

# A Decentralized Algorithm for Robust Constrained Model Predictive Control

Arthur Richards and Jonathan How  
Aerospace Control Laboratory  
Massachusetts Institute of Technology  
Cambridge MA 02139  
Email: {arthurr, jhow}@mit.edu

## ABSTRACT

A decentralized formulation is presented for model predictive control of systems with coupled constraints. The single large planning optimization is divided into small subproblems, each planning only for the states of a particular subsystem. Relevant plan data is exchanged between subsystems to ensure that all decisions are consistent with satisfaction of the coupled constraints. A typical application would be autonomous guidance of a fleet of UAVs, in which the systems are coupled by the need to avoid collisions, but each vehicle plans only its own path. The key property of the algorithm in this paper is that if an initial feasible plan can be found, then all subsequent optimizations are guaranteed to be feasible, and hence the constraints will be satisfied, despite the action of unknown but bounded disturbances. This is demonstrated in simulated examples, also showing the associated benefit in computation time.

**Keywords** Model Predictive Control, Decentralized Control

## I. INTRODUCTION

This paper presents a *decentralized* form of Model Predictive Control (DMPC) for systems comprised of multiple subsystems, each with independent dynamics and disturbances but with coupled constraints. By embedding the performance goals into the constraints, the DMPC algorithm can be used to design cooperative maneuvers for teams of vehicles, such as for formation-flight, collision avoidance and UAV arrival phasing.

MPC is a feedback control scheme in which a trajectory optimization is solved at each time step. The first control input of the optimal sequence is applied and the optimization is repeated at each subsequent step. Because the on-line optimization explicitly includes the operating constraints, MPC can operate closer to hard constraint boundaries than traditional control schemes. Centralized MPC has been widely developed for constrained systems [3], with many results concerning stability [4], [1] and robustness [10], [5], [11], and has been applied to the co-operative control of multiple vehicles [11], [13], [7], [8]. However, solving a single optimization problem for the entire team typically requires significant computation, which scales poorly with the size of the system (*e.g.* the number of vehicles in the team). To address this computational issue, attention has

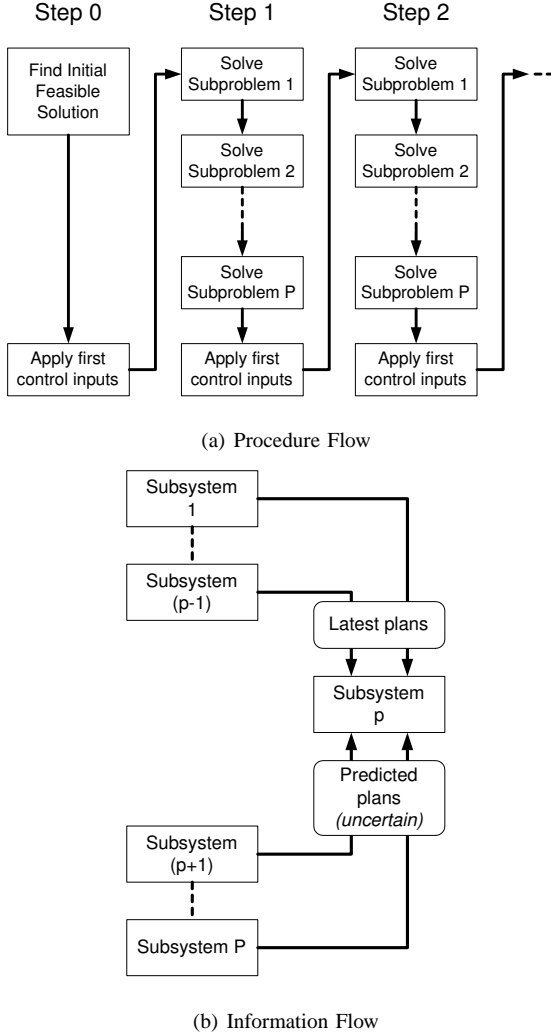
recently focused on decentralized MPC [17], [12] using a wide variety of approaches, including robustness to the actions of others [12], [16], penalty functions [9], [21], and partial grouping of computations [22]. The key point is that, when decisions are made in a decentralized fashion, the actions of each subsystem must be consistent with those of the other subsystems, so that decisions taken independently do not lead to a violation of the coupling constraints. The decentralization of the control is further complicated when disturbances act on the subsystems making the prediction of future behavior uncertain.

