

Decentralized Cooperative Trajectory Optimization for UAVs with Coupling Constraints

Yoshiaki Kuwata and Jonathan How

Abstract—Motivated by recent research on cooperative UAVs, this paper introduces a new decentralized trajectory optimization approach for systems with independent dynamics but coupled constraints. The primary objective is to improve the performance of the entire fleet by solving local optimization problems and without reproducing the global optimization problem for each agent. To achieve cooperation, the approach exploits the sparse structure of active couplings that is inherent in the trajectory optimization. This enables each local optimization to use a low-order parameterization of the other agents states, thereby facilitating negotiation while keeping the problem size small. The key features of this approach include (a) no central negotiator is required; and (b) it maintains feasibility over the iterations, so the algorithm can be stopped at any time. Furthermore, the local optimizations are shown to always decrease the overall cost. Simulation results are presented to compare the distributed, centralized and other (non-cooperative) decentralized approaches in terms of both computation and performance.

I. INTRODUCTION

Cooperative distributed control has many areas of application, especially in the field of multi-vehicle control such as formation flying spacecraft, unmanned ground vehicles, and Unmanned Aerial Vehicles (UAVs) [1]. To enable fleet-level cooperation, the controller must properly capture the complex interactions between these vehicles, at both the high (tasks) and low (trajectory) levels. One approach is to solve this problem globally, but centralized algorithms typically scale very poorly with the fleet size because of the computational effort involved. This paper presents a new decentralized optimization approach with specific application to trajectory optimization. Much of the current research on decentralized trajectory optimization uses a setup where each vehicle solves a local problem and communicates this intent information to its neighbors [2], [3], [4], [5], [6]. The challenge in this case is how to achieve fleet level cooperation. For example, previous work by Inalhan [2] achieves fleet level agreement by softening the constraints but the required iteration process could take a long time before it even reaches a feasible solution. The approach by Richards [3] ensures the feasibility of the entire fleet under the action of disturbances, but there is no consideration of the overall performance. Similarly, in the “communication-based approach” [7], each vehicle optimizes only for its own control and exchanges information only with neighboring subsystems. These decentralized algorithms typically lead

to a Nash equilibrium [8] or a Pareto optimal surface [9], [2], which is not necessarily the globally optimal/cooperative solution.

In order to achieve cooperative solution, an iterative decentralized scheme has been recently proposed [7] for systems coupled through dynamics. A dual decomposition approach has been proposed for systems coupled through objectives [10]. In this paper, we focus on UAVs with independent dynamics but with coupling constraints. The application examples include formation control of a fleet of UAVs, vehicle avoidance maneuvers, and multi-vehicle path planning under line-of-sight constraints. The proposed algorithm minimizes the global cost by solving local optimizations while satisfying all the constraints. This approach avoids the complexity of global optimization by reducing the order of the other systems while retaining the key couplings. In particular, it exploits the problem structure by parameterizing other vehicles’ decisions using the active coupling constraints. The approach is suitable for trajectory optimization because it tends to have only a few active couplings.

The paper is organized as follows. First, Section II introduces the overall problem and two available approaches. In Subsection III-A, a simple form of the proposed algorithm is presented first to highlight the implication of this approach. Subsection III-B presents the complete decentralized cooperative algorithm. Finally, Section V shows the simulation results and compares the algorithm with other available approaches in terms of performance and computation time.

II. PROBLEM STATEMENT

A. Problem Statement

The problem of interest is the general optimization for multi-vehicle system, with a particular emphasis on the path planning. A fleet of n_v vehicles are assumed to have independent dynamics. Different types of constraints are imposed, but they can be divided into (a) local constraints such as speed bounds, input saturation, and obstacle avoidance; and (b) coupling constraints such as vehicle avoidance, inter-vehicle communication range, and line-of-sight between vehicles.

The system dynamics are in discrete time, and an optimization is performed to obtain the optimal input for each vehicle over N steps into the future:

$$\forall i = 1, \dots, n_v, \quad \forall k = 0, \dots, (N - 1) :$$

$$\mathbf{x}_i(k + 1) = f_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) \quad (1)$$

$$g_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) \leq 0 \quad (2)$$

Y. Kuwata, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, kuwata@mit.edu

J. How, Associate Professor, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, jhow@mit.edu

$$A_i \mathbf{x}_i(k) + A_j \mathbf{x}_j(k) \leq b_{ij}, \quad \forall j = i+1, \dots, n_v \quad (3)$$

where Eq. 1 represents the nonlinear dynamics of vehicle i , Eq. 2 represents all local constraints imposed on the states and the inputs of vehicle i . The pair-wise constraints Eq. 3 capture the couplings between all pairs of vehicles and are assumed to be a combination of linear constraints, which can express various types of constraints including 1-norm, 2-norm, ∞ -norm, and polyhedral constraints.

