{x}_j^i - \overline{x}^o\right|$, $\hat{x}_j^i \ge \overline{x}^o$

else

if $\left|\hat{y}_i^j - \underline{y}^o\right| < \left|\hat{y}_i^j - \overline{y}^o\right|, \hat{y}_i^j \le \underline{y}^o$

else if $\left|\hat{y}_i^j - \underline{y}^o\right| > \left|\hat{y}_i^j - \overline{y}^o\right|$, $\hat{y}_i^j \ge \overline{y}^o$ (27)

else

if $\left|\hat{z}_i^j - \underline{z}^o\right| < \left|\hat{z}_i^j - \overline{z}^o\right|$, $\hat{z}_i^j \le \underline{z}^o$

else $\hat{z}_i^j \ge \overline{z}^o$

end if

end if

end if

end if

end if

We recall the following equivalences:

$$\left.\begin{aligned}
&\hat{x}_j^i \le \underline{x}^o \Leftrightarrow \hat{x}_j^i \le x^o - \frac{d_x}{2} \\
&\hat{x}_j^i \ge \overline{x}^o \Leftrightarrow -\hat{x}_j^i \le -x^o - \frac{d_x}{2} \\
&\hat{y}_i^j \le \underline{y}^o \Leftrightarrow \hat{y}_i^j \le y^o - \frac{d_y}{2} \\
&\hat{y}_i^j \ge \overline{y}^o \Leftrightarrow -\hat{y}_i^j \le -y^o - \frac{d_y}{2} \\
&\hat{z}_i^j \le \underline{z}^o \Leftrightarrow \hat{z}_i^j \le z^o - \frac{d_z}{2} \\
&\hat{z}_i^j \ge \overline{z}^o \Leftrightarrow -\hat{z}_i^j \le -z^o - \frac{d_z}{2}
\end{aligned}\right\} \quad (28)$$

Now by introducing three integers $\left(\eta_j^x, \eta_j^y, \eta_j^z\right)$ that can only taking values 0 or ±1, it is straightforward to see that algorithm (27) and linear obstacle avoidance constraints (29) are strictly equivalent at the j^{th} prediction time.

$$\eta_j \hat{Y}_j^i \le \eta_j Y^o - L_o$$

$$\eta_j = diag\left(\eta_j^x, \eta_j^y, \eta_j^z\right)$$

$$L_o = \left(\frac{d_x}{2} \quad \frac{d_y}{2} \quad \frac{d_z}{2}\right)^T$$

with:

$\eta_j^x = \eta_j^y = \eta_j^z = 0$

if $\hat{x}_j^i > \underline{x}^o$ and $\hat{x}_j^i < \overline{x}^o$

if $\hat{y}_j^i > \underline{y}^o$ and $\hat{y}_j^i < \overline{y}^o$

if $\hat{z}_j^i > \underline{z}^o$ and $\hat{z}_j^i < \overline{z}^o$

if $\left|\hat{x}_j^i - \underline{x}^o\right| < \left|\hat{x}_j^i - \overline{x}^o\right|$, $\eta_j^x = 1$

else if $\left|\hat{x}_j^i - \underline{x}^o\right| > \left|\hat{x}_j^i - \overline{x}^o\right|$, $\eta_j^x = -1$ (29)

else

if $\left|\hat{y}_i^j - \underline{y}^o\right| < \left|\hat{y}_i^j - \overline{y}^o\right|$, $\eta_j^y = 1$

else if $\left|\hat{y}_i^j - \underline{y}^o\right| > \left|\hat{y}_i^j - \overline{y}^o\right|$, $\eta_j^y = -1$

else

if $\left|\hat{z}_i^j - \underline{z}^o\right| < \left|\hat{z}_i^j - \overline{z}^o\right|$, $\eta_j^z = 1$

else $\eta_j^z = -1$

end if

end if

end if

end if

end if

Note that in case of inactive constraints (or equivalently in case of non-collision risk) at prediction time *j*, at least one of coefficients of matrix η_j is zero; in this case, the problem may become unfeasible (by nature) because:

$$\left.\begin{aligned}
&\eta_j \hat{Y}_j^i \le \eta_j Y^o - L_o \\
&\eta_j = diag\left(0,0,0\right)
\end{aligned}\right\} \Rightarrow 0 \le -L_o \Rightarrow \text{unfeasability} \quad (30)$$

