Decentralized Model Predictive Control of Cooperating UAVs

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

Abstract

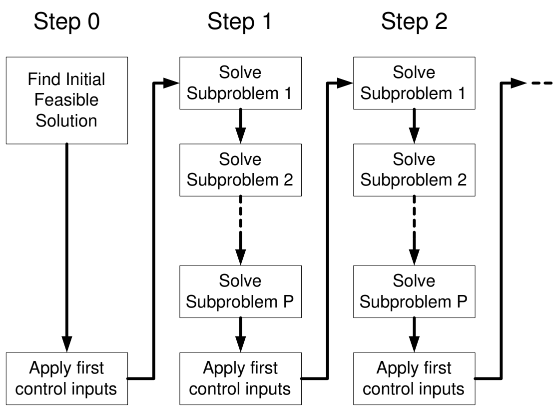
This paper implements robust Decentralized Model Predictive Control (DMPC) for a team of cooperating Uninhabited Aerial Vehicles (UAVs). The problem involves vehicles with independent dynamics but with coupled constraints to capture required cooperative behavior. Using a recently-developed form of DMPC, each vehicle plans only for its own actions, but feasibility of the sub-problems and satisfaction of the coupling constraints are guaranteed throughout, despite the action of unknown but bounded disturbances. UAVs communicate relevant plan data to ensure that decisions are consistent across the team. Collision avoidance is used as an example of coupled constraints and the paper shows how the speed, turn rate and avoidance distance limits in the optimization should be modified in order to guarantee robust constraint satisfaction. Integer programming is used to solve the non-convex problem of path-planning subject to avoidance constraints. Numerical simulations compare computation time and target arrival time under decentralized and centralized control and investigate the impact of decentralization on team performance. The results show that the computation required for DMPC is significantly lower than for its centralized counterpart and scales better with the size of the team, at the expense of only a small increase in UAV flight times.

I. Introduction

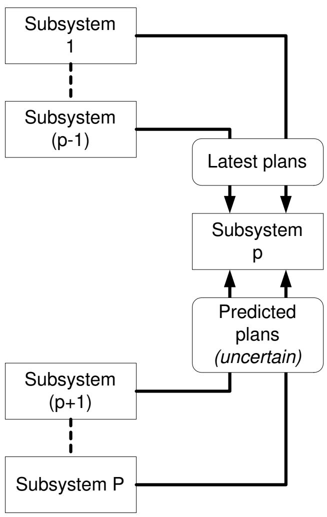
This paper combines Mixed-Integer Linear Programming (MILP) path-planning [2], [4] and Decentralized Model Predictive Control (DMPC) [1] to provide a decentralized algorithm for co-operative guidance of Uninhabited Aerial Vehicles (UAVs). This paper investigates its performance by simulating its behavior for multi-UAV collision avoidance problems. The avoidance constraints couple the behavior of the system and therefore exercise the ability of the controller to make cooperative decisions. The DMPC algorithm [1] was designed primarily to ensure satisfaction of constraints, and critical performance goals can be embedded in constraints. However, secondary goals remain in the cost function, *i.e.* quantities which should be small but have no explicit limit. An important purpose of this paper is to investigate the effect of decentralization on these secondary goals. The metrics of interest in the UAV example are the flight time, which appears in the cost function of the MPC optimizations, and computation time. In particular, we investigate how the algorithm scales with the number of UAVs involved and compare with the behavior of the corresponding centralized MPC.

MPC is a feedback control scheme in which a trajectory optimization is solved on-line 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. MPC has been widely developed for constrained systems [3], with many results concerning stability [5] and robustness [7], [8], and has been applied to the co-operative control of multiple vehicles [8], [10], [6] using centralized computation. 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 [11], with various approaches including robustness to the actions of others [9], penalty functions [12], [15], partial grouping of computations [16], loitering options for safety guarantees [13], and dynamic programming [14]. The challenge of making decisions in a decentralized fashion is to ensure that 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 employs a recently-developed approach to DMPC [1] that addresses both of these difficulties. The key features of this algorithm are that each vehicle only solves a sub-problem for its own plan, and each of these sub-problems is solved only once per time step, without iteration. Under the assumption of a bounded disturbance, each of these sub-problems is guaranteed to be feasible [1], thus ensuring robust constraint satisfaction across the group. The 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