This paper presents a new approach to DMPC that addresses both of these difficulties. The key features of this algorithm are that each subsystem only solves a subproblem for its own plan, and each of these subproblems is solved only once per time step, without iteration. Under the assumption of a bounded disturbance, each of these subproblems is guaranteed to be feasible, thus ensuring robust constraint satisfaction across the group. The disturbance is accommodated by including “margin” in the constraints [11], tightening them in a monotonic sequence. The decentralized method employs at each time step a sequential solution procedure, outlined in Fig. 1(a), in which the subsystems solve their planning problems one after the other. The plan data relevant to the coupled constraints is then communicated to the other subsystems. The information requirements for a typical subproblem  $p$  are shown in Fig. 1(b). Each subproblem accommodates (a) the latest plans of those subsystems earlier in the sequence and (b) predicted plans of those later in the sequence. At initialization, it is necessary to find a feasible solution to the centralized problem, although this need not be optimal.

The paper begins by defining the problem for a general case, followed by the centralized MPC formulation. The division of this problem into decentralized subproblems is then developed. Finally, numerical simulations are included to compare the new decentralized approach with two other methods.

## II. PROBLEM STATEMENT

Consider the problem of controlling  $N_p$  subsystems. Each subsystem, denoted by subscript  $p \in \{1, \dots, N_p\}$ , has



**Fig.1:** Overview of Decentralized Algorithm

linear, time-invariant, discretized dynamics

$$\mathbf{x}_p(k+1) = \mathbf{A}_p \mathbf{x}_p(k) + \mathbf{B}_p \mathbf{u}_p(k) + \mathbf{w}_p(k) \quad (1)$$

$$\mathbf{y}_p(k) = \mathbf{C}_p \mathbf{x}_p(k) + \mathbf{D}_p \mathbf{u}_p(k) \quad (2)$$

where  $\mathbf{x}_p(k) \in \mathbb{R}^{N_p^x}$  is the state vector of subsystem  $p$  at time  $k$ ,  $\mathbf{u}_p(k) \in \mathbb{R}^{N_p^u}$  is the control input to subsystem  $p$  and  $\mathbf{w}_p(k) \in \mathbb{R}^{N_p^w}$  is the disturbance acting upon subsystem  $p$ . Finally,  $\mathbf{y}_p(k) \in \mathbb{R}^{N_p^y}$  is the output. Assume all subsystems  $(\mathbf{A}_p, \mathbf{B}_p)$  are controllable and the complete states  $\mathbf{x}_p$  are available.

The disturbances are unknown *a priori* but lie in independent bounded sets

$$\forall k, p \quad \mathbf{w}_p(k) \in \mathcal{W}_p \subset \mathbb{R}^{N_p^w} \quad (3)$$

The whole system is subjected to the following constraint upon the summed outputs, which may include both decoupled constraints, such as individual control magnitude limits and state constraints, and coupled constraints, such

as relative collision avoidance

$$\forall k \quad \sum_{p=1}^{N_p} \mathbf{y}_p(k) \in \mathcal{Y} \subset \mathbb{R}^{N_y} \quad (4)$$

The objective function is assumed to be decoupled between subsystems and therefore written as the following summation

$$J = \sum_{p=1}^{N_p} \sum_k \ell_p(\mathbf{x}_p(k), \mathbf{u}_p(k)) \quad (5)$$

A decoupled objective is typical of many problems of interest, such as fuel minimization for multiple spacecraft. This assumption is not strictly necessary for the robust feasibility result, which depends only on the constraints. However, decoupling the objectives is consistent with the approach taken here of using constraints to capture coupled behavior between subsystems. The requirement is to control the subsystems (1) and (2) such that the constraints (4) are satisfied for all disturbance sequences satisfying (3).

### III. CENTRALIZED ROBUST MPC PROBLEM

The online optimization approximates the complete problem in Section II by solving it over a finite horizon of  $T$  steps. A control-invariant terminal set constraint is applied to ensure stability [4]. Predictions are made using the nominal system model, *i.e.* (1) and (2) without the disturbance term. The output constraints are tightened in a monotonic sequence [11] to ensure robust feasibility, retaining a margin based on a particular nilpotent candidate policy.

