Collision-free UAV Formation Flight Using Decentralized Optimization and Invariant Sets

Francesco Borrelli, Tamás Keviczky and Gary J. Balas*

***Abstract*— We consider the problem of formation flight for a set of Unmanned Air Vehicles (UAV). We propose a decentralized control design procedure which guarantees collision avoidance and constraint fulfillment. The control design is based on a decentralized Receding Horizon Control (RHC) scheme [1]. Vehicle collision avoidance is ensured by considering a collision-free emergency maneuver which is implemented when feasibility of the decentralized RHC scheme is lost. Bounds on speed and accelerations are computed off-line using simple polyhedral invariant set computations. Such bounds guarantee that the implementation of the emergency maneuver leads to collision-free trajectories.**

The proposed decentralized control scheme is formulated as mixed-integer linear programs of small sizes which can be translated into equivalent piecewise affine state-feedback controllers. These controllers can be implemented in real-time once the corresponding look-up tables are downloaded to the hardware platform of the UAVs.

I. Introduction

Interest in the formation control of Unmanned Air Vehicles (UAVs) has grown significantly over the last years. The main motivation is the wide range of military and civilian applications where UAV formations could provide a low cost and efficient alternative to existing technology. Among them, distributed sensing applications are envisioned to be the most appealing. Such applications include Synthetic Aperture Radar (SAR) interferometry, surveillance, damage assessment, reconnaissance, chemical or biological agent monitoring, exploration, vegetation growth analysis, assessment of topographical changes [2]. These kind of applications require the development of control system design techniques for large and tight formations.

Formation flight can be viewed as a large control problem which computes the inputs driving the UAVs along challenging maneuvers while maintaining relative positions as well as safe distances between each UAV pair. Optimal control has been the most successful technique to formulate and tackle such a problem [2], [3], [4], [5], [6], [7]. Centralized optimal or suboptimal approaches have been used in different studies. However, as the number of UAVs increases, the solution of big, centralized, non-convex optimization problems becomes prohibitive, even having the most advanced optimization solver, or using oversimplified linear vehicle dynamics [8].

In a recent work [1] we have proposed a decentralized optimal control framework which could help overcome the drawbacks listed above. In particular, we make use of distributed Receding Horizon Control (RHC) schemes. The main idea is to break a centralized RHC controller into distinct RHC controllers of smaller sizes. Each RHC controller is associated to a different UAV and computes the local control inputs based only on the states of itself and its neighbors. On each UAV, the current state and model of its neighbors are used to predict their possible trajectories and move accordingly. The information-exchange topology and inter-vehicle constraints are described by using graph topology terminology. The framework proposed in [1] has the following advantages:

- Different maneuvering objectives can be achieved by changing appropriate terms in the cost function (e.g. formation keeping, formation joining and formation flying).
- Individual vehicles use neighbor information to predict their behavior in order to avoid collisions and act in a cooperative, rather than worst-case way (similarly to what we do while driving cars).
- Can handle constrained MIMO linear models as well as constrained MIMO piecewise linear models of UAVs.
- The problem is formulated and solved as small MILPs which can be translated into equivalent gain scheduled controllers for real-time implementation.

The approach proposed in [1] has two main issues. First, the formulation used time-invariant interconnection graphs. Second, collision avoidance guarantees were formulated in terms of the robustness of each decentralized controller to prediction errors on neighbors' trajectories.

In this paper we modify the approach in [1] in order to add the following features to the list above:

- Handling of time-varying interconnection topologies.
- Use of emergency controllers and their invariant sets as protection zones to guarantee collision avoidance when the local RHC subproblems become infeasible.
- Use of inter-vehicle coordination rules (e.g. "right-of-way") which are formulated as binary decision variables in the local decentralized controllers.

The proposed framework is a step towards a systematic design procedure for real-time, decentralized, collision-free formation flight.