(a) Procedure Flow



(b) Information Flow

Fig.1: Overview of Decentralized Algorithm

data relevant to the coupled constraints is then communicated to the other subsystems. Fig. 1(b) shows the information requirements for sub-problem p. Each sub-problem accommodates (a) the latest plans of those subsystems earlier in the sequence and (b) predicted plans of those later in the sequence. The disturbance is accommodated by including "margin" in the constraints [8], tightening them in a monotonic sequence. At initialization, it is necessary to find a feasible solution to the centralized problem, although this need not be optimal.

The paper begins with the multi-UAV problem statement in Section II. Then Section III lays out the control formulation for this problem. Section IV gives the results of simulations using this controller for many different scenarios and team sizes.

II. UAV Problem Statement

The problem is to control multiple UAVs, each assumed to fly at constant altitude and with restricted speed and rate of turn. Each UAV has a single pre-assigned goal point. The objective is for all UAVs to reach their goals in minimum time without colliding, *i.e.* maintaining a minimum separation between every pair of UAVs at all times. This problem statement exercises the ability of the decentralized method to handle non-convex constraints and coupling between the actions of the UAVs.

Since the decentralized MPC method requires linear dynamics, we adopt the linear approximation of aircraft dynamics from [4]. Each vehicle is modeled as a point mass subject to speed and force limits. A disturbance force is also included in the model to account for uncertainty. Let UAV p have position $\mathbf{r}_p(t) \in \Re^2$ and velocity $\mathbf{v}_p(t) \in \Re^2$ and be acted on by a control force $\mathbf{f}_p(t) \in \Re^2$ and a disturbance force $\tilde{\mathbf{f}}_p(t) \in \Re^2$ giving dynamics

$$\dot{\mathbf{r}}_p(t) = \mathbf{v}_p(t) \qquad \dot{\mathbf{v}}_p(t) = \mathbf{f}_p(t) + \tilde{\mathbf{f}}_p(t)$$

and constraints

$$\|\mathbf{v}_p(t)\|_2 \leq V_p \qquad \|\mathbf{f}_p(t)\|_2 \leq F_p$$

(This model becomes equivalent to Dubins' car model [17] with the addition of a minimum speed constraint. This can be implemented using MILP, but was not done in these experiments for simplicity.) The disturbance force is assumed to be unknown but bounded

$$\|\tilde{\mathbf{f}}_p(t)\|_2 \leq \tilde{F}_p$$

The collision avoidance is expressed in terms of a square exclusion region of side $2L$ around each UAV which no other UAV may enter

$$\|\mathbf{r}_p(t) - \mathbf{r}_q(t)\|_\infty \geq L \ \forall t, p, q : q \neq p$$

Finally, let the goal of UAV p be $\mathbf{g}_p \in \Re^2$.

III. DMPC for UAVs

This section presents the robust MPC optimization problems for the UAV problem described in Section II and the algorithm in which they are employed. Both centralized and decentralized optimizations are presented. The centralized form is relevant because it is used to initialize the decentralized algorithm and also used in comparison tests in Section IV.

The MPC algorithm requires a linear discrete-time state-space dynamics model and linear, although not necessarily convex, constraints. It is straightforward to convert the dynamics to the form

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

with state $\mathbf{x}_p(k) = (\mathbf{r}_p(kT)^T \ \mathbf{v}_p(kT)^T)^T$, control input as $\mathbf{u}_p(k) = \mathbf{f}_p(kT)$, disturbance $\mathbf{w}_p(k) = \tilde{\mathbf{f}}_p(kT)$ and time step T. The speed and force limits are approximated by polyhedra and applied at each time step, along with the collision avoidance constraints

$$[\mathbf{0}\ \mathbf{G}]\mathbf{x}_p(k) \leq \mathbf{1}V_p \qquad (2a)$$
$$\mathbf{G}\mathbf{u}_p(k) \leq \mathbf{1}F_p \qquad (2b)$$
$$\|[\mathbf{I}\ \mathbf{0}]\left(\mathbf{x}_p(k) - \mathbf{x}_q(k)\right)\|_\infty \geq L' \ \forall p, q : q \neq p \qquad (2c)$$

where the rows of $\mathbf{G}$ are unit vectors and $L' > L$ includes some additional margin to account for the the discrete-time application of the constraints. Ref. [4] showed the use of binary logical variables within Mixed-Integer Linear Programming (MILP) to encode non-convex avoidance constraints such as (2c). Finally, the discrete-time disturbance is also converted to polyhedral form.