The objective function for the entire fleet is the sum of individual costs, which could be in conflict with each other.

$$\min_{\mathbf{u}(\cdot)} \sum_{i=1}^{n_v} J_i(\mathbf{x}_i, \mathbf{u}_i) \quad (4)$$

$$\text{s.t. } J_i = \sum_{k=0}^{N-1} l_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) + F_i(\mathbf{x}_i(N)), \quad \forall i \quad (5)$$

where $l_i(\mathbf{x}(\cdot), \mathbf{u}(\cdot))$ is a stage cost and $F_i(\mathbf{x}_i(N))$ is the terminal penalty.

B. Notation

We define the term *neighbor* of vehicle i as a set of vehicles that have any coupling constraint with vehicle i . In particular, if $\mathcal{N}(i)$ denotes this neighbor set for vehicle i , then there exists coupling constraints between i and $j \in \mathcal{N}(i)$. Furthermore, let $\mathcal{A}(i)$ denote a set of vehicles that have active coupling constraints with vehicle i .

For notational simplicity, let z_i denote the decision variable of the i th vehicle, i.e. $z_i \triangleq [\mathbf{u}_i(0)^T, \dots, \mathbf{u}_i(N-1)^T]^T$. Then, with some abuse of notation, a compact form of the optimization Eqs. 1 to 5 can be written as

$$\min_{z_1, \dots, z_{n_v}} \sum_{i=1}^{n_v} J_i(z_i) \quad (6)$$

$$\text{subject to } \forall i: \quad g(z_i) \leq 0 \\ h(z_i, z_j) \leq 0, \quad j \in \mathcal{N}(i)$$

where $g(z_i)$ represents the local constraints for vehicle i and $h(z_i, z_j)$ represents the coupling constraints between vehicles i and j .

C. Centralized Approach

The centralized approach directly solves the full (and potentially large) optimization given in Eq. 6. This approach produces the globally optimal solution; however, it scales poorly because the optimization becomes very complex for large fleets for certain problem types (i.e. quadratic programming and mixed-integer linear programming).

D. Decentralized Non-cooperative Approach

One decentralized approach is to decompose the centralized problem Eq. 6 into smaller subproblems. Figure 1 shows the procedural flow. Similar to Gauss-Seidel iteration [11], the approach sequentially solves the local subproblems and sends the solutions to other vehicles. In a subproblem, each vehicle i freezes the other vehicles' decision variables and

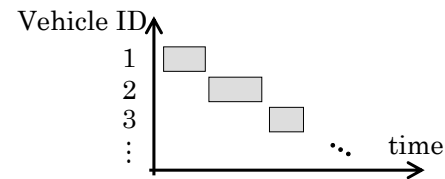


Fig. 1. Decentralized sequential planning. Each gray region represents that the vehicle is computing during that time.

solves for its own optimal input z_i^* . The local optimization for vehicle i can then be written as

$$\begin{aligned} \min_{z_i} J_i(z_i) & \quad (7) \\ \text{subject to } g(z_i) & \leq 0 \\ h(z_i, \bar{z}_j) & \leq 0, \quad j \in \mathcal{N}(i). \end{aligned}$$

The decision variables of the other vehicles are assumed to be constant, which is denoted as \bar{z}_j . The solution z_i^* is sent to the other vehicles, and the optimization of the next vehicle $i+1$ starts after receiving z_i^* .

The main advantage of this approach is that the subproblem has a smaller decision space (approximately $1/n_v$ of the centralized approach), with much fewer constraints. As a result, the computation time is much smaller and the algorithm scales much better than the centralized approach. Furthermore, all the constraints are satisfied while cycling through the vehicles. However, since each vehicle does not account for the objectives of the other vehicles, the resulting solution is coordinated but non-cooperative. This non-cooperative solution is called a Nash equilibrium, where no vehicle can improve its local cost by changing only its own decision, and has to be avoided in the cooperative approach. These benefits and limitations of this approach are clearly illustrated in the examples in Section V.

III. DECENTRALIZED COOPERATIVE OPTIMIZATION

This section presents a new decentralized cooperative algorithm. We first describe the approach for a simple version of the algorithm with only two vehicles i and j , one coupling constraint, and no local constraints.

A. Simple Version

Similar to the decentralized non-cooperative approach, each vehicle solves a local optimization problem in sequence by freezing the other vehicle's decision when the coupling constraints are inactive. The key difference is that when there is an active coupling constraint, the new approach recognizes that each vehicle should consider sacrifices to its local performance if it is possible to reap a larger benefit to the overall team performance.