Thus the previous constraints formulation has to be modified to avoid unfeasibility in case of non-collision risk. A solution is to introduce an additional matrix η_j' such that $\eta_j' = abs(\eta_j)$. Thus taking (30) again:

$$\left.\begin{aligned}
&\eta_j \hat{Y}_j^i \le \eta_j Y^o - \eta_j' L_o \\
&\eta_j = diag\left(0,0,0\right) \\
&\eta_j' = diag\left(0,0,0\right)
\end{aligned}\right\} \Rightarrow 0 \le 0 \Rightarrow \text{not unfeasable} \quad (31)$$

Finally at prediction time *j*, the non-collision constraint is written:

$$\left.\begin{aligned} &\eta_j \hat{Y}_j^i \le \eta_j Y^o - \eta_j' L_o \\ &\eta_j = diag\left(\eta_j^x, \eta_j^y, \eta_j^z\right) \\ &L_o = \begin{pmatrix} \frac{d_x}{2} & \frac{d_y}{2} & \frac{d_z}{2} \end{pmatrix}^T \\ &\eta_j' = abs\left(\eta_j\right) \end{aligned}\right\} \quad (32)$$

Where $\left(\eta_j^x, \eta_j^y, \eta_j^z\right)$ are always defined by (29).

Another difficulty of the method is the use of the predicted state to predict the collision, but the predicted state is determined by the control sequence one has to determine with the non-collision constraints we search to determine… To overcome this difficulty, an estimation of the predicted state can be used. Several methods can be envisaged:

- Using the previous optimal sequence (i.e. at time *k*-1) one step-shifted as a good estimation of the predicted position at time *k*.
- Assuming the measurement of the position and the velocity at time *k*, an estimation of the predicted position can be built with a linear extrapolation along the prediction (33).

$$\hat{x}_j^i = x_k^i + jT_e\dot{x}_k^i,\ \hat{y}_j^i = y_k^i + jT_e\dot{y}_k^i,\ \hat{z}_j^i = z_k^i + jT_e\dot{z}_k^i \quad (33)$$

Where T_e is the sampling period. Same rule can be considered for estimate the future positions of a dynamical object one has to avoid, assuming its position and velocity are measured at time *k* (34).

$$\hat{x}_j^o = x_k^o + jT_e\dot{x}_k^o,\ \hat{y}_j^o = y_k^o + jT_e\dot{y}_k^o,\ \hat{z}_j^o = z_k^o + jT_e\dot{z}_k^o \quad (34)$$

In that case, constraints (32) can be easily extended to a dynamical object (35).

$$\left.\begin{aligned} &\eta_j \hat{Y}_j^i \le \eta_j \hat{Y}_j^o - \eta_j' L_o \\ &\eta_j = diag\left(\eta_j^x, \eta_j^y, \eta_j^z\right) \\ &L_o = \begin{pmatrix} \frac{d_x}{2} & \frac{d_y}{2} & \frac{d_z}{2} \end{pmatrix}^T \\ &\eta_j' = abs\left(\eta_j\right) \end{aligned}\right\} \quad (35)$$

Now we are able to formulate the non-collision constraints all along the prediction (36).

$$\begin{aligned} &\eta_1 \hat{Y}_1^i \le \eta_1 \hat{Y}_1^o - \eta_1' L_o \\ &\eta_2 \hat{Y}_2^i \le \eta_2 \hat{Y}_2^o - \eta_2' L_o \\ &\vdots \\ &\eta_N \hat{Y}_N^i \le \eta_N \hat{Y}_N^o - \eta_N' L_o \end{aligned} \quad (36)$$

