Model Predictive Control for Decoupled Systems: A Study on Decentralized and Distributed Schemes

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

Abstract

We consider a set of decoupled dynamical systems and an optimal control problem where cost function and constraints couple the dynamical behavior of the systems. The coupling is described through a connected graph where each system is a node and, cost and constraints of the optimization problem associated to each node are only function of its state and the states of its neighbors. For such scenario, we describe different strategies for designing decentralized and distributed Model Predictive Control (MPC) control schemes.

In decentralized schemes a centralized MPC controller is broken into distinct MPC controllers of smaller sizes. Each MPC controller is associated to a different node and computes the local control inputs based only on the states of the node and of its neighbors. In distributed control schemes certain nodes compute the control inputs for low-priority neighboring nodes while using information coming from higher-priority neighboring nodes. In general, computation is distributed over the nodes and the number of MPC controllers is smaller than the total number of nodes.

We formulate decentralized and distributed control schemes in a rigorous mathematical framework. Moreover, we highlight the main issues involved in guaranteeing stability and constraint fulfillment for such schemes and the degree of conservativeness that the decentralized approach introduces.

1 Introduction

The interest in decentralized control goes back to the seventies. Probably Wang and Davison were the first in [1] to envision the "increasing interest in decentralized control systems" when "control theory is applied to solve problems for large scale systems". Since then the interest has grown more than exponentially despite some non-encouraging results on the complexity of the problem [2]. Decentralized control techniques today can be found in a broad spectrum of applications ranging from robotics and formation flight to civil engineering. Such a wide interest makes a survey of all the approaches that have appeared in the literature very difficult and goes also beyond the scope of this paper.

Approaches to decentralized control design differ from each other in the assumptions they make on: (i) the kind of interaction between different systems or different components of the same system (dynamics, constraints, objective), (ii) the model of the system (linear, nonlinear, constrained, continuous-time, discrete-time), (*iii*) the model of information exchange between the systems, (*iv*) the control design technique used.

Dynamically coupled systems have been the most studied. In [1] the authors consider a linear time-invariant system and give sufficient conditions for the existence of feedback laws which depend only on partial system outputs. Recently, in [3] the authors introduce the concept of quadratic invariance of a constraint set with respect to a system. The problem of constructing decentralized control systems is formulated as one of minimizing the closed loop norm of a feedback system subject to constraints on the control structure. The authors show that quadratic invariance is a necessary and sufficient condition for the existence of decentralized controllers. In [4] the authors consider spatially interconnected systems, i.e. systems composed of identical linear time-invariant systems which have a structured interconnection topology. By exploiting the interconnection topology, the authors study decentralized analysis and system control design using ℓ_2-induced norms and LMI-s.

In this report we will focus on *decoupled systems*. In a descriptive way, the problem of decentralized control for decoupled systems can be formulated as follows. A dynamical system is composed of (or can be decomposed into) distinct dynamical subsystems that can be independently actuated. The subsystems are dynamically decoupled but have common objectives and constraints which make them interact between each other. Typically the *interaction* is local, i.e. the goal and the constraints of a subsystem are function of only a subset of other subsystems' states. The interaction will be represented by an "interaction graph", where the nodes represent the subsystems and an arc between two nodes denotes a coupling term in the goal and/or in the constraints associated to the nodes. Also, typically it is assumed that the *exchange of information* has a special structure, i.e., it is assumed that each subsystem can sense and/or exchange information with only a subset of other subsystems. Often the *interaction graph* and the *information exchange graph* coincide. A decentralized control scheme consists of distinct controllers, one for each subsystem, where the inputs to each subsystem are computed only based on local information, i.e., on the states of the subsystem and its neighbors.

Our interest in decentralized control for dynamically decoupled systems arises from the study of formation flight. In formation flight a certain number of vehicles has to be controlled in order to let them behave as a formation. The vehicle dynamics are often assumed to be decoupled. A formation behavior is achieved only if each vehicle computes its control laws as a function of position and speed of neighboring vehicles. Moreover, each vehicle is required to keep a certain distance from its neighbors. Therefore, objective and constraints couple the overall dynamics. The way vehicles communicate and sense between each other define the *information exchange* graph while the objective and the constraint of the formation define the *interaction* graph.

Several studies have appeared on decentralized techniques for formation tasks. LMI techniques have been used in [5], control Lyapunov function in [6] and a vision-based framework in [7].

We will make use of Model Predictive Control schemes. The main idea of MPC is to use the *model* of the plant to *predict* the future evolution of the system [8]. Based on this prediction, at each time step t a certain performance index is optimized under operating constraints with respect to a sequence of future input moves. The first of such optimal moves is the *control* action applied to the plant at time t. At time $t + 1$, a new optimization is solved over a shifted prediction horizon.

Optimal control techniques for formation flight have been extensively studied. Unconstrained decentralized LQR control has been described in [9, 10]. Recently, centralized MPC schemes applied to formation flight have appeared in [11, 12]. In [13] decentralized MPC and potential functions have been used for flying multiple autonomous helicopters in a dynamical environment.

In this paper we describe different strategies for designing *decentralized* and *distributed* MPC control schemes. In decentralized schemes a centralized MPC controller is broken into distinct MPC controllers of smaller sizes. Each MPC controller is associated to a different node and computes the local control inputs based only on the states of the node and of its neighbors. In distributed control schemes certain nodes compute the control inputs for low-priority neighboring nodes while using information coming from higher-priority neighboring nodes. In general, computation is distributed over the nodes and the number of MPC controllers is smaller than the total number of nodes.

The main issue regarding decentralized schemes is that the inputs computed locally are, in general, not guaranteed to be globally feasible and to stabilize the overall team. In general, stability and feasibility of decentralized schemes are very difficult to prove and/or too conservative. A scheme with stability guarantees has been proposed in [14].

We will not give any proof of feasibility and stability of the decentralized and distributed schemes. Instead, we will formulate decentralized and distributed control schemes in a rigorous mathematical framework. We will highlight the main issues involved in guaranteeing stability and constraint fulfillment for such schemes and briefly discuss their conservativeness. We will show the applicability of the proposed approach when decentralized schemes are used for controlling a set of vehicles in formation flight.

2 Problem formulation

Consider a set of N_v decoupled dynamical systems, the i-th system being described by the discrete-time time-invariant state equation:

$$x^i_{k+1} = f^i(x^i_k, u^i_k) \tag{1}$$

where $x^i_k \in \mathbb{R}^{n^i}$, $u^i_k \in \mathbb{R}^{m^i}$, $f^i : \mathbb{R}^{n^i} \times \mathbb{R}^{m^i} \to \mathbb{R}^{n^i}$ are state, input and state update function of the i-system, respectively. Let $\mathcal{X}^i \subseteq \mathbb{R}^{n^i}$ and $\mathcal{U}^i \subseteq \mathbb{R}^{m^i}$ denote the set of feasible inputs and states of the i-th system, respectively:

$$x^i_k \in \mathcal{X}^i, \;\; u^i_k \in \mathcal{U}^i, \; k \geq 0 \tag{2}$$

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

$$\tilde{x}_{k+1} = f(\tilde{x}_k, \tilde{u}_k) \tag{3}$$

We denote by (x^i_e, u^i_e) 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 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. $(i,j)\in\mathcal{A}\Rightarrow(j,i)\in\mathcal{A}$. Before defining the optimal control problem, we need to define a graph

$$\mathcal{G}=\{\mathcal{V},\mathcal{A}\} \tag{4}$$

where $\mathcal{V}$ is the set of nodes $\mathcal{V}=\{1,\ldots,N_v\}$ and $\mathcal{A}\subseteq\mathcal{V}\times\mathcal{V}$ the sets of arcs (i,j) with $i\in\mathcal{V},\ j\in\mathcal{V}$.

Once the graph structure has been fixed, the optimization problem is formulated as follows. Denote with $\tilde{x}^i$ the states of all neighboring systems of the i-th system, i.e. $\tilde{x}^i=\{x^j\in\mathbb{R}^{n^j}|(j,i)\in\mathcal{A}\}$, $\tilde{x}^i\in\mathbb{R}^{\tilde{n}^i}$ with $\tilde{n}^i=\sum_{j|(j,i)\in\mathcal{A}}n^j$. Analogously, $\tilde{u}^i\in\mathbb{R}^{\tilde{m}^i}$ denotes the inputs to all the neighboring systems of the i-th system. Let

$$g^{i,j}(x^i,u^i,x^j,u^j)\le 0 \tag{5}$$

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

$$g^i(x^i,u^i,\tilde{x}^i,\tilde{u}^i)\le 0 \tag{6}$$

with $g^i:\mathbb{R}^{n^i}\times\mathbb{R}^{m^i}\times\mathbb{R}^{\tilde{n}^i}\times\mathbb{R}^{\tilde{m}^i}\to\mathbb{R}^{nc_i}$.