$$\mathbf{G}\mathbf{w}_p(k) \leq \mathbf{1}\tilde{F}_p \tag{3}$$

Centralized Optimization

$$\mathsf{P}_C\left(\mathbf{x}_1(k), \ldots, \mathbf{x}_P(k)\right) \;:\; J_C^* = \min_{\mathbf{u}_p, \mathbf{x}_p} \sum_{p=1}^{P} T_p$$

subject to $\forall p \in \{1 \ldots P\}$ $\forall j \in \{0 \ldots (T_p - 1)$ $\forall q > p\}$

$$\begin{aligned}
\mathbf{x}_p(k+j+1|k) &= \mathbf{A}\mathbf{x}_p(k+j|k) + \mathbf{B}\mathbf{u}_p(k+j|k) \\
\mathbf{x}_p(k|k) &= \mathbf{x}_p(k) \\
[\mathbf{0}\ \mathbf{G}]\mathbf{x}_p(k) &\leq \mathbf{1}(\bar{V}_p - c_p(j)) \\
\mathbf{G}\mathbf{u}_p(k) &\leq \mathbf{1}(\bar{f}_p - d_p(j))
\end{aligned}$$

$$\begin{aligned}
\|[\mathbf{I}\ \mathbf{0}]\mathbf{x}_p(k+T_p|k) - \mathbf{g}_p\|_\infty &\leq a_p(T_p) \\
\|[\mathbf{I}\ \mathbf{0}]\mathbf{x}_p(k+j|k) - \mathbf{r}_q(k+j|k)\|_\infty &\geq L' + b_{pq}(j)
\end{aligned}$$

where the quantities a, b, c, d are modifications to the constraints to provide margin for robustness

$$\begin{aligned}
a_p(0) &= 0 \\
a_p(1) &= \|[1\ 0\ 0\ 0]\mathbf{B}\|_2\,\tilde{F}_p \\
a_p(j) &= a_p(1) + \|[1\ 0\ 0\ 0]\mathbf{L}\mathbf{B}\|_2\tilde{F}_p \qquad \forall j \geq 2 \\
b_{pq}(j) &= a_p(j) + a_q(j) \qquad \forall j \\
c_p(0) &= 0 \\
c_p(1) &= \|[0\ 0\ 1\ 0]\mathbf{B}\|_2\tilde{F}_p\sqrt{2} \\
c_p(j) &= c_p(1) + \|[0\ 0\ 1\ 0]\mathbf{L}\mathbf{B}\|_2\tilde{F}_p\sqrt{2} \quad \forall j \geq 2 \\
d_p(0) &= 0 \\
d_p(1) &= \|[1\ 0]\mathbf{K}\mathbf{B}\|_2\tilde{F}_p\sqrt{2} \\
d_p(j) &= d_p(1) + \|[1\ 0]\mathbf{K}\mathbf{L}\mathbf{B}\|_2\tilde{F}_p\sqrt{2} \qquad \forall j \geq 2
\end{aligned}$$

where $\mathbf{K}$ is a two step nilpotent controller for the system $(\mathbf{A}, \mathbf{B})$ and $\mathbf{L} = (\mathbf{A} + \mathbf{B}\mathbf{K})$. Hence the nilpotency requirement for $\mathbf{K}$ gives $\mathbf{L} \times \mathbf{L} = \mathbf{0}$. A suitable controller $\mathbf{K}$ can be found since the system is a combination of two independent double-integrators.