II. UAV Model

The UAV dynamical model used in this paper reflects the simplified dynamics of the Organic Air Vehicle (OAV). The

*Department of Aerospace Engineering and Mechanics, University of Minnesota, 107 Akerman Hall, 110 Union Street S.E., Minneapolis, MN 55455, {borrelli,keviczky,balas}@aem.umn.edu

OAV is a hovering ducted fan vehicle under development at the Honeywell Laboratories. The OAV dynamics are highly nonlinear. An inner-loop dynamic-inversion based controller is used to stabilize the system by actuating the control surfaces of the vehicle [9]. At the outer level the OAV behaves as a MIMO linear system, where the inputs are the accelerations along the x,y,z-axes and the states are positions and velocities of the vehicle in the x,y,z-axes. We describe the OAV dynamics by using the following linear discrete-time model

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where the state update function $f : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^6$ is a linear function of its inputs and $x_k \in \mathbb{R}^6$, $u_k \in \mathbb{R}^3$ are states and inputs of the vehicle at time k, respectively. In particular,

$$x_k = \begin{bmatrix} x_{k,pos} \\ x_{k,vel} \end{bmatrix}, \quad u = \begin{bmatrix} \text{x-axis acceleration} \\ \text{y-axis acceleration} \\ \text{z-axis acceleration} \end{bmatrix}$$

and $x_{k,pos} \in \mathbb{R}^3$ is the vector of x, y and z coordinates and $x_{k,vel} \in \mathbb{R}^3$ is the vector of x-axis, y-axis and z-axis velocity components at time k. It is important to emphasize that the approach proposed in this paper can easily accommodate higher order, more complex linear or piecewise-linear models that describe the UAV dynamics with higher fidelity.

States and inputs of the UAV are constrained. We will consider two types of constraints. A set of "emergency" constraints and a set of "nominal" constraints. The "nominal" constraints define the operating region of the vehicle under normal operation. Nominal constraints are more restrictive than the actual operating limits of the UAV[1]. Maximum performance is used only in emergency situations and is defined by the set of "emergency" constraints.

- Nominal constraints

$$\begin{aligned} & x_{vel} \in \mathcal{X}_v = \\ & \quad \{z \in \mathbb{R}^3 | -10/\beta \text{ ft/s} \le z_i \le 10/\beta \text{ ft/s}, i = 1,2,3\} \\ & u \in \mathcal{X}_u = \\ & \quad \{z \in \mathbb{R}^3 | -3/\alpha \text{ ft/s}^2 \le z_i \le 3/\alpha \text{ ft/s}^2, i = 1,2,3\} \end{aligned} \tag{2}$$

- Emergency constraints

$$\begin{aligned} & x_{vel} \in \mathcal{X}_v^{ER} = \\ & \quad \{z \in \mathbb{R}^3 | -10 \text{ ft/s} \le z_i \le 10 \text{ ft/s}, i = 1,2,3\} \\ & u \in \mathcal{X}_u^{ER} = \\ & \quad \{z \in \mathbb{R}^3 | -3 \text{ ft/s}^2 \le z_i \le 3 \text{ ft/s}^2, i = 1,2,3\} \end{aligned} \tag{3}$$

III. UAV Emergency Maneuver and Invariant Set

In this section we discuss the main idea of the proposed approach. We consider a single UAV (1) and a state-feedback emergency controller

$$u_k = g(x_k, r_e) \tag{4}$$

[1]The positive constants α and β are suitably chosen, not critical, but rather practical values that restrict the use of excessive vehicle performance during nominal operation.

which controls the UAV to a chosen reference r_e under the constraints (3). Denote by t_e the time instant when an emergency maneuver starts. The closed-loop UAV dynamics during emergency maneuver are

$$x_{k+1} = f(x_k, g(x_k, r_e)) \quad \text{for} \quad k \ge t_e \tag{5}$$

The emergency controller $g(x_k, r_e)$ can be designed to achieve different objectives depending on the type of UAV and on its mission. For instance the emergency maneuver could consist of bringing the vehicle to a full stop. For winged UAVs it could perform a continuous flight in a circle of a given radius. Without loss of generality, in this report we assume that an emergency maneuver started at t_e brings the vehicle to the position which it had at time t_e and zero terminal speed, i.e., $r_e = \left[x_{t_e,pos}, \vec{0}\right]$.

We define an emergency region $\mathcal{X}_p^{ER} \subset \mathbb{R}^3$ centered at $x_{t_e,pos}$ to be a polytope in the x, y, z space containing the UAV position during emergency maneuvers. In order to guarantee this property, we compute off-line the set $\Xi(t_e) \in \mathbb{R}^6$ of vehicle positions and speeds at the time t_e such that closed loop dynamics (5) for $k \ge t_e$ and $x_{t_e} \in \Xi(t_e)$ lie in the emergency set $\mathcal{X}_p^{ER}$. $\Xi(t_e)$ is a positively invariant set of system (5) subject to constraints on speed and acceleration (3) and on position defined by $\mathcal{X}_p^{ER}$

$$\begin{aligned} & x \in \Xi(t_e) \iff \\ & \quad x_{vel} \in \mathcal{X}_v^{ER}, g(x) \in \mathcal{X}_u^{ER}, x_{pos} \in \mathcal{X}_p^{ER}(t_e), \\ & \quad f(x, g(x)) \in \Xi(t_e) \quad \forall t \ge t_e \end{aligned} \tag{6}$$

The UAV is guaranteed to satisfy emergency constraints on speed and acceleration and to stay in the emergency region $\mathcal{X}_p^{ER}$ only if the emergency maneuver (4) was started when all the states were in $\Xi(t_e)$.