Consider the following cost

$$l(\tilde{x},\tilde{u})=\sum_{i=1}^{N_v} l^i(x^i,u^i,\tilde{x}^i,\tilde{u}^i) \tag{7}$$

where $l^i:\mathbb{R}^{n^i}\times\mathbb{R}^{m^i}\times\mathbb{R}^{\tilde{n}^i}\times\mathbb{R}^{\tilde{m}^i}\to\mathbb{R}$ is the cost associated to the i-th system and is a function only of its states and the states of its neighbor nodes. Assume that l is a convex function and that $l^i(x^i_e,u^i_e,\tilde{x}^i_e,\tilde{u}^i_e)=0$ and consider the infinite time optimal control problem

$$\tilde{J}^*_\infty(\tilde{x})\triangleq \min_{\{\tilde{u}_0,\tilde{u}_1,\ldots\}}\sum_{k=0}^{\infty} l(\tilde{x}_k,\tilde{u}_k) \tag{8}$$

$$\text{subj. to}\ \begin{cases} x^i_{k+1}=f^i(x^i_k,u^i_k), \quad i=1,\ldots,N_v,\ \ k\ge 0\\ g^{i,j}(x^i_k,u^i_k,x^j_k,u^j_k)\le 0,\ i=1,\ldots,N_v,\ (i,j)\in\mathcal{A}\ \ k\ge 0\\ x^i_k\in\mathcal{X}^i,\quad u^i_k\in\mathcal{U}^i,\ k\ge 0\ i=1,\ldots,N_v,\\ \tilde{x}_0=\tilde{x}\end{cases} \tag{9}$$

For all $\tilde{x}\in\mathbb{R}^{N_v\times n^i}$, if problem (9) is feasible, then the optimal input $\tilde{u}^*_0,\tilde{u}^*_1,\ldots$ will drive the N_v systems to their equilibrium points x^i_e while satisfying state, input and interconnection constraints.

Remark 1. *Throughout the paper we assume that a solution to problem (9) exists and it generates a feasible and stable trajectory for the team system. Our assumption is not restrictive. If there is no infinite time centralized optimal control problem fulfilling the constraints, then there is no reason to look for a decentralized receding horizon controller with the same properties.*

Remark 2. *Since we assumed that the graph is undirected, there will be redundant constraints in problem (9).*

Remark 3. *Note the form of constraints (6) is rather general and it will include the case when only partial information about states of neighboring nodes is involved.*

With the exception of a few cases, solving an infinite horizon optimal control problem is computationally prohibitive. An infinite horizon controller can be designed by repeatedly solving finite time optimal control problems in a receding horizon fashion as described next. At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. The optimal command signal is applied to the process only during the following sampling interval. At the next time step a new optimal control problem based on new measurements of the state is solved over a shifted horizon. The resultant controller is often referred to as Model Predictive Control (MPC). More into details, assume at time t the current state $\tilde{x}_t$ to be available and consider the following constrained finite time optimal control problem

$$\tilde{J}_N^*(\tilde{x}_t) \triangleq \min_{\{U_t\}} \sum_{k=0}^{N-1} l(\tilde{x}_{k,t}, \tilde{u}_{k,t}) + l_N(\tilde{x}_{N,t}) \tag{10a}$$

$$\text{subj. to} \begin{cases} x^i_{k+1,t} = f^i(x^i_{k,t}, u^i_{k,t}), \quad i = 1, \dots, N_v, \;\; k \geq 0 \\ g^{i,j}(x^i_{k,t}, u^i_{k,t}, x^j_{k,t}, u^j_{k,t}) \leq 0, \;\; i = 1, \dots, N_v, (i,j) \in \mathcal{A} \;\; k = 1, \dots, N-1 \\ x^i_{k,t} \in \mathcal{X}^i, \quad u^i_{k,t} \in \mathcal{U}^i \;\; i = 1, \dots, N_v, \;\; k = 1, \dots, N-1 \\ \tilde{x}_{N,t} \in \mathcal{X}_f, \\ \tilde{x}_{0,t} = \tilde{x}_t \end{cases} \tag{10b}$$

where N is the prediction horizon, $\mathcal{X}_f \subseteq \mathbb{R}^{N_v \times n^i}$ is a terminal region, l_N is the cost on the terminal state. In (10) we denote with $U_t \triangleq [\tilde{u}_{0,t}, \dots, \tilde{u}_{N-1,t}]' \in \mathbb{R}^s$, $s \triangleq N_v \times mN$ the optimization vector, $x^i_{k,t}$ denotes the state vector of the i-th node predicted at time $t+k$ obtained by starting from the state x^i_t and applying to system (1) the input sequence $u^i_{0,t}, \dots, u^i_{k-1,t}$. The tilded vectors will denote the prediction vectors associated to the team system.

Let $U_t^* = \{\tilde{u}^*_{0,t}, \dots, \tilde{u}^*_{N-1,t}\}$ be the optimal solution of (10) at time t and $J^*(x_t)$ the corresponding value function. Then, the first sample of U_t^* is applied to the team system (3)

$$\tilde{u}_t = \tilde{u}^*_{0,t}. \tag{11}$$

The optimization (10) is repeated at time $t+1$, based on the new state x_{t+1}.

It is well known that stability is not ensured by the MPC law (10)–(11). Usually the terminal cost l_N and the terminal constraint set $\mathcal{X}_f$ are chosen to ensure closed-loop stability. A treatment of sufficient stability conditions goes beyond the scope of this work and can be found in the surveys [8, 15]. We assume that the reader is familiar with the basic concept of MPC and its main issues, we refer to [8] for a comprehensive treatment of the topic. In this report we will assume that terminal cost l_N and the terminal constraint set $\mathcal{X}_f$ have been appropriately chosen in order to ensure the stability of the closed-loop system.

In general, the optimal input u_t^i to the i-th system computed by solving (10) at time t, will be a function of the overall state information $\tilde{x}_t$. The main objective of this work is to describe how problem (10) can be decomposed into smaller subproblems whose independent computation can be distributed over the graph nodes. First, we propose a decentralized control scheme where problem (10) is decomposed into N_v finite time optimal control problems, each one associated to a different node. The i-th subproblem will be a function of the states of the i-th node and the states of its neighbors. The solution of the i-th subproblem will yield a control policy for the i-th node of the form $u_t^i = f^i(x_t^i, \tilde{x}_t^i)$. Secondly, we describe a distributed control scheme where certain nodes compute the control inputs for low-priority neighboring nodes while using information coming from higher-priority neighboring nodes. In such scheme computation is distributed over the nodes but the correspondence between nodes and optimization problems is not one to one. Next, we formulate decentralized and distributed control schemes in a rigorous mathematical framework. We will start from a centralized stable MPC and discuss the main issues involved in the decentralization of MPC problems.

Remark 4. *The techniques presented next will be meaningful only if the graph $\mathcal{G}$ is not fully connected. Often, the interconnection graph is not fully connected because of the nature of the problem. For instance, each node could represent a production unit of a certain plant and the production of a node could be related to only a few other units of the plant. Also the interconnection graph is not fully connected because some constraints associated to certain arcs are implicitly satisfied by interconnection constraints associated to other arcs. For instance, in formation flight, each vehicle is a node and the graph is fully connected (since each vehicle has to keep a certain distance form all the other vehicles of the formation). However, rigid graph topology [12] can be used in order to implicitly enforce constraints between two vehicles not connected by any arc of the graph.*

Remark 5. *In the formulation above, we are assuming that the equilibrium $(\tilde{x}_e, \tilde{u}_e)$ of the formation is know a priori. The equilibrium of the formation can be defined in several other different ways. For instance, we can assume that there is a leader (real or virtual) which is moving and the equilibrium is given in terms of distances of each vehicle from the leader. Also, it is possible to formulate the equilibrium by using relative distances between vehicles and signed areas [12]. The approach of this paper does not depend on the way used to define the formation equilibrium, as long as this is know a priori. In some formation control schemes, the equilibrium is not known apriori, but is the result of the evolutions of decentralized control laws. For such schemes the approach of the paper will not be useful.*

3 Decentralized Control Scheme

Consider the overall problem: systems (1), graph $\mathcal{G}$, and MPC policy (10)-(11). Consider the i-th system and the following finite time optimal control problem

$$(\mathcal{P}_i): \; J_N^{i*}(x_t^i, \tilde{x}_t^i) \triangleq \min_{\tilde{U}_t^i} \sum_{k=0}^{N-1} l^i(x_{k,t}^i, u_{k,t}^i, \tilde{x}_{k,t}^i, \tilde{u}_{k,t}^i) + l_N^i(x_{N,t}^i, \tilde{x}_{N,t}^i) \tag{12a}$$

$$\text{subj. to} \begin{cases} x_{k+1,t}^i = f^i(x_{k,t}^i, u_{k,t}^i), \quad k \geq 0 \\ x_{k,t}^i \in \mathcal{X}^i, \quad u_{k,t}^i \in \mathcal{U}^i, \quad k = 1, \ldots, N-1 \\ x_{k+1,t}^j = f^j(x_{k,t}^j, u_{k,t}^j), \quad (j,i) \in \mathcal{A}, \; k \geq 0 \\ x_{k,t}^j \in \mathcal{X}^j, \quad u_{k,t}^j \in \mathcal{U}^j \quad (j,i) \in \mathcal{A}, \; k = 1, \ldots, N-1 \\ g^{i,j}(x_{k,t}^i, u_{k,t}^i, x_{k,t}^j, u_{k,t}^j) \leq 0, \; (i,j) \in \mathcal{A}, \\ \qquad k = 1, \ldots, N-1 \\ g^{q,r}(x_{k,t}^q, u_{k,t}^q, x_{k,t}^r, , u_{k,t}^r) \leq 0, \; (q,i) \in \mathcal{A}, \; (r,i) \in \mathcal{A}, \\ \qquad k = 1, \ldots, N-1 \\ x_{N,t}^i \in \mathcal{X}_f^i, \\ x_{N,t}^j \in \mathcal{X}_f^j, (i,j) \in \mathcal{A} \\ x_{0,t}^i = x_t^i, \\ \tilde{x}_{0,t}^i = \tilde{x}_t^i, \end{cases} \tag{12b}$$

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. 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 $\mathcal{P}_i$. Note that problem $\mathcal{P}_i$ involves only the state and input variables of the i-th node and its neighbors.