Decentralized Optimization (for UAV p)

$$\mathsf{P}_p\left(\mathbf{x}_p(k), \tilde{\mathbf{r}}_{pq}(k \ldots k+T|k)\right) \;:\; J_p^* = \min_{\mathbf{u}_p, \mathbf{x}_p} T_p$$

subject to $\forall j \in \{0 \ldots T_p - 1\}$ $\forall q \neq p$

$$\begin{aligned}
\mathbf{x}_p(k+j+1|k) &= \mathbf{A}_p\mathbf{x}_p(k+j|k) + \mathbf{B}_p\mathbf{u}_p(k+j|k) \\
\mathbf{x}_p(k|k) &= \mathbf{x}_p(k) \\
\mathbf{G}\mathbf{v}_p &\leq \mathbf{1}(\bar{V}_p - c_p(j)) \\
\mathbf{G}\mathbf{f}_p &\leq \mathbf{1}(\bar{f}_p - d_p(j))
\end{aligned}$$

$$\begin{aligned}
\|[\mathbf{I}\ \mathbf{0}]\mathbf{x}_p(k+T_p|k) - \mathbf{g}_p\|_\infty &\leq a_p(T_p) \\
\|[\mathbf{I}\ \mathbf{0}]\mathbf{x}_p(k+j|k) - \tilde{\mathbf{r}}_{pq}(k+j|k)\|_\infty &\geq L' + \hat{b}_{pq}(j)
\end{aligned}$$

where $\tilde{\mathbf{r}}_{pq}(k \ldots k+T|k)$ represents the latest known intentions of other vehicles

$$\tilde{\mathbf{r}}_{pq}(k \ldots k+j|k) = \begin{cases} [\mathbf{I}\ \mathbf{0}]\mathbf{x}_q(k+j|k) & q < p \\ [\mathbf{I}\ \mathbf{0}]\mathbf{x}_q(k+j|k-1) & q > p \end{cases}$$

This represents the latest plans for those UAVs before p in the planning sequence and projected plans, using the continuation of the previous plans, for those after p. This data is exchanged by communication between UAVs and appears as a constrant parameter in sub-problem p. The avoidance margins $\hat{b}$ are modified for the decentralized problem

$$\begin{aligned}
\hat{b}_{pq}(0) &= 0 && q < p \\
\hat{b}_{pq}(0) &= a_q(0) && q > p \\
\hat{b}_{pq}(1) &= a_p(0) + a_q(0) && q < p \\
\hat{b}_{pq}(1) &= a_p(0) + a_q(1) && q > p \\
\hat{b}_{pq}(j) &= a_p(1) + a_q(1) && q \neq p\ \forall j \geq 2
\end{aligned}$$

This accounts for the additional uncertainty in the projected plans of later vehicles, as the actual plans of those UAVs will differ from the predictions due to the action of the disturbance.

Algorithm 1 (Decentralized MPC)

1) Find a solution to the initial centralized problem $\mathsf{P}_C(\mathbf{x}_1(0), \ldots, \mathbf{x}_P(0))$. 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 $1, \ldots, P$:
   a) Compile the plan data $\tilde{\mathbf{r}}_{pq}(k \ldots k+T|k)$ $\forall q \neq p$ from other subsystems
   b) Solve sub-problem $\mathsf{P}_p\left(\mathbf{x}_p(k), \tilde{\mathbf{r}}_{pq}(k \ldots k+T|k)\right)$
6) Go to 3

Theorem 1 (Robust Feasibility) *If a feasible solution to the initial problem $\mathsf{P}_C(\mathbf{x}_1(0), \ldots, \mathbf{x}_P(0))$, solved in Step 1 of Algorithm 1, can be found, then the system (1) controlled by Algorithm 1 and acted upon by disturbances obeying (3) will satisfy the constraints (2) and all sub-problem optimizations will be feasible.*

This theorem is quoted from [1] and is not proven here. Briefly, the result relies on the ability to construct a feasible solution to each sub-problem by combining its previous solution and a perturbation, using the nilpotent control policy $\mathbf{K}$ to counteract the disturbance.

Remarks

The first step of Algorithm 1 requires finding a solution to the centralized problem, but it is not necessary for this to be an optimal solution. Any method of generating a feasible solution can be employed.

Algorithm 1 ignores computation and communication delays. However, it can be extended to explicitly account for delays by propagating the state estimate forward [18].