If $g(x)$ is a linear state-feedback controller, then $\Xi(0)$ can be easily computed with simple techniques using polyhedral manipulations as in [10]. Another possibility is to design an infinite-time constrained linear quadratic regulator for system (1) subject to constraints (3), (8) as described in [11]. This procedure will compute a piecewise-linear controller $g(x)$ and the polyhedral invariant set $\Xi(0)$. The set $\Xi(k)$ is just a translation of the set $\Xi(0)$ to the position $x_{k,pos}$.

In conclusion, once Ξ has been computed, in order to guarantee that the vehicle performs the emergency maneuver within the emergency region we have to augment the nominal constraints (2) with the constraint

$$x_k \in \Xi(k) \tag{7}$$

In the following, we select $\mathcal{X}_p^{ER}$ as

$$x_{pos} \in \mathcal{X}_p^{ER}(t_e) = \{|x_{t_e,pos} - x_{pos}| \preceq 5\} \tag{8}$$

A cross section of the maximal positively invariant set is shown in Figure 1 using limits defined in (3), (8) and an LQR regulator. The trajectories of the UAV performing an emergency stop will lie in the set $\Xi(x_{t_e})$ if at time t_e the

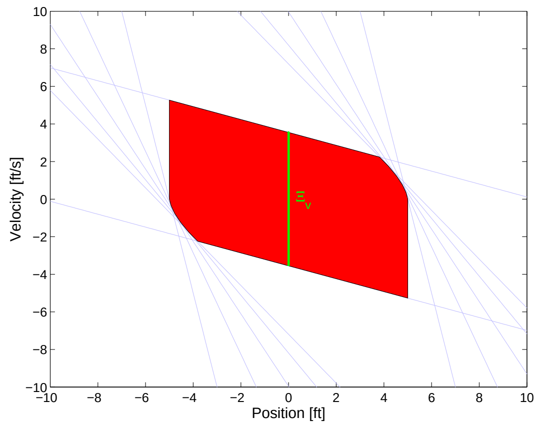


Fig. 1. Cross section of the invariant set $\Xi(t_e)$, showing the resulting Ξ_v constraint set, which has to be enforced on nominal speed in order to remain inside the invariant region using the emergency controller.

state of the UAV x_{t_e} belongs to the set $\Xi(t_e)$. Since $\Xi(x_{t_e})$ is centered in x_{t_e}, constraint (7) becomes

$$x_{t_e,vel} \in \Xi_v, \quad \Xi_v = \left\{ x_v \in \mathbb{R}^3 | \left(\vec{0}, x_v \right) \in \Xi \right\}$$

The set Ξ_v will constrain the speed of the UAV to lie within bounds from which an emergency stop can be accomplished without violating $\mathcal{X}_p^{ER}$. Ξ is a polyhedron and therefore Ξ_v will be a polyhedron as well. The size of Ξ_v is a function of $\mathcal{X}_p^{ER}, \mathcal{X}_v^{ER}$ and $\mathcal{X}_u^{ER}$. The bigger $\mathcal{X}_u^{ER}$ is, the faster the UAV can stop, which leads to a bigger set Ξ_v from which an emergency stop can start. The smaller $\mathcal{X}_p^{ER}$ is, the smaller the set of initial velocities becomes from which the vehicle can stop in $\Xi(t_e)$. In other words, there is a trade-off between the nominal vehicle speed limits and the extent to which vehicles can accelerate/decelerate.

Once the invariant set $\Xi(k)$ has been computed we design decentralized RHC controllers for formation flying which enforce: (i) the constraint (7), (ii) use protection zones larger than $\Xi(k)$, (iii) switch to an emergency maneuver when the constrained optimization problem becomes infeasible. The design of decentralized RHC controllers is detailed next.