We will define the following decentralized MPC control scheme.

1. The i-th node at time t knows its state x_t^i and the state of all its neighbors $\tilde{x}_t^i$.

2. Each node i solves problem $\mathcal{P}_i$.

3. Each node i implements the first sample of $\tilde{U}_t^{i*}$

$$u_t^i = u_{0,t}^{*i}. \tag{13}$$

4. Each node repeats steps 2 to 4 at time $t+1$, based on the new states information x_{t+1}^i, $\tilde{x}_{t+1}^i$.

Steps one to four describe a decentralized strategy that uniquely defines the control inputs to the team system. Each node knows 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. 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 $\mathcal{P}_i$ will be implemented on the i-th node.

Even if we assume N to be infinite, the approach described so far does not guarantee that solutions computed locally are centrally feasible and stable (i.e., feasible for problem (10)). The reason is simple. At the i-th node the prediction of the neighboring state x_k^j is done independently from the prediction of problem $\mathcal{P}_j$. Therefore, the trajectory of x^j predicted by problem $\mathcal{P}_i$ and the one predicted by problem $\mathcal{P}_j$, based on the same initial conditions, are different (since, in general, $\mathcal{P}_i$ and $\mathcal{P}_j$ will be different). This will imply that constraint fulfillment will be ensured by the optimizer u_t^{*i} for problem $\mathcal{P}_i$ but not for the overall problem (10).

There are three main issues that arise in the decentralized control scheme. In order to ensure central feasibility and stability of the decentralized control scheme,

- *Decoupled Terminal Cost.* How does one choose the terminal cost l_N^i for each problem $\mathcal{P}_i$?
- *Decoupled Terminal Region.* How does one choose the terminal region $\mathcal{X}_f^i$ for each problem $\mathcal{P}_i$?
- *Feasibility Issue.* Is it enough to choose the right decoupled terminal cost and terminal region?

We can anticipate here that the answer to the "feasibility issue" is negative. That is, a good choice of l_N^i and $\mathcal{X}_f^i$ is, in general, not sufficient to ensure stability and feasibility of the decentralized scheme. Problem (12) needs to be modified in order to guarantee feasibility. Also, the "feasibility issue" is the most complex one, while computing decentralized costs and terminal regions is less complex since we have assumed that the systems are dynamically decoupled.

3.1 Decoupled Terminal Costs

Stability in not a major issue for decentralized schemes for dynamically decoupled systems. We will assume that the terminal cost in (10) has been chosen as the sum of N_v terminal cost functions $l_N^i(x^i, \tilde{x}^i)$ associated to each node and function of the node and its neighbors' states:

$$l_N(\tilde{x}) = \sum_{i=1}^{N_n} l_N^i(x^i, \tilde{x}^i) \tag{14}$$

For instance for linear systems and quadratic objective function, it would be enough to use as terminal cost function the sum of the cost functions associated to the LQR designed for each independent system and its neighbors.

3.2 Decoupled Terminal Regions

The problem of the terminal set can be approached in two different ways. One can start from the terminal set $\mathcal{X}_f$ in problem (10) and decompose it into N_v non-empty sets $\mathcal{X}_f^i \subset \mathbb{R}^{n^i}$ which will be used in (12). The N_v sets $\mathcal{X}_f^i \subset \mathbb{R}^{n^i}$ can be also computed without taking into consideration the original invariant set $\mathcal{X}_f$. We prefer to follow the latter route for two main reasons; *(i)* it can be computationally prohibitive to compute the invariant set $\mathcal{X}_f$ in (10) for a large team of systems, *(ii)* it is difficult to decompose the invariant set $\mathcal{X}_f$ into N_v terminal sets, which used in (12) will guarantee the feasibility of the decentralized control schemes.

We propose the following construction of the sets $\mathcal{X}_f^i$. For each vehicle, we compute an hyper-rectangular inner approximation of the feasible space defined by the interconnection constraints which contains the equilibria $(x_e^i, \tilde{x}_e^i)$ as follows. Consider the i-th node and the set $S^{i,j} \subset \mathbb{R}^{n^i+n^j}$ for $(i,j) \in \mathcal{A}$ defined by the coupling constraints $g^{i,j}$:

$$S^{i,j} = \{x^i \in \mathbb{R}^{n^i},\ x^j \in \mathbb{R}^{n^j} \mid g^{i,j}(x^i, x^j) \leq 0\}.$$

Compute the sets $I_{i,j}^i$ and $I_{i,j}^j$ satisfying

$$I_{i,j}^i \times I_{i,j}^j \subseteq S^{i,j}$$

Let $I^i = \bigcap_{(i,j)\in\mathcal{A}} I_{i,j}^i$ and $\mathcal{X}_f^i$ be a controlled invariant set of the i-th system (1), subject to input and state constraints (2) and to the additional constraint $x_k^i \in I^i\ \forall k \geq 0$.

Through the procedure described above one can independently compute N_v terminal sets $\mathcal{X}_f^i$ which will be use in problem (12). Such sets have the following property. If each system enters its associated terminal set, we are ensured that all the interconnection constraints are satisfied and that there exists a decentralized control law which keeps each one in its respective terminal set. In the worst case each I^i will coincide with the equilibrium x^i. The sum of the ratios between the volumes of $I_{i,j}^i \times I_{i,j}^j$ and $S^{i,j}$ for all $(i,j) \in \mathcal{A}$ will be a good measure of the conservativeness of the method. The smaller this sum is, the smaller will be the region of attraction of the decentralized control scheme. Note that the sets I^i and I^j might be convex even if $S^{i,j}$ is not convex.

3.3 Ensuring Feasibility

We have mentioned that feasibility of the decentralized trajectories is the main issue in decentralized control schemes. In this section we discuss some modification to the original problem which can ensure feasibility.

3.3.1 Robust Constraint Fulfillment

Consider the coupling constraints of problem $\mathcal{P}_i$ at step k

$$g^i(x_{k,t}^i, \tilde{x}_{k,t}^i) \leq 0 \tag{15}$$

and by using the state update equations

$$\begin{aligned} x_{k+1,t}^i &= f^i(x_{k,t}^i, u_{k,t}^i), \quad k \geq 0 \\ x_{k+1,t}^j &= f^j(x_{k,t}^j, u_{k,t}^j), \quad (j,i) \in \mathcal{A},\ k \geq 0 \end{aligned} \tag{16}$$

rewrite them as

$$g_k^i(x_t^i, \tilde{x}_t^i, u_{[0,\ldots,k-1]}^i, \tilde{u}_{[0,\ldots,k-1]}^i) \leq 0 \tag{17}$$

where $u_{[0,\ldots,k-1]}^i \triangleq \{u_0^i, \ldots, u_{k-1}^i\}$ and $\tilde{u}_{[0,\ldots,k-1]}^i \triangleq \{\tilde{u}_0^i, \ldots, \tilde{u}_{k-1}^i\}$. In order to ensure the feasibility of the team system, a possible approach is to "robustify" the constraints (17) for all vehicles at all time steps. In other words, we can require that the coupling constraints at each node are satisfied for *all* possible behaviors of the neighboring nodes, once their initial condition is known. Therefore, the vector $\tilde{u}^i_{[0,\dots,k-1]}$ can be considered as a disturbance which can lead to possible infeasibility of constraint (17). There are two possible schemes: open-loop and closed-loop constraint fulfillment. An open-loop robust constraint fulfillment is formulated next. Substitute the functions g^i_k with $\bar{g}^i_k : \mathbb{R}^{n^i} \times \mathbb{R}^{\tilde{n}^i} \times \mathbb{R}^{m^i} \to \mathbb{R}^{n^i}$ where

1. For all $x^i_t, \tilde{x}^i_t, u^i_{[0,\dots,k-1]}$ which satisfy

$$\bar{g}^i_k(x^i_t, \tilde{x}^i_t, u^i_{[0,\dots,k-1]}) \leq 0 \tag{18}$$

   we have

$$g^i_k(x^i_t, \tilde{x}^i_t, u^i_{[0,\dots,k-1]}, \tilde{u}^i_{[0,\dots,k-1]}) \leq 0$$

   for all admissible[1] $\tilde{u}^i_{[0,\dots,k-1]}$.

2. The sets described by

$$\bar{g}^i_k(x^i_t, \tilde{x}^i_t, u^i_{[0,\dots,k-1]}) \leq 0 \tag{19}$$

   for $i = 1, \dots, N_v$ $k = 1, \dots, N-1$ are nonempty.

Robust closed loop formulation [16] is less conservative but more computationally involved. We will not describe the details of the robust closed loop formulation for a simple reason. "Robust constraint fulfillment" applied to decentralized control schemes results in a very conservative approach even for the closed-loop case. For instance, consider the case of formation flight. Assume we have only two aircraft and we want to design a local controller on the first aircraft using robust constraint fulfillment. The worst case scenario will include, in most cases, the collision of the two aircraft if they are not very far from each other and if they have the same dynamics and constraints. However in reality, neighboring vehicles collaborate between each other to fly in formation.