Or equivalently in matrix form:

$$\begin{aligned} &\Pi_{io}\hat{Y}^i \le \Pi_{io}\hat{Y}^o - \mathrm{H}_{io} \\ &\Pi_{io} = \begin{pmatrix} \eta_1 & & \\ & \ddots & \\ & & \eta_N \end{pmatrix},\ \mathrm{H}_{io} = \begin{pmatrix} \eta_1' & & \\ & \ddots & \\ & & \eta_N' \end{pmatrix}\begin{pmatrix} L_o \\ \vdots \\ L_o \end{pmatrix} \\ &\hat{Y}^o = \begin{pmatrix} \hat{Y}_1^o \\ \vdots \\ \hat{Y}_N^o \end{pmatrix} \end{aligned} \quad (37)$$

According to (14) one obtains the linear non-collision constraints formulation (38) with respect to control variations ΔU^i that we need to incorporate to the MPC (17)-(20) we have to solve for the i^{th} agent in order it to avoid the o^{th} object.

$$\Pi_{io}Z_i\Delta U^i + \Pi_{io}W_i\tilde{X}_k^i - \Pi_{io}\hat{Y}^o + \mathrm{H}_{io} \le 0 \quad (38)$$

Constraints (38) can be easily extended to several static or dynamic objects (other agents for instance) that i^{th} agent has to avoid.

4 SAFE FORMATION CONTROL OF A SWARM OF QUADROTOR HELICOPTERS

4.1 Safe Formation Problem Statement

In the following, all coordinates and distances are expressed in meters.

To illustrate the previous developments, one considers the safe guidance of a swarm of 5 quadrotor helicopters.

Agents are assumed to be identical and modelled by (10×10×10) cubes. Dynamic model will be considered in the next subsection. The planned swarm trajectory Y^{path} is helical type and is generated by equations (39).

$$\begin{aligned} &Y_{t\le 100}^{path} = \begin{pmatrix} 50\cos\left(\frac{2\pi}{40}t\right) & 50\sin\left(\frac{2\pi}{40}t\right) & 10t \end{pmatrix}^T \\ &Y_{t>100}^{path} = \begin{pmatrix} 10(t-100) & 10(t-100) & 1000 \end{pmatrix}^T \\ &Y_0^{path} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \end{aligned} \quad (39)$$

The initial formation is line type and is defined by the five shifting bias vectors given in (40).

$$\delta^1 = \begin{pmatrix} -40 \\ 0 \\ 0 \end{pmatrix},\ \delta^2 = \begin{pmatrix} -20 \\ 0 \\ 0 \end{pmatrix},\ \delta^3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},\ \delta^4 = \begin{pmatrix} 20 \\ 0 \\ 0 \end{pmatrix},\ \delta^5 = \begin{pmatrix} 40 \\ 0 \\ 0 \end{pmatrix} \quad (40)$$

At time t_c = 40s, formation changes into (41).

$$\delta^1 = \begin{pmatrix} -20 \\ 0 \\ 0 \end{pmatrix}, \delta^2 = \begin{pmatrix} -40 \\ 0 \\ 0 \end{pmatrix}, \delta^3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \delta^4 = \begin{pmatrix} 40 \\ 0 \\ 0 \end{pmatrix}, \delta^5 = \begin{pmatrix} 20 \\ 0 \\ 0 \end{pmatrix} \quad (41)$$

We can see that two neighboring agents are 20 meters apart.

Initial positions of the 5 UAVs are given below:

$$\begin{aligned} &Y_0^1 = (-50 \quad -50 \quad 0)^T, \; Y_0^2 = (100 \quad 100 \quad 0)^T \\ &Y_0^3 = (-110 \quad -110 \quad 0)^T, \; Y_0^4 = (25 \quad -150 \quad 0)^T \\ &Y_0^5 = (25 \quad 150 \quad 0)^T \end{aligned} \quad (42)$$

Finally, 2 static obstacles modelled as (10×10×10) cubes are positioned at points $(-65 \quad -40 \quad 255)^T$ and $(-65 \quad -45 \quad 270)^T$.