Define the centralized optimization problem

$$\begin{aligned} & P_C(\mathbf{x}_1(k), \dots, \mathbf{x}_{N_p}(k)) : \\ J^* = & \min_{\substack{\mathbf{u}_1 \dots \mathbf{u}_{N_p}, \\ \mathbf{x}_1 \dots \mathbf{x}_{N_p}, \\ \mathbf{y}_1 \dots \mathbf{y}_{N_p}}} \sum_{p=1}^{N_p} \sum_{j=0}^T \ell_p(\mathbf{x}_p(k+j|k), \mathbf{u}_p(k+j|k)) \end{aligned}$$

subject to

$$\begin{aligned} \forall j \in \{0 \dots T\} \quad \forall p \in \{1 \dots N_p\} \\ \mathbf{x}_p(k+j+1|k) &= \mathbf{A}_p \mathbf{x}_p(k+j|k) + \mathbf{B}_p \mathbf{u}_p(k+j|k) \\ \mathbf{y}_p(k+j|k) &= \mathbf{C}_p \mathbf{x}_p(k+j|k) + \mathbf{D}_p \mathbf{u}_p(k+j|k) \\ \mathbf{x}_p(k|k) &= \mathbf{x}_p(k) \\ \mathbf{x}_p(k+T+1|k) &\in \mathcal{Q}_p \\ \sum_{p=1}^{N_p} \mathbf{y}_p(k+j|k) &\in \mathcal{Y}_{N_p}(j) \end{aligned}$$

where the constraint sets are found using the following recursions

$$\mathcal{Y}_{N_p}(0) = \mathcal{Y} \quad (6a)$$

$$\forall j \in \{0 \dots T\} \quad \forall p \in \{2 \dots N_p\} \\ \mathcal{Y}_{(p-1)}(j) = \mathcal{Y}_p(j) \sim (\mathbf{C}_p + \mathbf{D}_p \mathbf{K}_p(j)) \mathbf{L}_p(j) \mathcal{W}_p \quad (6b)$$

$$\forall j \in \{0 \dots T-1\} \\ \mathcal{Y}_{N_p}(j+1) = \mathcal{Y}_1(j) \sim (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{K}_1(j)) \mathbf{L}_1(j) \mathcal{W}_1 \quad (6c)$$

The operator “ $\sim$ ” represents the Pontryagin difference [14], defined by

$$\mathcal{A} \sim \mathcal{B} = \{\mathbf{a} \mid (\mathbf{a} + \mathbf{b}) \in \mathcal{A} \ \forall \mathbf{b} \in \mathcal{B}\}$$

and  $\mathbf{K}_p(j)$   $j \in \{0 \dots T-1\}$  are nilpotent candidate control laws for each subsystem, with associated state transition matrices  $\mathbf{L}_p(j)$  obeying

$$\begin{aligned} \mathbf{L}_p(0) &= \mathbf{I} \\ \forall j \in \{0 \dots T-1\} \\ \mathbf{L}_p(j+1) &= (\mathbf{A}_p + \mathbf{B}_p \mathbf{K}_p(j)) \mathbf{L}_p(j) \end{aligned}$$

The recursions (6) tighten the constraints at future plan steps, ensuring the existence of a “margin” to allow for future feedback action in response to disturbance. The tightening procedure is divided into separate steps for each subsystem. The reason for this will become apparent when the decentralized algorithm is presented in the next section. The centralized problem uses only the constraint sets for the last subsystem  $N_p$ . The choice of the candidate policies  $\mathbf{K}_p(j)$  is left to the designer, subject to the nilpotency requirement. The constraint tightening can be performed offline and, for polyhedral sets, the Pontryagin difference is implemented in the Matlab Invariant Set Toolbox [19].

The terminal constraint sets  $\mathcal{Q}_p$  are control invariant admissible sets, with associated control laws  $\kappa_p$  satisfying

$$\begin{aligned} \forall \mathbf{x}_p \in \mathcal{Q}_p \ \forall p \in \{1 \dots N_p\} \\ \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p \kappa_p(\mathbf{x}_p) \in \mathcal{Q}_p \end{aligned} \quad (7a)$$

$$\begin{aligned} \forall (\mathbf{x}_1^T \dots \mathbf{x}_{N_p}^T)^T \in \{\mathcal{Q}_1 \times \dots \times \mathcal{Q}_{N_p}\} \\ \sum_{p=1}^{N_p} \{\mathbf{C}_p \mathbf{x}_p + \mathbf{D}_p \kappa_p(\mathbf{x}_p)\} \in \mathcal{Y}_{N_p}(T) \end{aligned} \quad (7b)$$

It can be shown that solving this optimization in an MPC scheme guarantees robust feasibility [11].

#### IV. DECENTRALIZED MPC ALGORITHM

This section describes a decentralized algorithm for solving the problem in Section II. The centralized optimization from Section III is divided into  $N_p$  subproblems, each involving the trajectory of only one subsystem. Fig. 1 shows an outline of the resulting algorithm. The subproblems are solved one after the other, but the overall scheme is not iterative: each subproblem is solved once per time step and is guaranteed to be feasible. An outline of the proof of robust feasibility is given at the end of this section. The constraint sets in (6) included tightening from one subproblem to another. This is exploited here to incorporate a margin for uncertainty in the predicted behaviour of other subsystems, as illustrated in Fig. 1(b).