3.3.2 Reducing Conservativeness

A less conservative approach for ensuring feasibility of the decentralized scheme has to take into consideration that systems in a team are cooperating, and therefore the trajectory that a node is predicting should not be extremely different form what its neighbors are executing. This idea can be formulated in several ways. For instance, one could allow the exchange of optimizers between the nodes in order to try to be as close as possible to what the neighboring system has predicted about a certain node. Another possibility is to tighten the coupling constraints (6) by a quantity which is an indirect measure of the cooperativeness of the team [14]

$$g^i_k(x^i_t, \tilde{x}^i_t, u^i_{[0,\dots,k-1]}, \tilde{u}^i_{[0,\dots,k-1]}) \leq \epsilon^i_k \tag{20}$$

where $\epsilon^i_k \leq 0$ is a new optimization variable. The variables ϵ^i_k might be computed off-line, based on a priori knowledge of the team behavior or could be used in the following two stage process. In the first stage of the optimization problems (12), the coupling constraints are substituted with

[1]admissible inputs have to satisfy constraints (2)

the one in (20). Their parametric solution [17] with respect to ϵ_k^i yield the optimizer function $u^{*i}(\epsilon_0^i, \ldots, \epsilon_N^i)$. In a second stage the nodes communicate between themselves in order to agree on a set of $\bar{\epsilon}_k^i$ for $i = 1, \ldots, N_v,\ k = 1, \ldots, N$ which ensures feasibility of the decentralized trajectories. If the agreement algorithm ends with a positive answer, each vehicle will implement $u^{*i}(\bar{\epsilon}_0^i, \ldots, \bar{\epsilon}_N^i)$.

3.4 Conservativeness and Stability Proofs

The schemes discussed in the previous sections have not been described in detail, since a formal stability proof of these schemes which is not too conservative is still under investigation. Our experience is that the more complex the decentralized control scheme is, the more difficult it is to give any stability or feasibility proofs. As in most of the MPC literature, such decentralized schemes work very well in practice even if there is any "theoretical stability proof". In Section 5 we will show some results when the main decentralized scheme is applied to formation flight. We will also point out some interesting behavior of the decentralized scheme which is different form what is observed in standard centralized MPC control theory.

4 Distributed Control Scheme

In this section we propose a distributed MPC control scheme. The intention is to modify the decentralized approach presented in Section 3 in a particular way that supports feasibility by allowing nodes to communicate their optimal solutions between each other. We will assume that the nodes of the interconnection graph can be grouped into N_P ($\leq N_v$) sets to which a certain priority is assigned. The idea behind imposing a prioritized graph structure is that each node will compute the optimal control input for the lower priority neighboring nodes, while checking the constraint satisfaction and implementing the solution of the higher priority neighboring node (parent). In contrast to the approach of Section 3, not all the nodes solve optimization problems: the computation of the optimal strategy is distributed over the parent nodes of the graph.

A priority ($p = 1, \ldots, N_P$) is assigned to every node in the following way. Each node with priority p can only be connected to nodes with priorities $p-1, p$ and $p+1$. Furthermore, it can be connected to only one node with higher priority $p-1$, which will be referred to as the parent node. The lower priority neighbors will be referred to as children. A node can be connected to other p priority nodes only if they have the same parent as the node itself. These will be called sibling nodes.

Such a prioritization of nodes can always be assigned to a tree graph, where N_P is limited by the longest path in the tree, however in general the assignment of priorities is not unique. The interconnection graph could be more complex than a tree. It could contain circles that consist of a parent node and a subset of its children or just a subset of the "siblings" themselves.

Each arc in this particular prioritized graph structure belongs to one of two categories. It either connects a parent node with one of its children, or it connects siblings with each other. The proposed control strategy assumes that arcs between a parent node and its descendants denote information exchange and constraints as well, as it was introduced before in Sections 1 and 2. However, arcs between siblings will only signify the existence of constraints between those nodes and not the exchange or propagation of information. Figure 1 illustrates the general

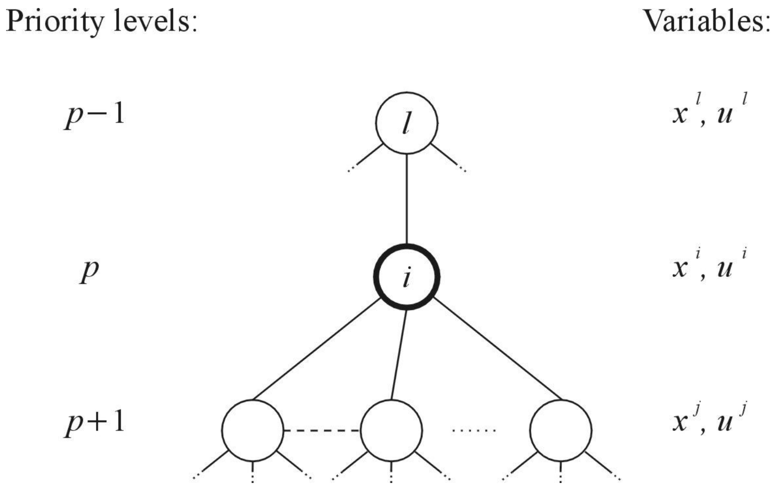


Figure 1: General subproblem of a single node.

subproblem for the i-th node with priority level p.

Let $\mathcal{A}^{iC}$ denote the set of arcs between the i-th node and its children and $\mathcal{A}^{iS}$ denote the set of arcs connecting the i-th node to its siblings. If $\mathcal{V}^p$ denotes the N_p number of nodes with priority p, then $\mathcal{A}^{iC} \subseteq \mathcal{V}^p \times \mathcal{V}^{p+1}$ and $\mathcal{A}^{iS} \subseteq \mathcal{V}^p \times \mathcal{V}^p$. The overall graph structure under consideration can be represented as

$$\mathcal{G} = \{\mathcal{V}, \{\mathcal{A}^{[1,\ldots,N_v]C}, \mathcal{A}^{[1,\ldots,N_v]S}\}\} \tag{21}$$

Clearly, this interconnection structure implies that there is a single "ultimate" leader L of highest priority ($p = 1$) with $\mathcal{A}^{LS} = \emptyset$ and that there is a certain number of nodes (referred to as leaves), which do not have any children. These are not necessarily "real" leaf nodes using graph theoretical terminology, their designation is implied only by the chosen prioritization.

Consider the systems (1), a graph $\mathcal{G}$ with the special structure described above, and the following finite time optimal control problem for node i of priority level $p > 1$.

$$(\hat{\mathcal{P}}_t^i): \ \hat{J}^{i*}(x_t^i, x_t^j, \tilde{U}_t^i, u_{0,t}^j) \triangleq \min_{U_t^j, (j,i) \in \mathcal{A}^{iC}} \sum_{k=0}^{N-1} l^i(x_{k,t}^i, u_{k,t}^i, x_{k,t}^j, u_{k,t}^j) + l_N^i(x_{N,t}^i, x_{N,t}^j) \tag{22a}$$

$$\text{subj. to} \begin{cases} x_{k+1,t}^i = f^i(x_{k,t}^i, u_{k,t}^i), \quad k \geq 0 \\ x_{k,t}^i \in \mathcal{X}^i, \quad u_{k,t}^i \in \mathcal{U}^i, \quad k = 1, \ldots, N-1 \\ x_{k+1,t}^j = f^j(x_{k,t}^j, u_{k,t}^j), \ k \geq 0 \\ x_{k,t}^j \in \mathcal{X}^j, \quad u_{k,t}^j \in \mathcal{U}^j, \ k = 1, \ldots, N-1 \\ \qquad (j,i) \in \mathcal{A}^{iC}, \\ g_k^{i,j}(x_t^i, x_t^j, u_{[0,\ldots,k-1]}^i, u_{[0,\ldots,k-1]}^j) \leq 0, \\ \qquad (j,i) \in \mathcal{A}^{iC}, \ k = 1, \ldots, N-1 \\ g_k^{q,r}(x_t^q, x_t^r, u_{[0,\ldots,k-1]}^q, u_{[0,\ldots,k-1]}^r) \leq 0, \\ \qquad (q,i) \in \mathcal{A}^{iC}, \ (r,i) \in \mathcal{A}^{iC}, \\ \qquad (q,r) \in \mathcal{A}^{rS} \text{ and } \mathcal{A}^{qS}, \ k = 1, \ldots, N-1 \\ x_{N,t}^j \in \mathcal{X}_f^j, \\ x_{0,t}^i = x_t^i, \\ x_{0,t}^j = x_t^j, \\ \tilde{U}_t^i = [U_{t-1}^{i*}, v^i], \\ u_{0,t}^j = \text{const} \end{cases} \tag{22b}$$

where $U_t^j \triangleq [u_{1,t}^j, \ldots, u_{N-1,t}^j]' \in \mathbb{R}^{m^j(N-1)}$ denotes the optimization vector of problem $\hat{\mathcal{P}}_t^i$. The value of $u_{0,t}^j$ is assumed to be a known constant (which is the first control value of the previous time solution of $\hat{\mathcal{P}}_{t-1}^i$, otherwise zero at times $t < p$). The control values $\tilde{U}_t^i \triangleq [u_{0,t}^i, \ldots, u_{N-1,t}^i]' = [U_{t-1}^{i*}, v^i] = [u_{1,t-1}^{i*} \ldots, u_{N-1,t-1}^{i*}, v^i]' \in \mathbb{R}^{m^i N}$ are obtained from the optimization vector solution of the higher priority (parent) problem $\hat{\mathcal{P}}_{t-1}^l$, $(i,l) \in \mathcal{A}^{lC}$; padded with a control value v^i that ensures $x_N^i \in \mathcal{X}_f^i$ at the end of the horizon. Such a feasible control input exists, since the terminal region $\mathcal{X}_f^i$ was defined to be control invariant.