Each agent is controlled with the linear MPC presented in the previous section, using virtual leader-follower formation control with obstacle and collision avoidance constraints. The dynamic model of each agent is considered below.

4.2 Quadrotor Helicopter Model

There are lots of literatures describing quadrotor helicopters non-linear models (Mahony et al, 2012), (Bouabdallah et al, 2014). In order to obtain a linear plant compatible with (1), the following assumptions are considered:

- Quadrotor helicopters are employed in hovering conditions: this assumption is often realized for model linearization.
- A high dynamic inner-loop attitude control exists and its bandwidth is high enough to be considered as ideal: this assumption is done only for readability and simplicity.
- The yaw angle ψ is maintained to 0. This assumption is done because the quadrotor motion doesn't depend on ψ.

Thus equations of motion governing dynamics of a quadrotor helicopter with respect to an earth-fixed coordinate system are reduced to the linear translation terms (43), where:

- x, y, z stand for the Cartesian coordinates of the helicopter,
- φ and θ are respectively the desired roll and pitch angles,
- m = 0,486 kg is the quadrotor helicopter mass,
- T stands for the thrust applied to the helicopter,
- K_{fx} = K_{fy} = 5,567e-4 Nm/m/s and K_{fz} = 6,354e-4 Nm/m/s represent aerodynamic frictions due to motion along x, y, and z axis.
- g = 9,81 m.s^{-2} is the gravity coefficient.

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \theta\frac{g}{m} - \frac{K_{fx}}{m}\dot{x} \\ -\varphi\frac{g}{m} - \frac{K_{fy}}{m}\dot{y} \\ \frac{T}{m} - g - \frac{K_{fz}}{m}\dot{z} \end{pmatrix} \quad (43)$$

(43) can be easily expressed in state-space matrix form:

$$\begin{aligned} \frac{dX}{dt} &= AX + BU \\ Y &= CX \end{aligned} \quad (44)$$

With:

$$X = (x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z})^T$$

$$Y = (x \quad y \quad z)^T, \; U = (\theta \quad \varphi \quad \tilde{T})^T, \; \tilde{T} = T - mg$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{K_{fx}}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_{fy}}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{K_{fz}}{m} \end{pmatrix} \quad (45)$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{g}{m} & 0 & 0 \\ 0 & -\frac{g}{m} & 0 \\ \frac{1}{m} & 0 & 0 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Thruster, pitch and roll angles stand for the control signal vector. Position vector Y is assumed to be measured.

Finally the discrete model (46) is computed using Euler approximation with sampling time T_e = 0,1s.

$$\begin{aligned} X_{k+1} &= (T_e A + I) X_k + T_e B U_k \\ Y_k &= C X_k \end{aligned} \quad (46)$$

Due to actuators limitations and capabilities of quadrotor helicopters, control constraints have to be considered as follows:

$$\begin{pmatrix}\theta_{min}\\ \varphi_{min}\\ \tilde{T}_{min}\end{pmatrix} \le \begin{pmatrix}\theta_{k}\\ \varphi_{k}\\ \tilde{T}_{k}\end{pmatrix} \le \begin{pmatrix}\theta_{max}\\ \varphi_{max}\\ \tilde{T}_{max}\end{pmatrix} \Leftrightarrow U^{min} \le U_k \le U^{max}$$

$$\begin{aligned}&\theta_{max} = -\theta_{min} = \pi/4\\ &\varphi_{min} = -\varphi_{min} = \pi/4\\ &\tilde{T}_{min} = T_{min} - mg,\ \tilde{T}_{max} = T_{max} - mg\\ &T_{min} = 0,\ T_{max} = 4mg\end{aligned} \quad (47)$$

Note that U_k is computed by (5) and the thrust control signal really applied to the system is written:

$$T_k = \tilde{T}_k + mg \quad (48)$$

4.3 MPC Configuration

Each UAV is controlled with a linear MPC having the following characteristics:

- MPC horizon size: N = 100.
- Interior-Point quadratic optimization method with 30 iterations at most and initialization using the warm start method.
- Sampling period: T_e = 0,1 s.
- MPC cost weights are set according to (3).