Define the  $p^{\text{th}}$  subproblem, in which the sequence of output vectors  $\tilde{\mathbf{y}}_p(k, \dots, k+T|k)$ , encoding the intentions of the other subsystems, and the current state of subsystem  $p$

are parameters

$$\begin{aligned} P_p(\mathbf{x}_p(k), \tilde{\mathbf{y}}_p(k \dots k+T|k)) : \\ J_p^* = \min_{\mathbf{u}_p, \mathbf{x}_p, \mathbf{y}_p} \sum_{j=0}^T \ell_p(\mathbf{x}_p(k+j|k), \mathbf{u}_p(k+j|k)) \end{aligned}$$

subject to

$$\begin{aligned} \forall j \in \{0 \dots T\} \\ \mathbf{x}_p(k+j+1|k) &= \mathbf{A}_p \mathbf{x}_p(k+j|k) + \mathbf{B}_p \mathbf{u}_p(k+j|k) \\ \mathbf{y}_p(k+j|k) &= \mathbf{C}_p \mathbf{x}_p(k+j|k) + \mathbf{D}_p \mathbf{u}_p(k+j|k) \\ \mathbf{x}_p(k|k) &= \mathbf{x}_p(k) \\ \mathbf{x}_p(k+T+1|k) &\in \mathcal{Q}_p \\ \mathbf{y}_p(k+j|k) + \tilde{\mathbf{y}}_p(k+j|k) &\in \mathcal{Y}_p(j) \end{aligned}$$

The output sequence  $\tilde{\mathbf{y}}_p(k \dots k+T|k)$  has two components, as shown in Fig. 1(b): (a) the most recent plans of those subsystems earlier than  $p$  in the planning sequence and (b) predicted plans for subsystems later in the sequence

$$\begin{aligned} \tilde{\mathbf{y}}_p(k+j|k) = \left[ \sum_{q \in \{1 \dots N_p | q < p\}} \mathbf{y}_q^*(k+j|k) \right] + \\ \left[ \sum_{q \in \{1 \dots N_p | q > p\}} \bar{\mathbf{y}}_q(k+j|k) \right] \end{aligned} \quad (8)$$

where  $\mathbf{y}_q^*(k+j|k)$  denotes the outputs from the optimal solution to subproblem  $q$  at time  $k$  and where  $\bar{\mathbf{y}}_q(\cdot)$  denotes an predicted sequence for those later in the planning sequence, constructed from the remainder of the previous plan and a single step of the control law  $\kappa_q$ , defined in (7) to keep the subsystem state in its terminal set

$$\begin{aligned} \forall j \in \{0 \dots T-1\} \\ \bar{\mathbf{y}}_q(k+j|k) = \mathbf{y}_q^*(k+j|k-1) \end{aligned} \quad (9a)$$

$$\begin{aligned} \bar{\mathbf{y}}_q(k+T|k) = \mathbf{C}_q \mathbf{x}_q^*(k+T|k-1) + \\ \mathbf{D}_q \kappa_q(\mathbf{x}_q^*(k+T|k-1)) \end{aligned} \quad (9b)$$

The output constraints here use the intermediate sets from the recursions in (6), whereas the centralized problem only used those for the final subproblem  $N_p$ . These recursions involve constraint tightening from one subproblem to the next and one time step to the next, consistent with the need to retain margin for both future disturbances and the intentions of those subsystems yet to plan. Tightening from one subproblem to the next is necessary because of the inclusion of predicted intentions  $\bar{\mathbf{y}}$  from (9) for those subsystems later in the sequence, as shown in Fig. 1(b). These are nominal predictions only and the actual plans of those subsystems will differ due to the disturbance, hence margin must be retained for those modifications.

The subproblem  $P_p$  is employed in the following algorithm, also outlined in Fig. 1.

#### Algorithm 1 Decentralized MPC

- 1) Find a solution to the initial centralized problem  $P_C(\mathbf{x}_1(0), \dots, \mathbf{x}_{N_p}(0))$ . If solution cannot be found, stop (problem is infeasible).

- 2) Set  $k = 0$
- 3) Apply control  $\mathbf{u}_p^*(k|k)$  to each subsystem  $p$
- 4) Increment  $k$
- 5) For each subsystem  $p$  in order  $1, \dots, N_p$  :
  - a) Gather, by communication, the plan data  $\tilde{\mathbf{y}}_p(k \dots k + T|k)$  from other subsystems, defined by (8) and (9)
  - b) Solve subproblem  $P_p(\mathbf{x}_p(k), \tilde{\mathbf{y}}_p(k \dots k + T|k))$
- 6) Go to 3

Note that Step 1 does not require finding the optimal solution of the centralized problem, although that is one way of finding the initial plan.