Crucially, this does not introduce a delay equal to the entire solution sequence. Rather, each subsystem accounts for a delay corresponding to the solution of its own sub-problem.

Algorithm 1 can be run as an "anytime" algorithm. The result in Theorem 1 follows from the ability to construct a feasible solution to each sub-problem before starting its computation. Therefore, an arbitrarily small computation time-limit can be enforced with the guarantee that a solution will always be available.

IV. Simulation Results

This section presents results of numerical simulations demonstrating the application of the DMPC method and investigating its performance, including comparison to the centralized counterpart (CMPC). Fifty random instances of problems for each team size between two and seven were generated and simulated using decentralized MPC. In every case, the UAVs began on the line between $(-5, 5)$ and $(-5, -5)$, heading in the $+X$ direction. The goals were chosen randomly on the line between $(5, 5)$ and $(5, -5)$. Fig. 2 shows a representative example of a problem of each size. The simulation model included a randomly-generated disturbance force of up to 10% of the control force. The simulations were run in Matlab on a desktop PC with Pentium 2.2 GHz processor and 512MB RAM using CPLEX 7.0 for MILP optimization. The same computer was used to solve all UAV sub-problems, consistent with the sequential nature of the algorithm. For comparison, the same fifty instances for team sizes up to five were also simulated, on the same computer, using the CMPC. The same initial solution was used for both algorithms, with CPLEX configured to search primarily for feasible, rather than optimal, solutions and taking the first one found by its search procedure. For the on-line optimizations, CPLEX was configured to seek optimal solutions and subjected to a ten minute computation time-limit, after which the best feasible solution available was employed. The larger problems, with six and seven UAVs, were not attempted using the centralized method as the computation times became prohibitively long.

In all the scenarios, all UAVs reached their respective goals without incurring infeasibility and maintaining the required separations. The examples in Fig. 2 show how the UAVs divert from their direct routes to their goals to avoid collisions. The order of planning, outlined in Fig. 1(a), is marked in the figures. The results indicate that the significance of planning order is unclear and highly problem specific. In the two-, three- and six-vehicle cases, UAV 1 goes straight to its goal, but in the four- and five-vehicle cases, UAV 1 diverts to avoid others. Thus there is no apparent consistent advantage to planning first or last.

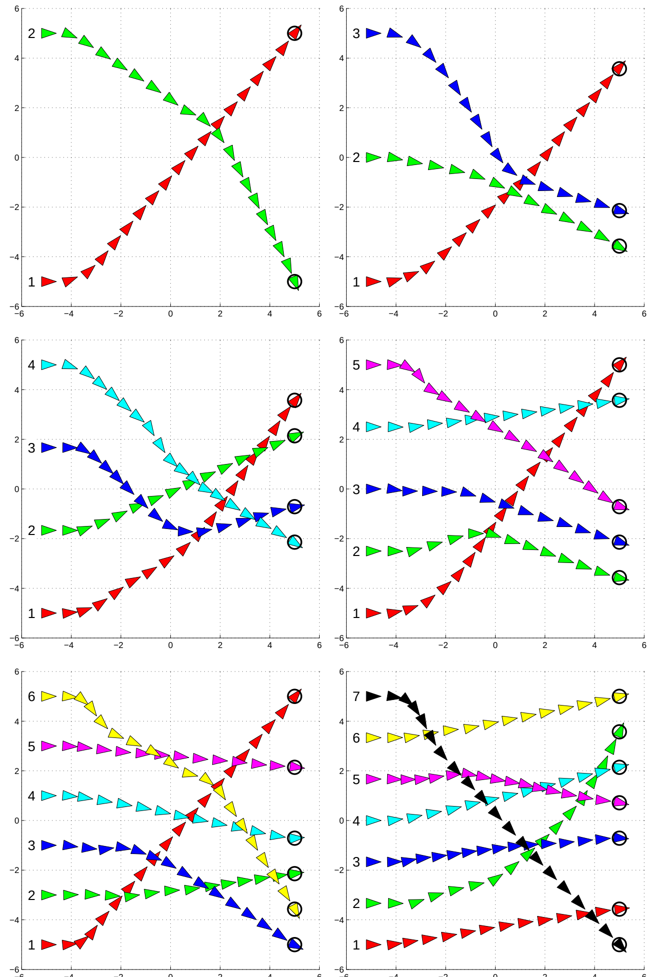


Fig. 2: Example Trajectories for Simulations of Teams of 2-7 UAVs using Decentralized MPC. The numbers denote the planning order and circles mark the goals.