The proposed distributed scheme represents the following strategy. Consider a node i and the associated problem $\hat{\mathcal{P}}_t^i$. Given an optimal control sequence $\tilde{U}_t^i$ for node i, which is calculated by its parent node, the i-th node formulates and solves a finite time optimal control problem to obtain control sequences for its children. Node i will be implementing a control sequence $\tilde{U}_t^i$ received from its parent and the children of node i will be implementing the optimal control solutions U_t^{j*} calculated and transmitted by node i assuming a one time step communication delay. These optimal control solutions for the children of node i have to respect parent-children constraints represented by $g_k^{i,j}$ in (22b) and any sibling constraints that exist between the children of node i, denoted by $g_k^{q,r}$. Since leaf nodes do not possess any children, they do not perform any calculations and there is no optimization problem assigned to them.

The optimization problem for node number 1 (ultimate leader) is slightly different from the general problem of "follower-nodes" in (22), since it needs to solve for its own control solution vector as well, not only for the solution of the followers. The ultimate leader node solves the following finite time optimal control problem.

$$(\hat{\mathcal{P}}_t^L): \ \hat{J}^{L*}(x_t^L, x_t^j, u_{0,t}^j) \triangleq \min_{\tilde{U}_t^L, U_t^j, (j,L)\in\mathcal{A}^{LC}} \sum_{k=0}^{N-1} l^L(x_{k,t}^L, u_{k,t}^L, x_{k,t}^j, u_{k,t}^j) + l_N^L(x_{N,t}^L, x_{N,t}^j) \quad (23a)$$

$$\text{subj. to} \begin{cases} x_{k+1,t}^L = f^L(x_{k,t}^L, u_{k,t}^L), \quad k \geq 0 \\ x_{k,t}^L \in \mathcal{X}^L, \quad u_{k,t}^L \in \mathcal{U}^L, \quad k = 1, \dots, N-1 \\ x_{k+1,t}^j = f^j(x_{k,t}^j, u_{k,t}^j), \ k \geq 0 \\ x_{k,t}^j \in \mathcal{X}^j, \quad u_{k,t}^j \in \mathcal{U}^j, \ k = 1, \dots, N-1 \\ \qquad (j, L) \in \mathcal{A}^{LC}, \\ g_k^{L,j}(x_t^L, x_t^j, u_{[0,\dots,k-1]}^L, u_{[0,\dots,k-1]}^j) \leq 0, \\ \qquad (j, L) \in \mathcal{A}^{LC}, \ k = 1, \dots, N-1 \\ g_k^{q,r}(x_t^q, x_t^r, u_{[0,\dots,k-1]}^q, u_{[0,\dots,k-1]}^r) \leq 0, \\ \qquad (q, L) \in \mathcal{A}^{LC}, \ (r, L) \in \mathcal{A}^{LC}, \\ \qquad (q, r) \in \mathcal{A}^{rS} \ \text{and} \ \mathcal{A}^{qS}, \ k = 1, \dots, N-1 \\ x_{N,t}^L \in \mathcal{X}_f^L, \\ x_{N,t}^j \in \mathcal{X}_f^j, \\ x_{0,t}^L = x_t^L, \\ x_{0,t}^j = x_t^j, \\ u_{0,t}^j = \text{const} \end{cases} \quad (23b)$$

Figure 2 illustrates the propagation of the solution in a simple case where the nodes are connected as a string, following one after another.

5 Examples

This section presents simulation examples of the decentralized control scheme (12)-(13) described in Section 3. The examples describe formation flight of vehicles flying at a certain altitude. Each vehicle is modeled as a point mass in two dimensions with constraints on states and inputs. The coupling between vehicles stems from the common objective of the team (moving in formation) and its constraints (vehicles are not allowed to violate each others protection zones).

Our intention is to provide some insight to feasibility issues associated with the proposed decentralized scheme through a few simulation scenarios. We will describe each simulation scenario first and then summarize our observations in Section 5.3.

The dynamics (1) of the i-th vehicle is obtained by discretizing a double integrator at 5 Hz

$$x_{k+1}^i = \overbrace{\begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{A} x_k^i + \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}}^{B} u_k^i \quad (24)$$

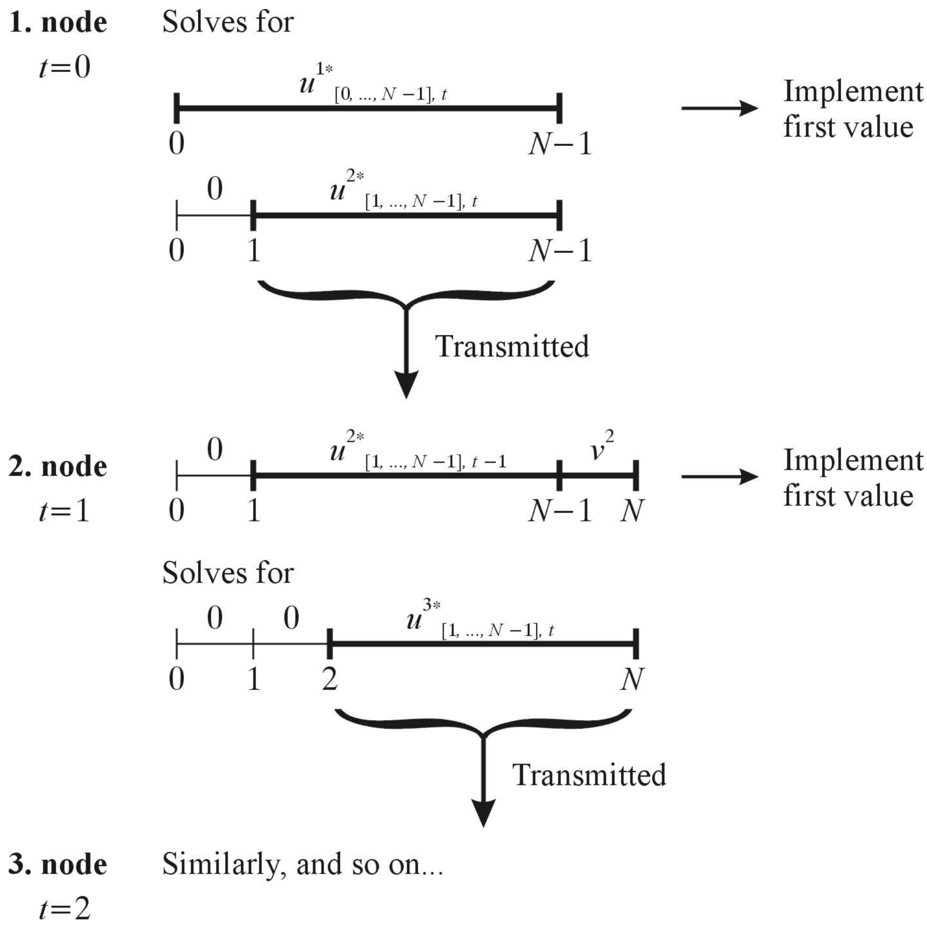


Figure 2: Propagation of the MPC subproblem solutions.

where

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

and $x_{k,pos}^i$ is the vector of x and y coordinates and $x_{k,vel}^i$ denotes a vector of x-axis and y-axis velocity components.

Each node solves the decentralized optimization problem (12) with:

1. Linear cost function:

$$
\begin{aligned}
l^i(x^i_{k,t}, u^i_{k,t}, \tilde{x}^i_{k,t}, \tilde{u}^i_{k,t}) = &\left\|Q_u[u^i_{k,t}, \tilde{u}^i_{k,t}]'\right\|_\infty + \\
&\max\Bigg(\left\|Q_{pos}\left(x^i_{k,pos} - x^i_{f,pos}\right)\right\|_\infty, \left\|Q_{vel}\left(x^i_{k,vel} - x^i_{f,vel}\right)\right\|_\infty, \\
&\max_{j,(i,j)\in\mathcal{A}} \left\|\tilde{Q}_{pos}\left(x^j_{k,pos} - x^j_{f,pos}\right)\right\|_\infty, \max_{j,(i,j)\in\mathcal{A}} \left\|\tilde{Q}_{vel}\left(x^j_{k,vel} - x^j_{f,vel}\right)\right\|_\infty, \\
&\max_{j,(i,j)\in\mathcal{A}} \left\|Q_{rpos}\left(\left(x^i_{k,pos} - x^j_{k,pos}\right) - \left(x^i_{f,pos} - x^j_{f,pos}\right)\right)\right\|_\infty, \\
&\max_{j,(i,j)\in\mathcal{A}} \left\|Q_{rvel}\left(\left(x^i_{k,vel} - x^j_{k,vel}\right) - \left(x^i_{f,vel} - x^j_{f,vel}\right)\right)\right\|_\infty, \\
&\max_{q,r,(i,q)\in\mathcal{A},(i,r)\in\mathcal{A}} \left\|\tilde{Q}_{rpos}\left(\left(x^q_{k,pos} - x^r_{k,pos}\right) - \left(x^q_{f,pos} - x^r_{f,pos}\right)\right)\right\|_\infty, \\
&\max_{q,r,(i,q)\in\mathcal{A},(i,r)\in\mathcal{A}} \left\|\tilde{Q}_{rvel}\left(\left(x^q_{k,vel} - x^r_{k,vel}\right) - \left(x^q_{f,vel} - x^r_{f,vel}\right)\right)\right\|_\infty \Bigg)
\end{aligned}
$$