IV. DECENTRALIZED CONTROL STRATEGY

We consider a set of N_v dynamical systems representing the UAVs, where the i-th system is described by the discrete-time time-invariant state equation:

$$x_{k+1}^i = f^i(x_k^i, u_k^i) \tag{9}$$

where $x_k^i \in \mathbb{R}^n$, $u_k^i \in \mathbb{R}^m$, $n = 6$, $m = 3$ are states and inputs of the i-th system, respectively, and f^i is the state update function (1). The speed and acceleration of each UAV is constrained according to (2) and (7). Each UAV can perform the same emergency maneuver and to each UAV we assign a logic state $x_{k,L}^i$ which is 1 if the i-th vehicle is performing an emergency maneuver at time k, $x_{k,L}^i = 0$ if the vehicle is operating under nominal conditions.

We will refer to the set of N_v constrained systems as *team system*. Let $\bar{x}_k \in \mathbb{R}^{N_v \times n}$ and $\bar{u}_k \in \mathbb{R}^{N_v \times m}$ be the vectors which collect the states and inputs of the team system at time k, i.e. $\bar{x}_k = [\bar{x}_k^1, \ldots, \bar{x}_k^{N_v}]$, $\bar{u}_k = [\bar{u}_k^1, \ldots, \bar{u}_k^{N_v}]$, with

$$\bar{x}_{k+1} = \bar{f}(\bar{x}_k, \bar{u}_k) \tag{10}$$

We denote by (x_e^i, u_e^i) the equilibrium pair of the i-th system and $(\tilde{x}_e, \tilde{u}_e)$ the corresponding equilibrium for the team system.

So far the individual systems belonging to the team system are completely decoupled. We consider an optimal control problem for the team system where cost function and constraints couple the dynamic behavior of individual systems. We use a graph topology to represent the coupling in the following way. We associate the i-th system to the i-th node of the graph, and if an edge (i, j) connecting the i-th and j-th node is present, then the cost and the constraints of the optimal control problem will have a component, which is a function of both x^i and x^j. The graph will be *undirected*, i.e. $(i, j) \in \mathcal{A} \Rightarrow (j, i) \in \mathcal{A}$, and the edge will be present if the nodes are close enough.

Therefore, before defining the optimal control problem, we need to define a time-varying graph

$$\mathcal{G}(t) = \{\mathcal{V}, \mathcal{A}(t)\} \tag{11}$$

where $\mathcal{V}$ is the set of nodes $\mathcal{V} = \{1, \ldots, N_v\}$ and $\mathcal{A}(t) \subseteq \mathcal{V} \times \mathcal{V}$ the set of time-varying arcs (i, j) with $i \in \mathcal{V},\ j \in \mathcal{V}$. (Note that in the following description time-dependent graphs are used as opposed to the time-independent graph structures in [1]).Choosing the time-dependence of the set of arcs is not a straightforward problem. Even if we assume that each vehicle can sense every other, there are several ways of selecting who will be considered as a neighbor and who will not. Clearly, a small number of neighbors is preferred, otherwise each node would solve a centralized problem, if all of them are taken into account. On the other hand, if we assume a more realistic scenario, where not all vehicles can communicate with or sense every other, a particular neighbor-selection policy can easily lead to a disconnected graph, which could prevent the team system from reaching the desired objective. Conditions on the graph structure and different ways of ensuring the connectedness of the time-varying graph using appropriate neighbor-selection rules are under investigation. One practical way of helping the formation to rejoin (regain connectedness) is to have terminal position references in each maneuver, which would guarantee a connected graph using a particular neighbor-selection policy. For the following exposition, we assume that the time-dependence of the set of arcs will be function of the relative distance of the vehicles. We define here the set $\mathcal{A}(t)$ as

$$\mathcal{A}(t) = \{(i, j) \in \mathcal{V} \times \mathcal{V} \ \mid\ \|x_{t,pos}^i - x_{t,pos}^j\| \leq d_{min}\} \tag{12}$$

that is the set of all the arcs, which connect two nodes whose distance is less than or equal to d_{min}. The choice of the parameter d_{min} will be discussed later.

Denote with $\tilde{x}_k^i$ the states of all neighbors of the i-th system at time k, i.e. $\tilde{x}_k^i = \{x_k^j \in \mathbb{R}^{n^j} | (j,i) \in \mathcal{A}(k)\}$, $\tilde{x}_k^i \in \mathbb{R}^{\tilde{n}_k^i}$ with $\tilde{n}_k^i = \sum_j \dim\{n_k^j | (j,i) \in \mathcal{A}(k)\}$. Analogously, $\tilde{u}_k^i \in \mathbb{R}^{\tilde{m}_k^i}$ denotes the inputs to all the neighbors of the i-th system at time k. Let

$$g^{i,j}(x_{pos}^i, x_{pos}^j) \leq 0 \tag{13}$$

define the safety distance constraints between the i-th and the j-th UAV, with $g^{i,j} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{nc_{i,j}}$. We will often use the following shorter form of the interconnection constraints defined between the i-th system and all its neighbors:

$$g_k^i(x_k^i, \tilde{x}_k^i) \leq 0 \tag{14}$$

with $g_k^i : \mathbb{R}^{n^i} \times \mathbb{R}^{\tilde{n}_k^i} \to \mathbb{R}^{nc_{i,k}}$.