A. Computation Time Comparison

Fig. 3 and Table I present data comparing the total computation times for the centralized and decentralized algorithms. Only computation times for the second time step are considered, as the problem complexity varies as UAVs proceed towards their goals. The results for the decentralized algorithm are the summed times for all the sub-problems, since they are solved sequentially as shown in Fig. 1(a). Looking at the mean times (black dot) and the standard devation envelopes (dark gray bar), it is clear that the decentralized method offers a considerable computational advantage, averaging over twenty times faster than centralized for problems of five UAVs. The ranges (light gray bar) show that there are, however, some rare cases in which the DMPC controller can take around ten minutes to solve. The DMPC computation load still grows nonlinearly with team size, to be expected as the inherent complexity of the problem grows as the UAV "density" increases. However, it is clear that the rate of increase is much slower than for CMPC. Fig. 4 shows the average solution times for the decentralized MPC experiments broken down by sub-problem. This shows that there is no clear pattern to the distribution of computation between sub-problems.

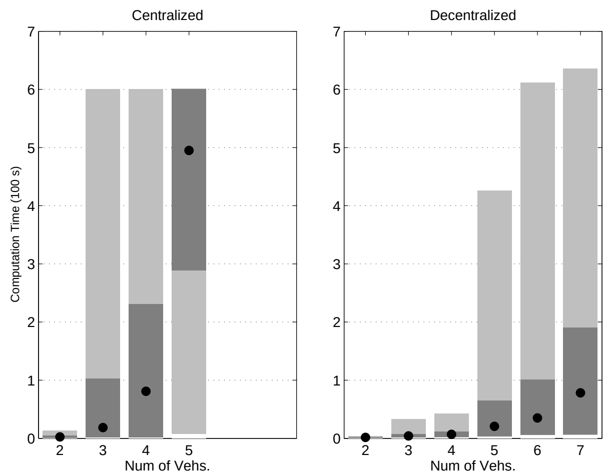


Fig. 3: Comparison of Computation Times for Randomly-Chosen Instances. The plot shows the mean computation times (·), range of times (union of light and dark gray) and standard deviation about mean (dark gray).

TABLE I: Computation Times

Number of Vehicles	Computation Times (s) Min	Mean	Max	St.Dev
	Centralized			
2	0.73	2.51	13.14	2.64
3	1.77	19.45	600.50	87.07
4	1.67	71.91	600.59	134.14
5	8.00	495.12	601.25	206.21
	Decentralized			
2	0.92	1.65	3.55	0.61
3	1.53	4.07	32.49	3.51
4	2.11	6.88	42.86	5.50
5	3.25	20.59	426.14	45.19
6	6.41	34.94	611.70	66.69
7	6.38	78.14	635.61	112.75

B. Flight Time Comparison

The results in Section IV-A showed that the DMPC algorithm offers a considerable improvement over its centralized counterpart in terms of computation time. This section compares the performance of the two algorithms *i.e.* the optimization cost function, in this case the UAV flight time from start to goal. Fig. 5 shows the differences in flight time between DMPC and CMPC for the same fifty random simultions described in Section IV-A. Fig. 5(a) shows the mean and range of the average difference, *i.e.* averaging across all UAVs in the team for each simulation. For up to four UAVs, total flight-times using DMPC are no shorter than for CMPC, but only longer by about one time step per UAV. It is intuitive that DMPC can be no better than CMPC as both solve the same problem, but DMPC in a more constrained manner. However, for the five-vehicle cases, some DMPC results are better than CMPC, occurring when the CMPC optimization is terminated by its computation time-limit with an inferior solution. Fig. 5(b) shows the flight time comparison broken down by individual UAV. The trends in the mean flight time differences suggest a slight advantage to being early in the planning order, but