2. No terminal cost and constraint: $l^i_N(x^i_{N,t}, \tilde{x}^i_{N,t}) = 0, \quad \mathcal{X}_f \equiv \mathbb{R}^4$
3. Identical vehicle dynamics (24)
4. Linear constraints on states and inputs:

$$
|x| \leq \begin{bmatrix}1000 & 1000 & 24 & 24\end{bmatrix}', \quad |u| \leq \begin{bmatrix}2 & 2\end{bmatrix}'
$$

5. Non-convex interconnection constraints:

$$
\begin{aligned}
g^{i,j}(x^i_k, u^i_k, x^j_k, u^j_k) &= \left\|x^i_{k,pos} - x^j_{k,pos}\right\|_\infty \geq d_{min} \\
g^{q,r}(x^q_k, u^q_k, x^r_k, u^r_k) &= \left\|x^q_{k,pos} - x^r_{k,pos}\right\|_\infty \geq d_{min}
\end{aligned}
$$

Note that the cost function above includes terms that weigh the maximum control effort and the infinity norm of a vector which collects all the absolute and relative errors of the i-th node and its neighbors with respect to the final reference values. The infinity norm of a vector is defined as $\|v\|_\infty \triangleq \max_i |v_i|$, where $v = [v_1\ v_2\ \cdots\ v_n]'$.

Notice also, that the interconnection constraints in item 5 define square protection zones around vehicles that cannot intersect each other. Solutions generated by node i enforce these collision avoidance constraints not only between itself and its neighbors, but among the neighbors as well.

The specific parameters of the problem will be given further in this section, along with the number of vehicles and graph structure corresponding to different scenarios.

The above choice of dynamics, cost and constraints allow us to rewrite problem 12 as a Mixed Integer Linear Program (MILP) [18, 19], for which efficient branch-and-bound solvers are available [20]. Note that any other linear or piecewise linear formulation of constraints, cost and dynamics can be cast as an MILP [18].

Next, we simulate two different scenarios where each node follows the decentralized MPC control scheme described in steps 1–4 of (13), including comparisons with a centralized MPC scheme.

5.1 Three-vehicle scenarios

The first set of simulations was conducted using three vehicles arranged in the graph structure shown in Figure 3. Such an interconnection graph implies that the decentralized problem solved

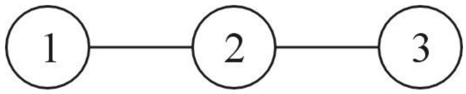


Figure 3: Three-vehicle formation graph.

by vehicle #1 is a function of its own states and the states of vehicle #2 only. The second vehicle "sees" all three vehicles and would in fact be solving a centralized problem if the other two vehicles were implementing the solution it calculates. The last vehicle #3 is in the same situation as vehicle #1 by knowing only about its neighbor #2. The objective of the team is to get from their initial positions to designated target points while taking up the associated formation defined by the relative positions and speeds at the targets. The protection zone of each vehicle is given as $d_{min}/2 = 1.2$.

The three vehicles have to perform a maneuver specified by the following initial and final conditions:

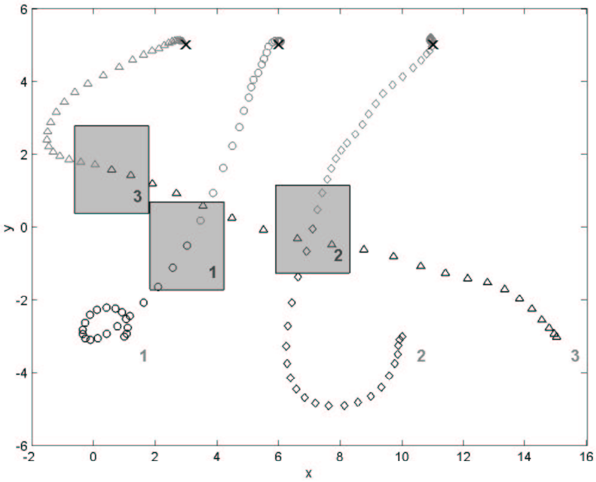
$$\begin{aligned} x_0^1 &= \begin{bmatrix}1 & -3 & 0 & 0\end{bmatrix}', & x_f^1 &= \begin{bmatrix}6 & 5 & 0 & 0\end{bmatrix}', \\ x_0^2 &= \begin{bmatrix}10 & -3 & 0 & 0\end{bmatrix}', & x_f^2 &= \begin{bmatrix}11 & 5 & 0 & 0\end{bmatrix}', \\ x_0^3 &= \begin{bmatrix}15 & -3 & 0 & 0\end{bmatrix}', & x_f^3 &= \begin{bmatrix}3 & 5 & 0 & 0\end{bmatrix} \end{aligned}$$

This setup intends to mimic a typical "challenging" conflict scenario where vehicles have to reach their final targets by crossing each others paths.

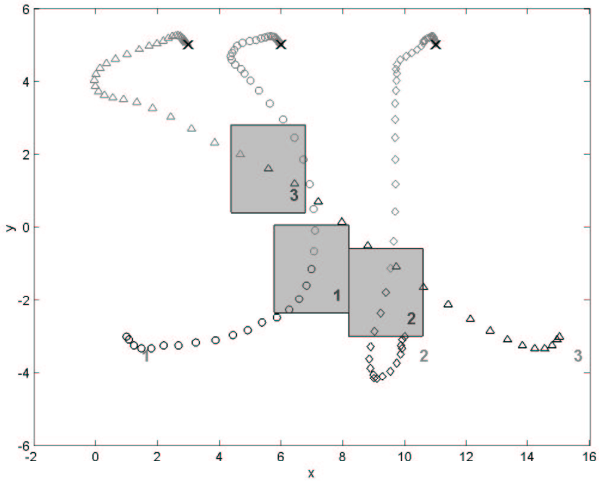
Feasible solutions

Figure 4(a) shows the resulting trajectories when the decentralized scheme (12)-(13) is applied to the problem using a prediction horizon length of 9 steps (1.8 seconds). Weights in the cost function were chosen to be $Q_u = 0.1I_4$ for vehicles #1, #3 and $Q_u = 0.1I_6$ for vehicle #2, where $I_n \in \mathbb{R}^n$ denotes the identity matrix. The dimension of the weight Q_u is determined by how many vehicles and control inputs are involved in the optimization problem (e.g. vehicle #1 solves for its own two control inputs and the two inputs of its single neighbor, which means $Q_u \in \mathbb{R}^4$). Other weights had equal values of $Q_{pos} = Q_{vel} = \tilde{Q}_{pos} = \tilde{Q}_{vel} = Q_{rpos} = Q_{rvel} = \tilde{Q}_{rpos} = \tilde{Q}_{rvel} = 100I_2$.

It is interesting to observe that the decentralized scheme shows signs of a collective behaviour that could be attributed to an intuitive centralized solution. Even though vehicle #1 cannot see vehicle #3 and would almost certainly collide with it by simply flying towards its target, the collective motion of vehicle #1 and #2, induced by their relative position objective, yields to vehicle #3 by moving away or hovering, while #3 is speeding towards its target.



(a) Decentralized solution at $t = 4.4$ sec.



(b) Centralized solution at $t = 3.4$ sec.

Figure 4: Decentralized and centralized solutions of the three-vehicle example. Final targets are denoted by X's. The three shaded boxes represent protection zones associated with each vehicle.

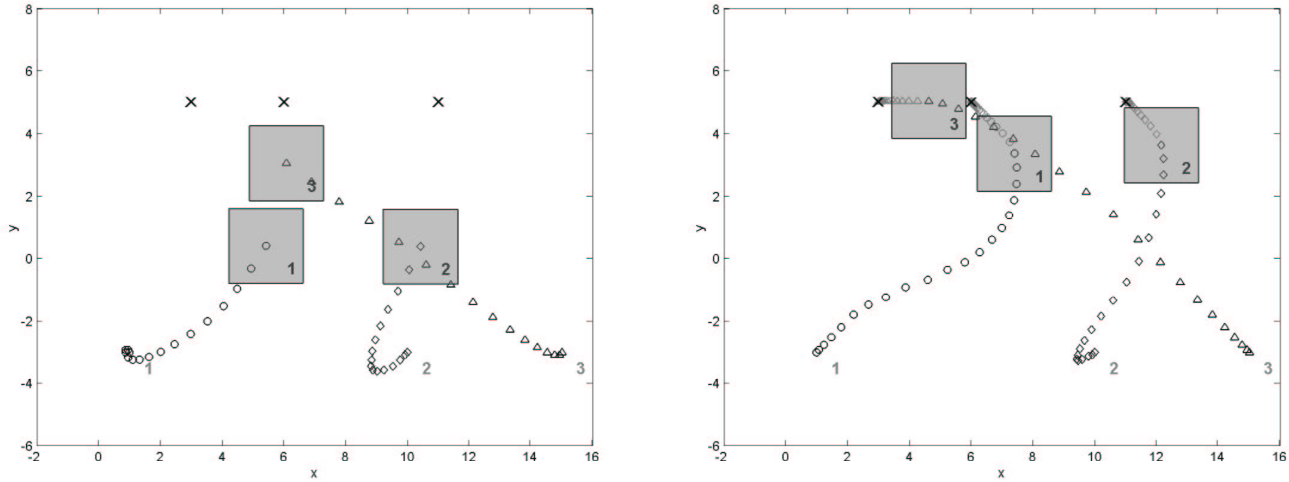
In order to offer a baseline for evaluating the decentralized solution, the same problem was solved using the centralized scheme (10)-(11). The solution is depicted in Figure 4(b).