**Theorem: Robust Feasibility** If a feasible solution to the initial problem  $P_C(\mathbf{x}_1(0), \dots, \mathbf{x}_{N_p}(0))$ , solved in Step 1 of Algorithm 1, can be found, then following Algorithm 1, all subsequent optimizations will be feasible and the constraints will be satisfied at every time step for all disturbance sequences obeying (3).

Space restrictions preclude a complete proof, but the outline is as follows. It is necessary to consider only one time step and show that feasibility at time  $k$  implies feasibility at time  $k + 1$  for all disturbances  $\mathbf{w}_p(k)$  obeying (3). Then, by recursion, feasibility at time 0 implies feasibility at all future steps. The recursion is broken into multiple stages:

- 1) Assume a set of solutions is known for all subsystems at time  $k$  satisfying the centralized problem  $P_C(k)$  (where the shortened index  $(k)$  denotes the problem for all states at that time);
- 2) Show that given the assumption in Step 1, a feasible solution can be found to the first subproblem  $P_1(k+1)$  for all disturbances  $\mathbf{w}_1(k)$  by showing feasibility of a candidate solution. The candidate is constructed by shifting the previous plan, assumed known in Step 1, by one time step and adding a perturbation sequence using the predetermined controller

$$\delta \mathbf{u}_1(k + j + 1|k + 1) = \mathbf{K}_1(j) \delta \mathbf{x}_1(k + j + 1|k + 1)$$

- 3) Show that, under the assumption in Step 1, given any solution to the subproblem  $P_p(k + 1)$  for  $p \in \{1, \dots, N_p - 1\}$  then the next subproblem  $P_{(p+1)}(k + 1)$  is feasible, again by showing feasibility of a candidate sequence;
- 4) Show that if a feasible solution to the final subproblem  $P_{N_p}(k + 1)$  can be found, then the set of solutions to all subproblems  $P_p(k + 1)$  satisfies the constraints of the centralized problem  $P_C(k + 1)$

So the recursion sequence is: (solutions at  $k$  satisfy  $P_C(k)$ )  $\Rightarrow$  (subproblem  $P_1(k + 1)$  is feasible)  $\Rightarrow$  (subproblem  $P_2(k + 1)$  is feasible)  $\Rightarrow \dots$  (subproblem  $P_{N_p}(k + 1)$  is feasible)  $\Rightarrow$  (solutions at  $k + 1$  satisfy  $P_C(k + 1)$ ). This sequence is analogous to the procedure flow in Fig. 1: the feasibility of each optimization problem follows from the feasibility of the one immediately before it. The order of constraint tightening is also related to the procedure

flow in Fig. 1(a), such that each subproblem is slightly relaxed compared to its immediate predecessor, permitting additional compensation for uncertainty at each step of the algorithm.

**Remark: Communication Requirements** The formulation presented involves a general form of team-wide output constraints, with all constraints, coupling and individual, represented in the same output vector. However, each subsystem is typically only affected by a subset of these constraints, and this structure can be exploited to reduce the communication load. Two circumstances in particular offer significant simplification: (a) if a constraint does not affect a particular subsystem, it can be ignored in that subproblem, and (b) if a constraint applies to only one subsystem, it can be ignored by all others. These cases can be expressed formally in terms of the output matrices. For (a), if all entries in row  $i$  of matrices  $\mathbf{C}_p$  and  $\mathbf{D}_p$  are zero, then the corresponding output and constraints can be omitted from subproblem  $P_p$  and subsystem  $p$  does not need to receive information on that output from other subsystems. For (b), if all entries in row  $i$  of matrices  $\mathbf{C}_q$  and  $\mathbf{D}_q$  for all *other* subsystems  $q \neq p$  are zero, then output  $i$  and the corresponding constraints affect *only* subsystem  $p$ . Then subproblem  $P_p$  must still include output  $i$  and enforce constraints upon it, but no information on that output need be exchanged with other subsystems.