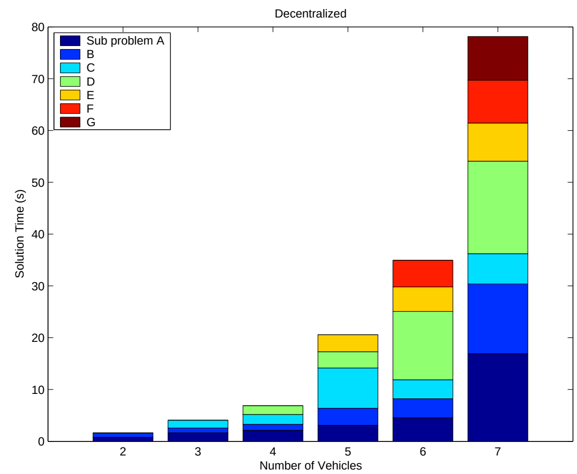


Fig. 4: Mean Solution Times for DMPC with up to Seven UAVs, showing Breakdown among Sub-Problems.

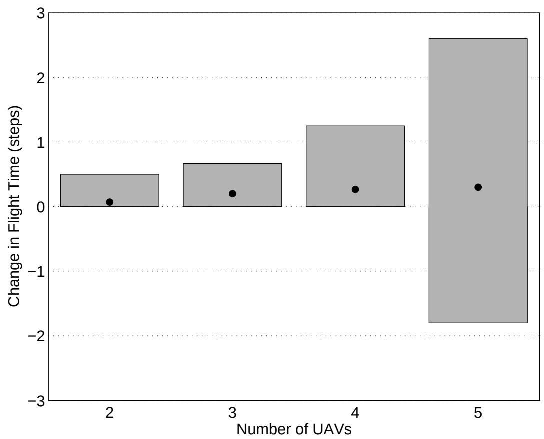
there is considerable variation and no firm conclusion can be drawn.

C. In-Flight Goal Change

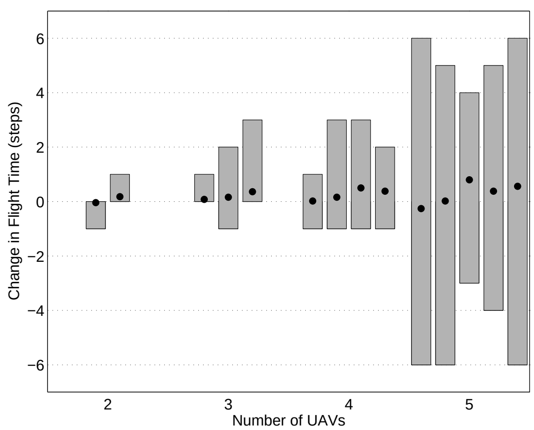
This section shows an example involving a change of the goals part way through the flight. This class of uncertainty is not covered by Theorem 1, but the example shows that DMPC can potentially handle such a change and, crucially, does not require re-initializing using a centralized process. Figs. 6(a) and 6(b) show the trajectories for an example scenario with fixed goals using CMPC and DMPC, respectively. Observe that the trajectories are similar for both controllers and that UAV 3 does not interact with the other two. In the simulations shown in Figs. 6(c) and 6(d), the goals were initially the same as in Figs. 6(a) and 6(b), but at the third time step the goal for UAV 3 was moved to the location shown, requiring UAV 3 to interact with the other two. Comparing Figs. 6(a) and 6(c), it can be seen that all UAVs are diverted to accommodate the changed goal. However, comparing Figs. 6(b) and 6(d), it can be seen that only the path of UAV 3 changes when its goal is changed. This is irrespective of the planning order: when UAV 3 replans for its new goal, it must choose a path consistent with the pre-existing intentions of UAVs 1 and 2. Therefore, UAV 3 takes a more circuitous route to its new goal and there is never any cause to change the paths of UAVs 1 and 2. This example suggests that DMPC can handle broader classes of uncertainty than a simple disturbance action, such as ad-hoc team changes, adding or removing UAVs or goal changes.