This example shows that a decentralized scheme can find reasonable solutions to cooperative problems even though feasibility can be compromised depending on initial conditions of the vehicles. The size of the protection zones have a significant influence on overall feasibility and the quality of solutions as well.

Changing horizon length

The next simulation intends to demonstrate that the role of the prediction horizon length can be quite different from what standard MPC theory would suggest, mainly because of the decentralized nature of the problem. This means, for instance, that longer horizon lengths do not necessarily provide a better solution in general, since predictions about the future behaviour of neighboring vehicles can be completely inaccurate. The example shown in Figure 5(a) demonstrates this phenomenon by changing the horizon length to 14 steps (2.8 seconds) in the decentralized problem of Figure 4(a). The receding horizon problem of vehicle #2 becomes infeasible at 3.2 seconds, since due to inaccurate knowledge about the neighbors' future intentions, the three vehicles reached a point, from where vehicles #3 and #1 cannot avoid each others protection zones.

Since the centralized approach (10) resembles a standard receding horizon MPC problem, we would anticipate that longer horizon lengths lead to better solutions. Figure 5(b) shows the centralized solution using increased horizon length, illustrating that this is in fact the case.



(a) Decentralized solution at $t = 3.2$ sec.

(b) Centralized solution at $t = 4$ sec.

Figure 5: Decentralized and centralized solutions using longer horizon length (vehicle #2 becomes infeasible at $t = 3.2$ seconds in the decentralized case).

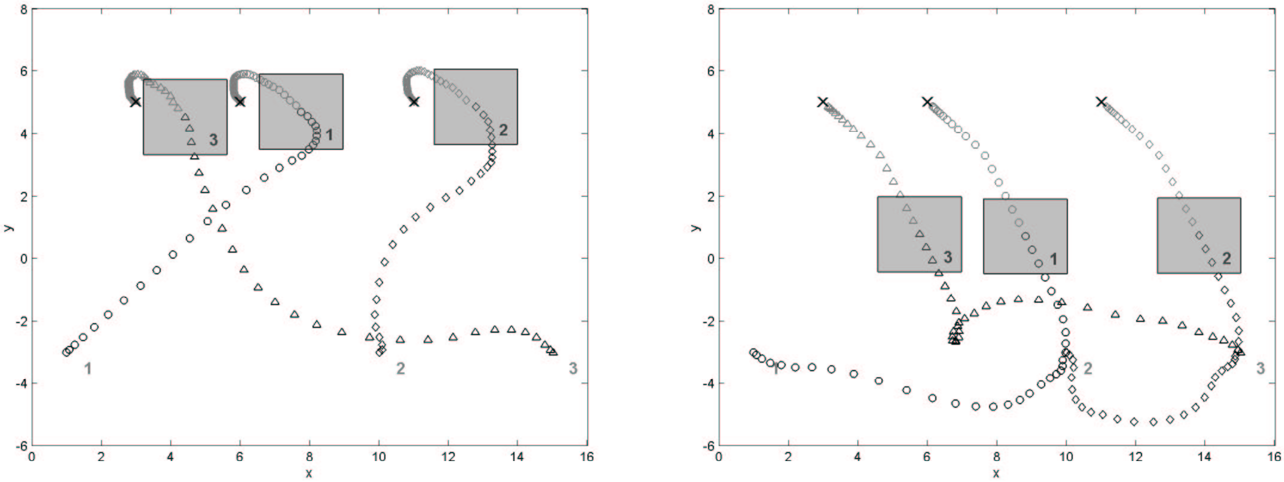
Changing cost weights

The following simulations intend to point out that feasibility of the decentralized problem without terminal cost and constraints is a function of the "strategy" that vehicles follow. This can be influenced by the selection of weights in the cost function.

The previous infeasible decentralized example for instance, can be made feasible if the relative state errors in the cost function are weighted much more heavily. This is obtained by setting $Q_{pos} = Q_{vel} = \tilde{Q}_{pos} = \tilde{Q}_{vel} = I_2$ and $Q_{rpos} = Q_{rvel} = \tilde{Q}_{rpos} = \tilde{Q}_{rvel} = 100I_2$. This setting prompts the vehicles to get into the desired formation first and then move together to the final target points as shown in Figure 6(a). This "strategy" seems to have a beneficial effect in the team's overall ability to perform the maneuver in a feasible way. The same phenomenon was observed in more complex, six-vehicle scenarios described in Section 5.2.

A possible explanation of this effect might be that vehicles are prompted to reach their desired relative states (formation) and resolve associated conflicts within a time frame that is comparable to their horizon lengths. Once the formation is attained, the common remaining goal of each vehicle is to "drift" to their target points. This at the same time becomes a much simpler objective to accomplish even in a decentralized way.

The centralized approach depicted in Figure 6(b) also serves to demonstrate that choosing larger weights on relative position results in an attractive alternative solution to the problem by reaching the desired formation before actually moving towards the final targets. Figure 6(b) shows the three vehicles "swapping position" relatively quickly, before "coasting" to their final positions together in formation. The selection of the weights was the same here as in the decentralized scheme.



(a) Decentralized solution at $t = 5.4$ sec.

(b) Centralized solution at $t = 6.8$ sec.

Figure 6: Decentralized and centralized solutions using longer horizon length and weighting relative state errors more.

5.2 Six-vehicle scenarios

In this section we consider scenarios with six vehicles. The problem setup is defined by assigning the graph shown in Figure 7 to the vehicles that are lined up one after another. The objective is to move the vehicles into a triangular formation given in Figure 7. Note that from a computational point of view, the decentralized problems solved by each vehicle have the same complexity as in the three-vehicle example of the previous section.

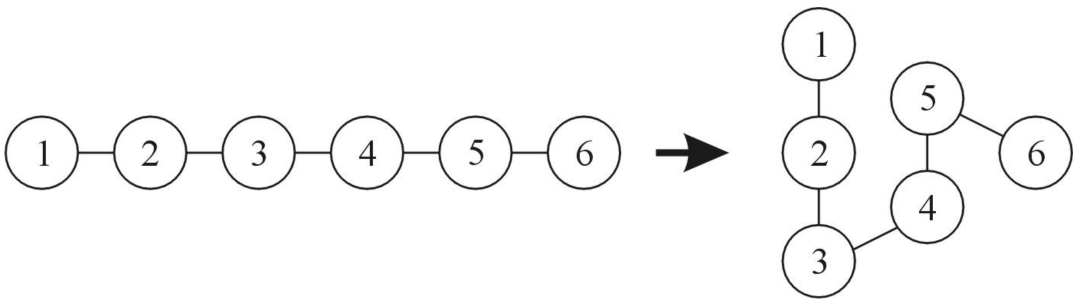


Figure 7: Six-vehicle formation graphs.

5.2.1 Stationary formation (surveillance)

In this example, the initial and final states of the individual vehicles are specified as follows

$$
\begin{aligned}
x_0^1 &= \begin{bmatrix}-10 & 0 & 0 & 0\end{bmatrix}', & x_f^1 &= \begin{bmatrix}-6 & 2 & 0 & 0\end{bmatrix}', \\
x_0^2 &= \begin{bmatrix}-8 & 0 & 0 & 0\end{bmatrix}', & x_f^2 &= \begin{bmatrix}-6 & 0 & 0 & 0\end{bmatrix}', \\
x_0^3 &= \begin{bmatrix}-6 & 0 & 0 & 0\end{bmatrix}', & x_f^3 &= \begin{bmatrix}-6 & -2 & 0 & 0\end{bmatrix}, \\
x_0^4 &= \begin{bmatrix}-4 & 0 & 0 & 0\end{bmatrix}', & x_f^4 &= \begin{bmatrix}-5 & -1 & 0 & 0\end{bmatrix}', \\
x_0^5 &= \begin{bmatrix}-2 & 0 & 0 & 0\end{bmatrix}', & x_f^5 &= \begin{bmatrix}-5 & 1 & 0 & 0\end{bmatrix}', \\
x_0^6 &= \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}', & x_f^6 &= \begin{bmatrix}-4 & 0 & 0 & 0\end{bmatrix}
\end{aligned}
$$

The protection zone of each vehicle is given as $d_{min}/2 = 0.3$. Horizon lengths were chosen to be 9 steps (1.8 seconds). Weights in the cost function had the same values as in the basic three-vehicle scenario. Figure 8 illustrates a feasible decentralized solution to this problem.