**Remark: Set Approximation** In some cases, the Pontryagin differences in the set recursions (6) can lead to an increase in the total number of constraints in the problem [20]. In these circumstances, it may be desirable to use inner approximations of the constraint sets, leading to a simpler but conservative controller. The robust feasibility depends on the following consequence of the definition of the Pontryagin difference

$$\mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{C} \sim \mathcal{A} \Rightarrow (\mathbf{a} + \mathbf{b}) \in \mathcal{C} \quad (10)$$

Therefore, in place of the sets defined in (6), the designer may substitute any sets obeying the following conditions

$$\begin{aligned} \mathcal{Y}_{N_p}(0) &\subseteq \mathcal{Y} \\ \forall j \in \{0 \dots T\} \forall p \in \{2 \dots N_p\} \\ \mathcal{Y}_{(p-1)}(j) &\subseteq \mathcal{Y}_p(j) \sim (\mathbf{C}_p + \mathbf{D}_p \mathbf{K}_p(j)) \mathbf{L}_p(j) \mathcal{W}_p \\ \forall j \in \{0 \dots T - 1\} \\ \mathcal{Y}_{N_p}(j + 1) &\subseteq \mathcal{Y}_1(j) \sim (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{K}_1(j)) \mathbf{L}_1(j) \mathcal{W}_1 \end{aligned}$$

It is often possible to use techniques such as norm bounding and induced norms to find suitable sets, using the property in (10) to test inclusion within the Pontryagin difference.

## V. EXAMPLES

The first example involves a simple problem in which the positions of two 1-D double integrator systems are to remain within 0.3 m of each other at all times in the presence of a disturbance of up to 10% of the control authority. For comparison, the problem is solved by three

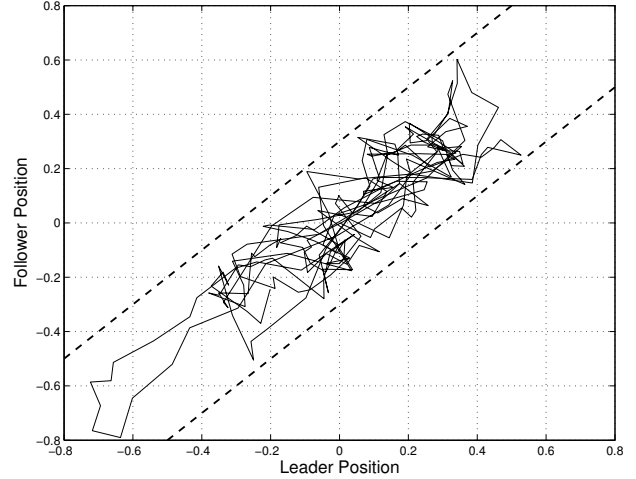
**Table I:** Performance Comparison of Algorithms: total control energy cost averaged over four randomly-disturbed simulations.

Algorithm	Control Cost
Centralized	1.94
Decentralized	2.51
Decoupled	3.71

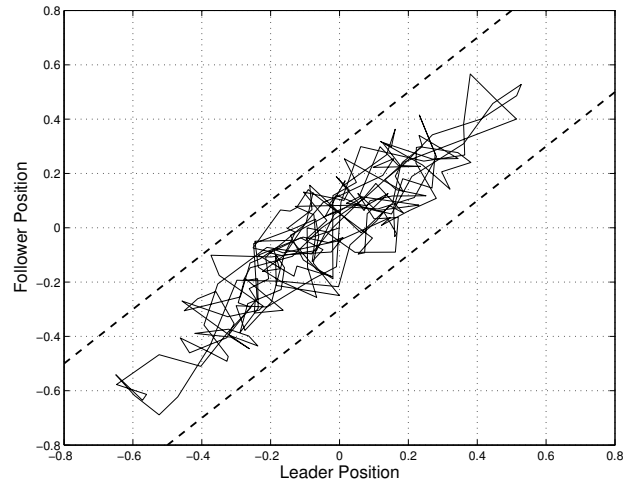
different methods, centralized MPC, decentralized MPC and a decoupled approach, in which each position is individually constrained to remain within 0.15 m of the origin, ensuring satisfaction of the coupled constraints without solving a coupled problem. The cost function is the control energy.

Fig. 2(a) is a graph of the position states of the follower plotted against those of the leader, both under the control of centralized MPC. The coupling constraint requires that the states remain between the dashed lines. The figure shows that the constraints are satisfied at all times. Fig. 2(b) shows the results from the decentralized algorithm, also obeying the constraints at all times and giving a similar result to centralized MPC. Fig. 2(c) shows the results from the decoupled method. The dashed box shows the decoupled constraints used to enforce satisfaction of the coupling constraints. The results satisfy the coupling constraints but it is clear that the problem is far more constrained than either of the other two methods. Note that the results from the decentralized method in Fig. 2(b) do not remain within any possible decoupling constraints. Table I compares the total control energy used by the three algorithms, averaged over four different simulations, each with randomly-chosen disturbances. The centralized algorithm has the lowest cost, as it finds the system-wide optimal solution to the coupled problem. The decoupled method has the highest cost, since it involves artificially tight constraints. As might be expected, the cost of the decentralized method lies in between that of the other two methods.