Consider the following cost

$$l(\tilde{x}, \tilde{u}) = \sum_{i=1}^{N_v} l_k^i(x^i, u^i, \tilde{x}_k^i, \tilde{u}_k^i) \tag{15}$$

where $l^i : \mathbb{R}^{n^i} \times \mathbb{R}^{m^i} \times \mathbb{R}^{\tilde{n}_k^i} \times \mathbb{R}^{\tilde{m}_k^i} \to \mathbb{R}$ is the cost associated to the i-th system and is a function of its states and the states of its neighbor nodes. Assume that l is a positive convex function with $l(\tilde{x}_e, \tilde{u}_e) = 0$ and consider the following decentralized scheme.

Let the following finite time optimal control problem $\mathcal{P}_i(t)$ be associated to the i-th system at time t

$$\min_{\tilde{U}_t^i} \quad \sum_{k=0}^{N-1} l_t^i(x_{k,t}^i, u_{k,t}^i, \tilde{x}_{k,t}^i, \tilde{u}_{k,t}^i) + l_N^i(x_{N,t}^i, \tilde{x}_{N,t}^i)$$

$$\begin{aligned}
\text{subj. to} \quad & x_{k+1,t}^i = f^i(x_{k,t}^i, u_{k,t}^i), k \geq 0 \\
& x_{k,t,vel}^i \in \mathcal{X}_v, \quad u_{k,t}^i \in \mathcal{X}_u, \\
& \quad k = 1, \ldots, N-1 \\
& \text{IF} \quad x_{t,L}^j = 0 \\
& \quad x_{k+1,t}^j = f^j(x_{k,t}^j, u_{k,t}^j), \quad (j,i) \in \mathcal{A}(t), \\
& \qquad k \geq 0 \\
& \quad x_{k,t,vel}^j \in \mathcal{X}_v, \quad u_{k,t}^j \in \mathcal{X}_u, \quad (j,i) \in \mathcal{A}(t), \\
& \qquad k = 1, \ldots, N-1 \\
& \text{ELSE} \\
& \quad x_{k+1,t}^j = f^j(x_{k,t}^j, g(x_{k,t}^j)), \quad k \geq 0 \\
& \text{END} \\
& g^{i,j}(x_{k,t,pos}^i, x_{k,t,pos}^j) \leq 0, \quad (i,j) \in \mathcal{A}(t), \\
& \quad k = 1, \ldots, N-1 \qquad \text{(16a)} \\
& g^{q,r}(x_{k,t,pos}^q, x_{k,t,pos}^r) \leq 0, \qquad \text{(16b)} \\
& \quad (q,i) \in \mathcal{A}(t), \ (r,i) \in \mathcal{A}(t), \\
& \quad k = 1, \ldots, N-1 \\
& x_{k,t,vel}^i \in \Xi_v, \ k \geq 0 \qquad \text{(16c)} \\
& x_{k,t,vel}^j \in \Xi_v, (j,i) \in \mathcal{A}(t), \ k \geq 0 \qquad \text{(16d)} \\
& x_{N,t}^i \in \mathcal{X}_f^i, \quad x_{N,t}^j \in \mathcal{X}_f^j, (i,j) \in \mathcal{A}(t) \\
& x_{0,t}^i = x_t^i, \quad \tilde{x}_{0,t}^i = \tilde{x}_t^i \qquad \text{(16e)}
\end{aligned}$$

where $\tilde{U}_t^i \triangleq [u_{0,t}^i, \tilde{u}_{0,t}^i, \ldots, u_{N-1,t}^i, \tilde{u}_{N-1,t}^i]' \in \mathbb{R}^s$, $s \triangleq (\tilde{m}^i + m^i)N$ denotes the optimization vector, $x_{k,t}^i$ denotes the state vector of the i-th node predicted at time $t+k$ obtained by starting from the state x_t^i and applying to system (1) the input sequence $u_{0,t}^i, \ldots, u_{k-1,t}^i$. The tilded vectors denote the prediction vectors associated to the neighboring systems *assuming a constant interconnection graph*. Denote by $\tilde{U}_t^{i*} = [u_{0,t}^{*i}, \tilde{u}_{0,t}^{*i}, \ldots, u_{N-1,t}^{*i}, \tilde{u}_{N-1,t}^{*i}]$ the optimizer of problem $\mathcal{P}_i(t)$. Note that problem $\mathcal{P}_i(t)$ involves only the state and input variables of the i-th node and its neighbors at time t.