V. Conclusions

Decentralized Model Predictive Control (DMPC) for teams of cooperative UAVs has been developed and demonstrated. Simulation results for the representative example of multi-UAV collision avoidance have been presented. DMPC guarantees constraint satisfaction, in this case avoidance,

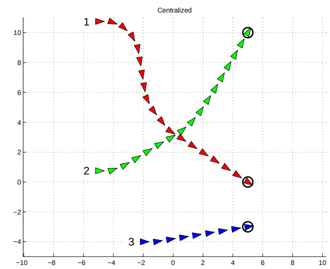


(a) Team Average Flight Time Difference

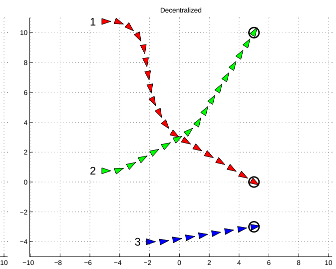


(b) Flight Time Difference for Each UAV

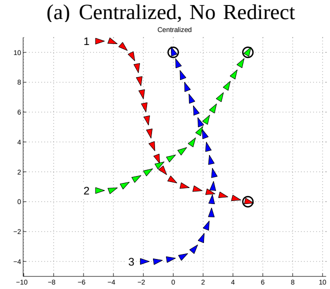
Fig. 5: Differences in UAV Flight Times Between CMPC and DMPC. Positive difference indicates UAV reaches goal earlier using CMPC. Units of time are time steps, and typical flight times are 15 steps per vehicle. The plots show the mean difference (·) and range (gray bar) over fifty instances.



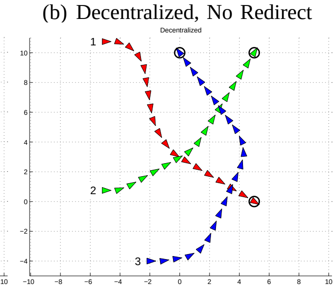
(a) Centralized, No Redirect



(b) Decentralized, No Redirect



(c) Centralized, with Redirect



(d) Decentralized, with Redirect

Fig.6: Diverting a UAV in Mid-Mission

and offers significant computation improvement, compared to the equivalent centralized algorithm, for only a small degradation in performance, in this case UAV flight time.

Acknowledgements

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

References

[1] A. G. Richards, J. How, "A Decentralized Algorithm for Robust Constrained Model Predictive Control," IEEE ACC, Boston MA, June 2004.

[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] A. G. Richards, J. How, "Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming," IEEE ACC, Anchorage AK, May 2002.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, 36(2000), Pergamon Press, UK, pp. 789–814.

[6] 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*, 2002.

[7] P. O. M. Scokaert and D. Q. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Trans. on Automatic Control*, Vol. 43, No. 8, Aug. 1998, p 1136.

[8] 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.

[9] D. Jia and B. Krogh, "Min-Max Feedback Model Predictive Control For Distributed Control with Communication," IEEE ACC, Anchorage AK, 2002, pp.4507-45.

[10] 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.

[11] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations" IEEE CDC, 2002.

[12] D. H. Shim, H. J. Kim and S. Sastry, "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," IEEE CDC, Maui, December 2003, p 3621.

[13] T. Schouwenaars, J. P. How and E. Feron, "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," AIAA Guidance Navigation and Control Conference, paper no. 2004-5141, Providence RI, 2004.

[14] M. Flint, M. Polycarpou and E. Fernandez-Gaucherand, "Cooperative Path-Planning for Autonomous Vehicles using Dynamics Programming," 15th IFAC World Congress, IFAC, Barcelona, Spain, 2002.

[15] S. L. Waslander, G. Inalhan and C. J. Tomlin, "Decentralized Optimization via Nash Bargaining," Conference on Cooperative Control and Optimization, 2003.

[16] T. Keviczky, F. Borrelli and G. J. Balas, "Model Predictive Control for Decoupled Systems: A Study on Decentralized Schemes," IEEE ACC, Boston MA, 2004.

[17] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," American Journal of Mathematics, Vol 79 No 3, pp. 497-516, 1957.

[18] R. Franz, M. Milam and J. Hauser, "Applied Receding Horizon Control of the Caltech Ducted Fan," IEEE ACC, Anchorage AK, p. 3735, 2002.