4.4 Formation Control Results

First we proceed to a simulation realized without collision avoidance constraints. As we can see in Figure 4 formation control based on virtual leader-follower principle is efficient but collisions between agents occur during the swarm initialization and the reconfiguration phases. Furthermore, agents don't avoid static obstacles. As we can see in Figure 5, all control constraints are satisfied.

Secondly we proceed to the same scenario using the collision and obstacle avoidance linear constraints presented in this work. Result is depicted in Figure 6. As we can see, formation control becomes safe. Control signals are depicted in Figure 7. Collision avoidance constraints become active when the distance agents/agents and/or agents/obstacles tends to be lower than 10m which makes no collision occurs: it clearly proves the efficiency of our approach.

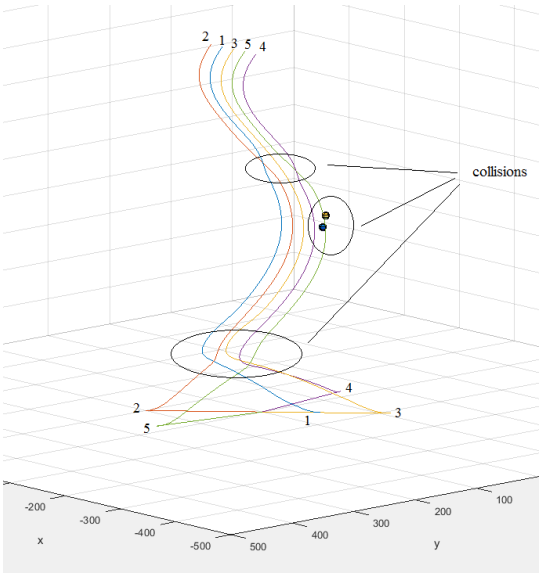

Figure 4: Formation control without collision avoidance constraints.

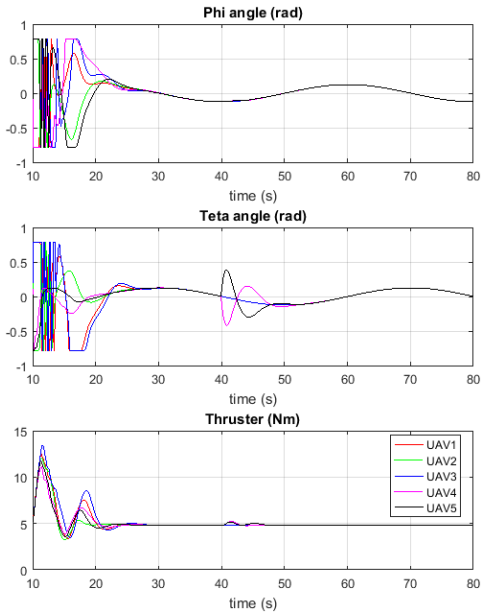

Figure 5: Control signals of UAVs without collision avoidance constraints.

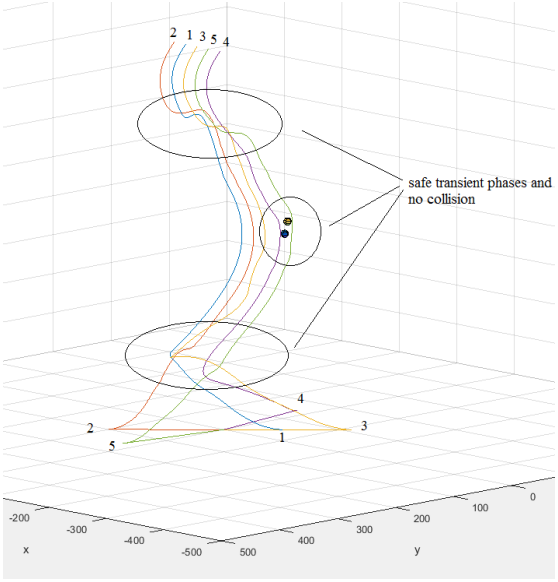

Figure 6: Formation control with collision avoidance constraints.