We will define the following decentralized RHC control scheme. At time t (assuming logic states are initialized at time 0)

1) Compute graph connection $\mathcal{A}$(t) according to (12)
2) Each node i solves problem $\mathcal{P}_i(t)$
3) If $\mathcal{P}_i(t)$ is feasible then $x_{t,L}^i = 0$ and node i implements the first sample of $\tilde{U}_t^{i*}$
$$u_t^i = u_{0,t}^{*i}. \tag{17}$$
4) else $x_{t,L}^i = 1$, and node i implements the emergency controller
$$u_t^i = g(x_t^i). \tag{18}$$
5) Each node repeats steps 1 to 4 at time $t+1$, based on the new state information x_{t+1}^i, $\tilde{x}_{t+1}^i$.

In order to solve problem $\mathcal{P}_i(t)$ each node needs to know its current states, its neighbors' current states, its terminal region, its neighbors' terminal regions and models and constraints of its neighbors. Based on such information each node computes its optimal inputs and its neighbors' optimal inputs assuming a constant set of neighbors over the horizon. The input to the neighbors will only be used to predict their trajectories and then discarded, while the first component of the i-th optimal input of problem $\mathcal{P}_i(t)$ will be implemented on the i-th node. Moreover, each node takes into account a possible emergency maneuver of its UAV neighbors, using their emergency logic state information.

Without Step 4, even if we assume N to be infinite, the decentralized RHC approach does not guarantee that solutions computed locally are centrally feasible and stable. The reason is simple: At the i-th node the prediction of the neighboring state x^j is done independently from the prediction of problem $\mathcal{P}_j(t)$. Therefore, the trajectory of x^j predicted by problem $\mathcal{P}_i(t)$ and the one predicted by problem $\mathcal{P}_j(t)$, based on the same initial conditions, are different (since, in general, $\mathcal{P}_i(t)$ and $\mathcal{P}_j(t)$ will be different). This will imply that constraint fulfillment will be ensured by the optimizer u_t^{*i} for problem $\mathcal{P}_i(t)$ but not for the real closed loop trajectories of all the UAVs. Further discussion on the feasibility issue of decentralized RHC can be found in [1].

Step 4 guarantees collision-free formation flight. The main idea of the scheme is to use an emergency stop maneuver if a vehicle's local RHC problem is infeasible. Constraint (16c) on each vehicle will ensure that the emergency maneuver can be performed as discussed in Section III. If the polyhedron $\mathcal{X}_p^{ER}$ is contained within the protection zones then the vehicles are guaranteed not to collide.

The scheme above can lead to three different behaviors: (i) the vehicles fly or hover in formation (with occasional use of the emergency controllers to recover feasibility), (ii) all the vehicles are in an emergency status, i.e. $x_{t,L}^i = 1,\ \forall i \in \mathcal{V}$ or (iii) a subset of vehicles are in emergency status and the rest are under nominal operation, yet kept from achieving their desired goals due to the coupling in cost between the emergency and nominal vehicles' relative positions. The neighboring nominal vehicles also remain stationary in an equilibrium where the cost decrement of moving towards the target point is balanced by the cost increment that would result from leaving the "formation" of stalled vehicles. This results in a dead-lock, where not all vehicles are in emergency mode.

In the first case, vehicles could spend a finite amount of time performing an emergency maneuver before the RHC problem recovers from infeasibility. The second and third cases however could end up with a dead-lock situation where all or some subset of UAVs are in an emergency status. In this case an emergency centralized scheme would be needed in order to let the UAVs recover from the dead-lock condition. A feasible solution to such a centralized scheme with sufficiently long horizon is always guaranteed to exist (e.g. the vehicles could fly one after the other starting from the outside).

The control scheme presented in this section could result in a jerky behavior of the formation by frequent switches to emergency controllers and it could be quite conservative. In the next section we will discuss in detail different ways of addressing these two issues.