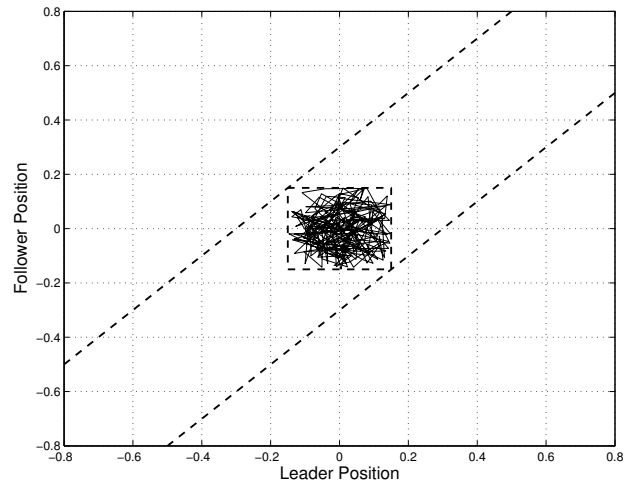
Fig. 3 shows the position time histories under decentralized MPC for a similar problem with six subsystems, each the same double-integrator as before and each pair constrained to remain within 0.3 m. The lower plot in Fig. 3 shows that the maximum separation is always less than 0.3 m, hence the coupling constraints are satisfied. Note that in the upper plot, the positions move beyond the  $\pm 0.15$  m range of the decoupled solution, showing that the decentralized method is again making use of its greater constraint space. Fig. 4 compares averaged solution times using the centralized and decentralized MPC schemes. The decentralized MPC subproblem times are shown stacked as they are solved sequentially. These results show that the decentralized MPC has faster computation and is also more scalable.



(a) Centralized

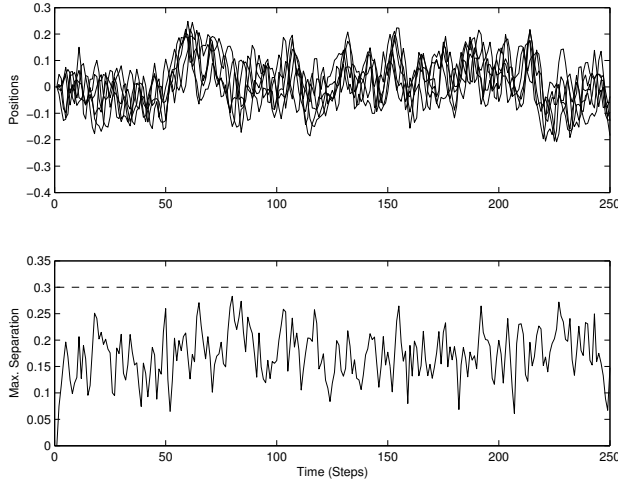


(b) Decentralized

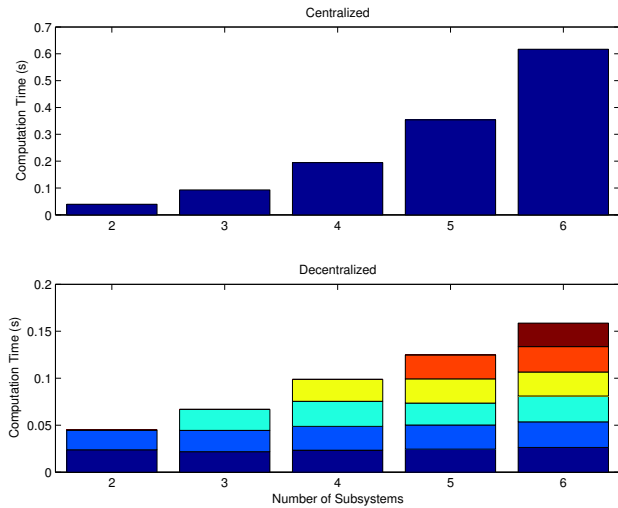


(c) Decoupled

**Fig.2:** Follower Positions vs. Leader Position using Three Forms of MPC. The dashed lines mark the constraints.



**Fig.3:** Position Time Histories (top) and Maximum Separation History (bottom) under DMPC.



**Fig.4:** Solution time comparisons of MPC and DMPC for different numbers of subsystems. Note different vertical scales.