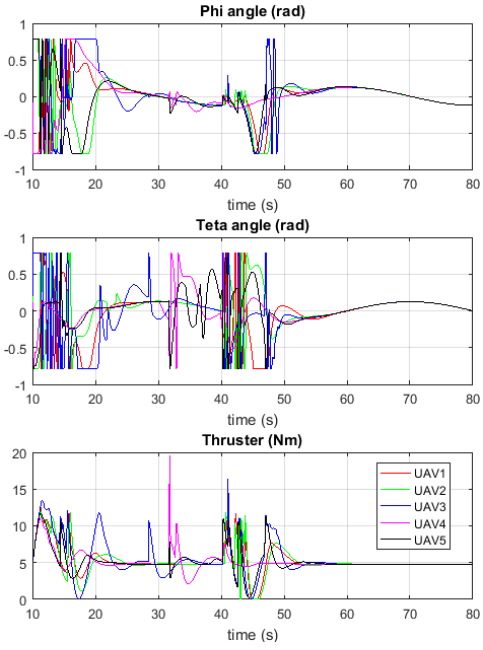

Figure 7: Control signals of UAVs with collision avoidance constraints.

6 CONCLUSIONS

We developed in this paper an efficient safe formation control technic based on the virtual-leader follower principle in association with linear MPC whose flexibility allows adding collision avoidance linear constraints that have been formulated towards an original and flexible approach. The efficiency of this approach has been illustrated with the formation control and reconfiguration of a swarm of UAVs. Future works deal with heterogeneous multi-agents swarming activities.

REFERENCES

Rawlings, J. B., Mayne, D. Q., Diehl, M. M., 2017. Model predictive control, Computation and Design. *Nob Hill Publishing*, Madison, Wisconsin.

Mayne, D.Q, Rawlings, J. B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained Model Predictive Control: Stability and Optimality. In: *Automatica*, Vol 36, pp789-814.

Dutta, A. 2014. Design and Certification of Industrial Predictive Controllers. *PhD Thesis*, Universiteit Gent.

Bahadir Saltik, M., Ozkan, L., Ludlage, J. H. A., Weiland, S., Van den Hof, P. M. J., 2018. An Outlook on Robust Model Predictive Control Algorithms: reflections on Performance and Computational Aspects. In: *Journal of Process Control*, Vol 61, pp77-102.

Mayne, D.Q, Kerrigan E.C., 2007. Tube-based Robust Nonlinear Model Predictive Control. In: *7th IFAC Symposium on Nonlinear Control Systems*, Pretoria, South Africa, 21-24 August 2007.

Garimella, G., Sheckells, M., Kobilarov, M., 2017. Robust Obstacle Avoidance for Aerial Platforms using Adaptive Model Predictive Control. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May-3 June 2017, Singapore, Singapore.

Xi, W., Baras, S., 2007. MPC Based Motion Control of Car-like Vehicle Swarms. In: *2007 Mediterranean Conference on Control and Automation*, July 27-29, Athens, Greece.

Nikou, A., Verginis, C., Heshmati-Alamdari, S., Dimarogonas, D.V., 2017. A Nonlinear Model Predictive Control Scheme for Cooperative Manipulation with Singularity and Collision Avoidance. In: *2017 25th Mediterranean Conference on Control and Automation (MED)*, 3-6 July 2017, Valletta, Malta.

Kamel, M., Alonso-Moray, J., Siegwart, R., Nieto, J., 2017. Robust Collision Avoidance for Multiple Micro Aerial Vehicles Using Nonlinear Model Predictive Control. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24-28 Sept. 2017, Vancouver, BC, Canada

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., Diehl, M., 2016. From linear to nonlinear MPC: bridging the gap via the real tine iteration. In: *International Journal of Control*, Sept 2016.