V. Emergency Maneuver and Practical Implementation Issues

In order to reduce the frequent occurrence of emergency maneuvers we modify problem P_i in two ways. We make use of slack variables to avoid optimal maneuvers that involve touching protection zones. The other approach can be used to establish inter-vehicle coordination (e.g. "right-of-way") rules by means of including binary decision variables, that arise in the MILP problem formulation, in the cost function or in the constraints of the local decentralized controllers.

Simulations with the decentralized scheme applied to formation flight showed that constraints can often become active during maneuvers. This implies that a small error between the predicted trajectories of neighbors and their real trajectories can lead to infeasibility of the decentralized scheme. Optimal maneuvers can be moved away from the boundary of protection zones by modifying constraints (16a)-(16b) as

$$g^{i,j}(x^i_{k,t,pos}, x^j_{k,t,pos}) \leq d_{safe}(\varepsilon^{k,i,j} - 1), \quad (19)$$

$$0 \leq \varepsilon^{k,i,j}, \quad (20)$$

$$\varepsilon^{k,i,j} \leq 1 \quad (21)$$

and weighting in the cost the slack variables $\varepsilon^{k,i,j}$

$$\min_{\tilde{U}_t^i} \sum_{k=0}^{N-1} \Bigg(l^i(x^i_{k,t}, u^i_{k,t}, \tilde{x}^i_{k,t}, \tilde{u}^i_{k,t}) + l^i_N(x^i_{N,t}, \tilde{x}^i_{N,t}) + \sum_{j|(i,j)\in\mathcal{A}} \rho\varepsilon^{k,i,j} \Bigg) \quad (22)$$

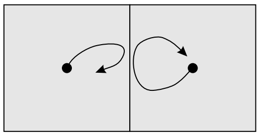
The parameters $\rho > 0$ and $d_{safe} > 0$ will be defined by the user. The higher d_{safe} is, the less compact the formation will look like.

In order to improve the likelihood of feasibility of the decentralized scheme different "right-of-way" priorities can be introduced which allows to have better prediction about neighbors' trajectories. This can be easily achieved if protection zones are modeled as parallelepipeds and the disjunctions are modeled as binary variables [8]. "Right-of-way" priorities can be translated into weights and constraints on the binary variables which describe the location of a vehicle with respect to a parallelepipedal protection zone of another vehicle (six binary variables in three dimensions for each vehicle couple [8]).

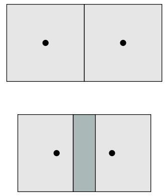
Note that these practical techniques will not imply feasibility by themselves but reduce the frequency of emergency maneuvers avoiding undesirable formation behavior.

There are a few important practical observations that are due regarding the proposed emergency maneuver-based collision avoidance. Notice that if the protection zone of each vehicle is chosen to be equal to the invariant set $\mathcal{X}_p^{ER}$ described in Section III, the emergency maneuver guarantees only collision avoidance, not protection zone-sized separation of vehicles at all times, as illustrated by Figure 2(a) for a simple example. This means that protection zones should be chosen larger than the invariant set calculated in Section III if a certain minimum separation is required.

We should also realize that due to the discrete nature of the problem formulation and controller implementation, at a certain time instant vehicles can become infeasible, when the protection zones have already been violated as illustrated in Figure 2(b). This situation can easily occur due to disturbances, model mismatch or incorrect predictions about neighbors. The emergency controller should still guarantee collision avoidance in this case, which can be achieved again, by enlarging the protection zones to account for the one-time-step worst-case behavior of neighboring vehicles. In other words, the emergency invariant sets should be contained in a set that is obtained by shrinking the protection zone with a one-step worst-case maneuver of the neighbor.



(a) Invariant sets guarantee collision avoidance, not minimum separation.



(b) Infeasibility can occur after protection zone violation.

Fig. 2. Typical situations that illustrate the need to select the protection zone size to be larger than the invariant set of the emergency controller.

A. Conservativeness and Design Parameters

The conservativeness of the presented scheme is a function of several parameters. The key parameters are $\mathcal{X}_p^{ER}$ which affects Ξ_v, α and β. These all influence the selection of the emergency region and the protection zones. If $\mathcal{X}_p^{ER}$ is very small, then depending on the acceleration limits, Ξ_v might become small, which implies that the UAVs could only fly at very low speeds. On the other hand, if $\mathcal{X}_p^{ER}$ is big, then the vehicles are required to fly very far from each other. As a result the system might perform far from its optimal point. It is also conceivable to use an hybrid strategy: once the desired formation has been reached we can relax the nominal constraints on speed and use maximum vehicle performance, whereas in case of reconfiguration the constraints $x_{vel} \in \Xi_v$ are reinserted. This will allow to fly at low speeds only during decentralized maneuvers.