## VI. CONCLUSIONS

A decentralized formulation of model predictive control has been presented. It solves the problem of control of multiple subsystems subjected to coupled constraints but otherwise independent. Each subsystem solves a subproblem involving only its own state predictions. The scheme is proven to guarantee robust constraint satisfaction under the assumption of bounded disturbance action. Each subproblem is guaranteed to be feasible and no iteration between subsystems is required.

## ACKNOWLEDGEMENTS

The authors would like to thank Tamas Keviczky and Prof. Francesco Borrelli at the Department of Aerospace

Engineering and Mechanics, University of Minnesota, for conversations on this subject.

Research funded in part by AFOSR grant # F49620-01-1-0453.

## REFERENCES

- [1] A. Bemporad and M. Morari, "Control of Systems Integrating Logic, Dynamics, and Constraints," *Automatica*, Pergamon / Elsevier Science, New York NY, Vol. 35, 1999, pp. 407–427.
- [2] T. Schouwenaars, B. DeMoor, E. Feron and J. How, "Mixed Integer Programming for Multi-Vehicle Path Planning," *European Control Conference*, Porto, Portugal, September 2001, pp. 2603-2608.
- [3] J.M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, England, 2002.
- [4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Pergamon / Elsevier Science, New York NY, Vol. 36, 2000, pp. 789–814.
- [5] E. C. Kerrigan and J. M. Maciejowski, "Robust Feasibility in Model Predictive Control: Necessary and Sufficient Conditions," *40th IEEE CDC*, Orlando FL, December 2001, pp. 728–733.
- [6] A. G. Richards, T. Schouwenaars, J. How, E. Feron, "Spacecraft Trajectory Planning With Collision and Plume Avoidance Using Mixed-Integer Linear Programming," *AIAA JGCD*, Vol. 25 No. 4, July 2002.
- [7] A. G. Richards, J. S. Bellingham, M. Tillerson and J. P. How, "Coordination and Control of Multiple UAVs", paper no. 2002-4588, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey CA, 2002.
- [8] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra and F. Y. Hadaegh, "A Model Predictive Control-based Approach for Spacecraft Formation Keeping and Attitude Control," *IEEE ACC*, San Diego CA, June 1999.
- [9] D. H. Shim, H. J. Kim and S. Sastry, "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," *42nd IEEE CDC*, Maui, December 2003, p 3621.
- [10] P. O. M. Scokaert and D. Q. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 8, Aug. 1998, p 1136.
- [11] A. G. Richards and J. P. How, "Model Predictive Control of Vehicles Maneuvers with Guaranteed Completion Time and Robust Feasibility," *IEEE ACC*, Denver CO, 2003.
- [12] D. Jia and B. Krogh, "Min-Max Feedback Model Predictive Control For Distributed Control with Communication," *IEEE ACC*, Anchorage AK, 2002, pp.4507-45.
- [13] J. S. Bellingham, A. G. Richards and J. P. How, "Receding Horizon Control of Autonomous Aerial Vehicles," *IEEE ACC*, Anchorage AK, 2002, pp. 3741-3746.
- [14] I Kolmanovsky and E. G. Gilbert, "Maximal Output Admissible Sets for Discrete-Time Systems with Disturbance Inputs," *IEEE ACC*, Seattle WA, 1995, p.1995.
- [15] A. G. Richards, "Trajectory Control Using Mixed Integer Linear Programming," S.M. Thesis, MIT, 2002.
- [16] Camponogara, E., Jia, D., Krogh, B.H. and Talukdar, S., "Distributed model predictive control", *IEEE Control Systems Magazine*, 2002.
- [17] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations" *41st IEEE CDC*, 2002.
- [18] A. N. Venkat and J. B. Rawlings, "Communications Enhanced Large Scale MPC," poster, available at <http://www.che.utexas.edu/twmcc/Presentation0203/venkat/poster03spring.pdf>
- [19] E. Kerrigan, Invariant Set Toolbox for Matlab, available at <http://www-control.eng.cam.ac.uk/eck21>, July 2003.
- [20] E. C. Kerrigan, "Robust Constraint Satisfaction: Invariant Sets and Predictive Control," PhD Thesis, Cambridge University, November 2000.
- [21] S. L. Waslander, G. Inalhan and C. J. Tomlin, "Decentralized Optimization via Nash Bargaining," *Conference on Cooperative Control and Optimization*, 2003.
- [22] T. Keviczky, F. Borrelli and G. J. Balas, "Model Predictive Control for Decoupled Systems: A Study on Decentralized Schemes," *IEEE ACC*, Boston MA, 2004.