Ferreau, H.J., Almer, S., Verschueren, R., Diehl, M., Frick, D., Domahidi, A., 2017. Embedded Optimization Methods for Industrial Automatic

Control. In: *20th IFAC World Congress*, July 9-14 2017, Toulouse, France, pp 13736-13751.

Jiang, H., Wang, Z., Chen, Q., Zhu, J., 2016. Obstacle avoidance of autonomous vehicles with CQP-based model predictive control. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 9-12 Oct. 2016, Budapest, Hungary.

Gao, Y., Lin T., Borrelli, F., Tseng, E., Hrovat, D., 2010. Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads. In: *ASME 2010 Dynamic Systems and Control Conference*, Vol. 1, Cambridge, Massachusetts, USA, September 12–15 2010, pp. 265-272

Rasekhipour, Y., Khajepour, A., Chen, S.-K., Litkouhi. 2017. A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles. In: *IEEE Transactions on Intelligent Transportation Systems*, Vol 18, Issue 5, May 2017, pp 1255 – 1267.

Boucek, J., 2016. Model Predictive Control of Unmanned Helicopter with Obstacle Avoidance. *Bachelord's Thesis*, Czech Technical University In Prague, Faculty of Electrical Engineering, Prague, May 2016.

Papen, A., Vandenhoeck, R., Bolting, J., Defay, F., 2017. Collision-Free Rendezvous Maneuvers for Formations of Unmanned Aerial Vehicles. In: IFAC-PapersOnLine, vol. 50 (n° 1). pp. 282-289.

Richards, A., How, J.P., 2002. Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming In: *Proceedings of the 2002 IEEE American Control Conference*, 8-10 May 2002, Anchorage, AK, USA, USA.

Iskandarani, M., Givigi, S. N., Fusina, G., Beaulieu, A., 2014. Unmanned Aerial Vehicle formation flying using Linear Model Predictive control. In: *2014 IEEE International Systems Conference Proceedings*, 31 March-3 April 2014, Ottawa, ON, Canada.

Hafez, A. T., Iskandarani, M., Givigi, S. N., Youssefi, S., Beaulieu, A., 2014. UAVs in formation and Dynamic Encirclement via Model Predictive Control. In: *19th IFAC World Congress*, Cape Town, South Africa, August 24-29 2014.

Maciejowski, J.M., 2002. Predictive Control with Constraints. *Prentice Hall*, Harlow, England.

Garriga, J. L., Soroush, M., 2010. Model predictive tuning methods: a review. In: *Industrial & Engineering Chemistry Research*, Vol 49, pp 3505–3515

Boucher, P., Dumur, D., 1996. La commande prédictive. *Ophrys Editions*.

Wang, L., 2009. Model Predictive Control Design with Matlab. *Springer-Verlag*, London.

Antonelli, G., 2013. Interconnected Dynamic Systems, an overview on distributed systems. In: *IEEE Control systems magazine*, Fev 2013, pp76-88.

De Vries, E., Subbarao, K., 2011. Cooperative Control of Swarms of Unmanned Aerial Vehicles. In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 4 - 7 January 2011, Orlando, Florida.

Kahn, A., Marzat, J., Piet-Lahanier, H., 2013. Formation Flying Control via Elliptical Virtual Structure. In: *10th IEEE International Conference on Networking, sensing and Control (ICNSC)*, 10-12 April 2013, Evry, France.

Soni, A., Hu, H., 2018. Formation Control for a Fleet of Autonomous Ground Vehicles: A Survey. In: *In robotics*, Vol 7, n°67, MDPI Editions.

Mahony, R., Kumar, V., Corke,P., 2012. Multirotor Aerial Vehicle - Modeling, Estimation, and Control of Quadrotor", In: *IEEE Robotics & Automation Magazine*, Sept 2012, pp20-32.

Bouabdallah, S., Murrieri, P., Siegwart, R., 2014. Design and control of an indoor micro quadrotor. In: *IEEE International Conference on Robotics and Automation*, January 2014.