B. Real-time implementation

The presence of nonlinearities and constraints on one hand, and the simplicity needed for real-time implementation on the other, would discourage the design of optimal control strategies as presented above. Recently, a new framework for modeling constrained switched systems and an algorithm to synthesize piecewise affine optimal controllers for such systems has been proposed [12]. Based on such framework, the design of the decentralized controllers will be performed in two steps. First, the decentralized RHC controllers based on linear or piecewise linear UAV model are tuned in simulation until the desired performance is achieved. The RHC controllers are not directly implementable, as it would require the mixed-integer linear programs to be solved on-line on each UAV. Therefore, for implementation, in the second phase the explicit piecewise affine form of the RHC law is computed off-line by using the multiparametric mixed integer programming solver presented in [12]. The use of equivalent piecewise affine form of the RHC law will have several advantages. It is immediate to implement on a UAV platform as a simple look-up table of gain-scheduled controllers. It can also be easily verified (an on-line optimization solver is impossible to verify). Its worst case computational time can also be computed immediately.

VI. Examples

The ideas presented in this paper are demonstrated in a simulation example involving six UAVs performing planar motion for easier illustration. A movie of the simulation can be found on the web page [13]. The vehicles are lined up beside each other moving with the same velocity initially. They are required to change into an also moving triangular formation of six vehicles, while avoiding obstacles and collisions with each other. Each UAV uses the decentralized RHC scheme introduced in [1] and augmented with the emergency controllers described in this paper to ensure collision-avoidance. The protection zones and speed constraints incorporated into the decentralized RHC problems were obtained from the invariant sets associated with simple LQR emergency controllers. The time-varying interconnection graph is obtained by each UAV communicating with at most two of its closest neighbors. Depending on the line-of-sight obstruction caused by obstacles, the number of visible neighbors might be less than two at a certain time instant. This interconnection policy means that the graph becomes directed, since being the closest neighbor to another UAV is not necessarily a mutual relationship.

The simulation example illustrates the use of the emergency mode controllers when the decentralized RHC problems of three UAVs become infeasible. Slightly after these UAVs begin their collision-free emergency stop maneuvers, they recover feasibility and catch up with the other three to get into formation while avoiding obstacles.

References

[1] T. Keviczky, F. Borrelli, and G. J. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proc. American Contr. Conf.*, 2004.

[2] D. Stipanovic, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica, to appear*, 2003.

[3] A. Richards, J. Bellingham, M. Tillerson, and J. P. How, "Coordination and control of multiple uavs," in *AIAA Guidance, Navigation, and Control Conference, Monterey, CA*, 2002.

[4] D. Shim, H. Kim, and S. Sastry, "Decentralized reflective model predictive control of multiple flying robots in dynamic enviroment," Department of Electical Engineering and Computer Sciences. University of California at Berkeley, Tech. Rep., 2003.

[5] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formation," in *Proc. 41th IEEE Conf. on Decision and Control*, 2002.

[6] R. Olfati-Saber, W. B. Dunbar, and R. M. Murray, "Cooperative control of multi-vehicle systems using cost graphs and optimization," in *Proc. American Contr. Conf.*, 2003.

[7] F. Giulietti, L. P. L., and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 34–44, 2000.

[8] F. Borrelli, D. Subramanian, A. U. R. amd L. T. Biegler, and T. Samad, "A comparison between mixed integer programming and nonlinear programming techniques for 3d conflict resolution of multiple aircraft," Department of Aerospace Engineering and Mechanics. University of Minnesota, Minneapolis., Tech. Rep., 2003.

[9] F. Kingsley and F. Borrelli, "Oav formation flight: a decentralized optimization based approach," Honeywell Laboratories, Minneapolis., Tech. Rep., 2004.

[10] E. C. Kerrigan, "Robust constraints satisfaction: Invariant sets and predictive control," Ph.D. dissertation, Department of Engineering, University of Cambridge, Cambridge, England, 2000.
[11] P. Grieder, F. Borrelli, F. Torrisi, and M. Morari, "Constrained infinite horizon linear quadratic regulator," *Automatica*, vol. 40, no. 4, pp. 701–708, July 2004.
[12] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. Springer, 2003, vol. 290.
[13] [Online]. Available: http://www.aem.umn.edu/people/others/borrelli/ff.htm