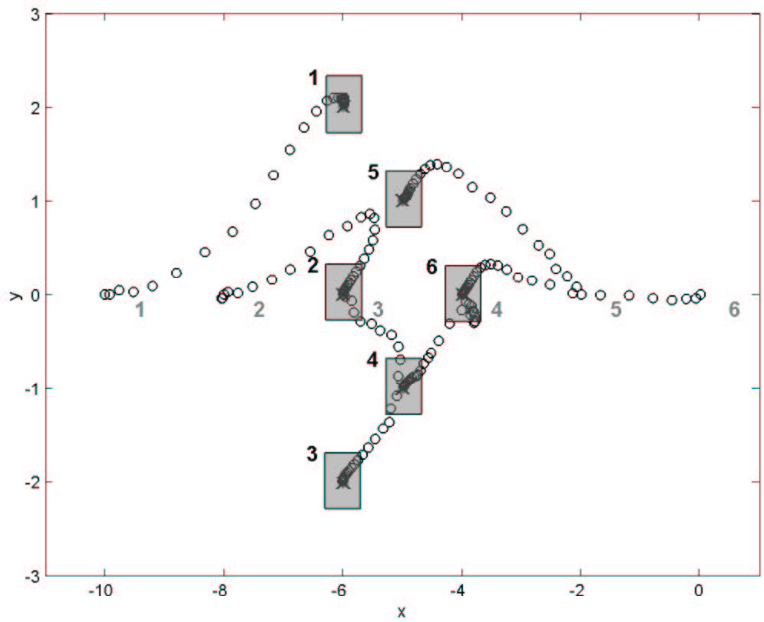


Figure 8: Decentralized solution of the stationary formation problem at $t = 6$ sec.

5.2.2 Moving formation

The stationary "surveillance-type" scenario of the previous section is modified in this example by assigning non-zero initial and final velocities to the vehicles:

$$\begin{aligned}
x_0^1 &= \begin{bmatrix}-10 & 0 & 5 & 0\end{bmatrix}', & x_f^1 &= \begin{bmatrix}-6 & 2 & 5 & 0\end{bmatrix}', \\
x_0^2 &= \begin{bmatrix}-8 & 0 & 5 & 0\end{bmatrix}', & x_f^2 &= \begin{bmatrix}-6 & 0 & 5 & 0\end{bmatrix}', \\
x_0^3 &= \begin{bmatrix}-6 & 0 & 5 & 0\end{bmatrix}', & x_f^3 &= \begin{bmatrix}-6 & -2 & 5 & 0\end{bmatrix}, \\
x_0^4 &= \begin{bmatrix}-4 & 0 & 5 & 0\end{bmatrix}', & x_f^4 &= \begin{bmatrix}-5 & -1 & 5 & 0\end{bmatrix}', \\
x_0^5 &= \begin{bmatrix}-2 & 0 & 5 & 0\end{bmatrix}', & x_f^5 &= \begin{bmatrix}-5 & 1 & 5 & 0\end{bmatrix}', \\
x_0^6 &= \begin{bmatrix}0 & 0 & 5 & 0\end{bmatrix}', & x_f^6 &= \begin{bmatrix}-4 & 0 & 0 & 0\end{bmatrix}
\end{aligned}$$

The terminal position states are excluded from the cost function by selecting $Q_{pos} = \tilde{Q}_{pos} = 0 \cdot I_2$, in order to allow the formation to maintain the final velocity while establishing the desired relative positions. The single purpose of specifying terminal positions is to indicate the desired final relative formation. Other weights were chosen as $Q_{vel} = Q_{rvel} = \tilde{Q}_{vel} = \tilde{Q}_{rvel} = 10I_2$ and $Q_{rpos} = \tilde{Q}_{rpos} = 50I_2$. The remaining parameters of the problem are the same as in the previous section.

A feasible decentralized solution of the moving and changing formation problem is shown in Figure 9. Note that using other initial conditions, the resulting maneuver might cause protection zone violations. However, it is important to emphasize that the decentralized problem is still feasible even if vehicles not linked by common neighbors or arcs cross each others path. If the chosen interconnection graph is not complete, the absence of connections between certain vehicles represents the lack of collision avoidance constraints between them. An incomplete graph for formation flight can be justified if unconnected vehicles fly at different altitudes or if the particular graph structure is chosen to represent a rigid formation [12].

5.3 Final observations and remarks

Simulation examples show that the decentralized approach to formation flight can provide feasible solutions even in challenging scenarios. Depending on the particular problem and initial conditions, feasibility issues might arise using the proposed scheme 12. A few examples were given to illustrate how the horizon length and weights in the cost function can influence the solution and feasibility of the decentralized problem. A number of alternative approaches similar to the hierarchical distributed scheme presented in Section 4 are currently under investigation to ensure feasibility in a decentralized way.

Another important aspect of the proposed framework is real-time implementability. The maximum computational time associated with a single decentralized subproblem at any sampling time was 0.063 seconds using a horizon length of 9 steps in the presented simulations. Furthermore, assuming a modest number of neighboring vehicles, explicit solutions of the underlying

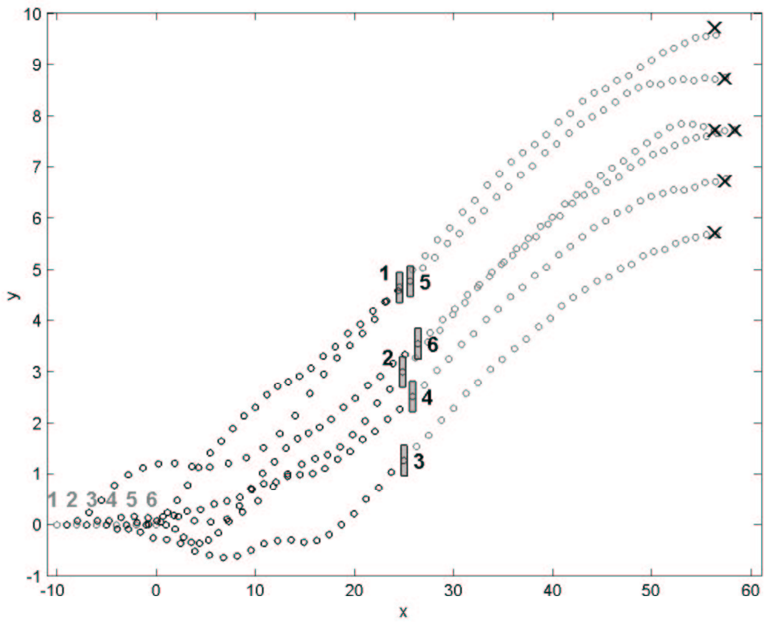


Figure 9: Decentralized solution of the moving formation scenario at $t = 4.8$ sec.

MILP problem can be computed off-line, which reduces the required number of calculations to a function evaluation [21].

For more comprehensive illustration of simulations we refer to videos accessible on the website [22].

References

[1] S. Wang and E. J. Davison. On the stabilization of decentralized control systems. *IEEE Trans. Automatic Control*, 18(5):473–478, 1973.

[2] V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

[3] M. Rotkowitz and S. Lall. Decentralized control information structures preserved under feedback. In *Proc. 41th IEEE Conf. on Decision and Control*, 2002.

[4] R. D'Andrea and G.E. Dullerud. Distributed control of spatially interconnected systems. *IEEE Trans. Automatic Control*, to appear.

[5] D. Stipanovic, G. Inalhan, R. Teo, and C. Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. In *Proc. 41th IEEE Conf. on Decision and Control*, 2002.

[6] W. Kang, N. Xi, and A. Sparks. Theory and applications of formation control in a perceptive referenced frame. In *Proc. 39th IEEE Conf. on Decision and Control*, 2000.

[7] Aveek Das, Rafael Fierro, Vijay Kumar, Jim Ostrowski, John Spletzer, and Camillo Taylor. A vision-based formation control framework. *IEEE Transaction on Robotics and Automation*, 18(5):813–825, 2002.

[8] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[9] F. Giulietti, L. Pollini L., and M. Innocenti. Autonomous formation flight. *IEEE Control Systems Magazine*, 20(6):34–44, 2000.

[10] C.J. Schumacher and R. Kumar. Adaptive control of UAVs in close-coupled formation flight. In *Proc. American Contr. Conf.*, pages 849–853, 2000.

[11] William B. Dunbar and Richard M. Murray. Model predictive control of coordinated multi-vehicle formation. In *Proc. 41th IEEE Conf. on Decision and Control*, 2002.

[12] Reza Olfati-Saber, William B. Dunbar, and Richard M. Murray. Cooperative control of multi-vehicle systems using cost graphs and optimization. In *Proc. American Contr. Conf.*, 2003.

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

[14] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, February 2002.

[15] D.Q. Mayne. Control of constrained dynamic systems. *European Jornal of Control*, 7:87–99, 2001.

[16] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automatic Control*, 43(8):1136–1142, 1998.

[17] A V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press, London, U.K., 1983.

[18] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[19] A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proc. American Contr. Conf.*, 2002.

[20] ILOG, Inc. *CPLEX 7.0 User Manual.* Gentilly Cedex, France, 2000.

[21] F. Borrelli. *Discrete Time Constrained Optimal Control.* PhD thesis, Automatic Control Labotatory - ETH, Zurich, 2002. `http://control.ethz.ch`.

[22] `http://www.aem.umn.edu/people/others/borrelli/ff.htm`.