More formally, when vehicle i solves the optimization after vehicle j , the coupling constraint $a_i z_i + a_j \bar{z}_j \leq b_{ij}$ is modified, if active, in the following way

$$a_i z_i + a_j \bar{z}_j \leq b_{ij} - \beta. \quad (8)$$

The parameter $\beta \geq 0$ tightens the constraint for vehicle i , which could make the local performance worse. However,

vehicle i can account for the potential benefit to the other vehicle j by adding an extra term to the objective function

$$\min_{z_i, \beta} J_i(z_i) - \lambda \beta \quad (9)$$

where λ is a Lagrange multiplier of the coupling constraint that is obtained from the previous solution of the optimization by vehicle j . This Lagrange multiplier represents the amount of improvement the optimization of vehicle j can obtain given some change in the right hand side of the coupling constraint $a_i \bar{z}_i + a_j z_j \leq b_{ij}$. In the hierarchical setup, the Lagrange multiplier could be used to represent the “price” of each vehicle’s solution that the centralized negotiator can use to obtain a coordinated solution [12]. Note that this approach is meaningful only when the coupling constraint is active and hence $\lambda \neq 0$.

The decision variables of vehicle i ’s optimization in Eq. 9 are its local decisions z_i and the negotiation parameter β for the other vehicle. The parameter β allows vehicle i to sacrifice its local cost if that leads to more benefit to the other vehicle j , which is a cooperative behavior.

B. Full Version

This section generalizes the idea in III.A to cases with multiple coupling and local constraints. When there are multiple active coupling constraints, simply including the effect $\lambda\beta$ for each active coupling constraint could doubly count the benefit that the other vehicle can obtain. Also, the scope of the simple problem must be expanded because it is possible to have vehicle i tighten its coupling constraint, as in Eq. 8, to enlarge the operating region of vehicle j , but a local constraint of vehicle j prevents it from using that extra region and obtaining the expected benefit $\lambda\beta$.

The simple version used the negotiation parameter β in vehicle i ’s optimization as an implicit decision variable for vehicle j . In the full version, vehicle i makes an explicit decision for vehicle j , which is denoted by δz_j . Then, vehicle i ’s optimization is

$$\min_{z_i, \delta z_j} J_i(z_i) + J_j(\bar{z}_j + \delta z_j) \quad (10)$$

$$g(z_i) \leq 0, \quad g(\bar{z}_j + \delta z_j) \leq 0 \quad (11)$$

$$A_i z_i + A_j(\bar{z}_j + \delta z_j) \leq b_{ij} \quad (12)$$

where δz_j is related to β through $A_j \delta z_j \Leftrightarrow \beta$. The modified local optimization in Eq. 10 appears similar to the one solved in the centralized approach, but the key point is that, for the problems of interest in this work, the decision space can be reduced significantly, as discussed below.

1) *Sparse active coupling*: The approach avoids reproducing the global optimization problem for each agent by exploiting the structure of active couplings that is typical of trajectory optimization. For example, the vehicle avoidance constraints are imposed over all the time steps of the plan, but they are typically active only at one or two time steps. Communication range limit constraints could be expressed as a nonlinear two-norm constraint on the relative position, but a combination of several linear constraints can approximate it.

In such a case, only a few of the many existing constraints are active. The algorithm exploits this *sparse* structure of the active coupling constraints to reduce the size of the optimization problem.

2) *Low-order parameterization*: Without loss of generality, the upper rows of the coupling constraints Eq. 12 can be regarded as active and the lower rows as inactive.

$$\begin{bmatrix} A_j^{\text{active}} \\ A_j^{\text{inactive}} \end{bmatrix} \delta z_j = \begin{bmatrix} \beta^{\text{active}} \\ * \end{bmatrix}$$

We focus on changing β^{active} by δz_j , because a change in these active coupling constraints can lead to the direct improvement of the other vehicle’s cost. This corresponds to focusing on the non-zero λ in the simple version. In order to address the change in β^{active} , a low-order parameterization of δz_j can be used because $\dim(\beta^{\text{active}}) \ll \dim(\delta z_j)$.

Let m denote the row rank of A_j^{active} , which is also a number of elements in β^{active} that any δz_j can change independently. Therefore, a new variable $\alpha_j \in \mathbb{R}^m$ could replace δz_j , where in the trajectory optimization problems the dimension of α_j is significantly smaller than the dimension of δz_j . Let \check{A} denote a matrix composed of the m independent row vectors extracted from A_j^{active} . Then, δz_j is parameterized by α_j as

$$\delta z_j = \check{A}^T (\check{A} \check{A}^T)^{-1} \alpha_j \triangleq T_j \alpha_j. \quad (13)$$

The inverse in this equation exists because the product $(\check{A} \check{A}^T)$ is a matrix of full rank m , so the parameterization matrix T_j also exists. With this new variable α_j , the local optimization can be rewritten as

$$\min_{z_i, \alpha_j, j \in \mathcal{A}(i)} \left\{ J_i(z_i) + \sum_{j \in \mathcal{A}(i)} J_j(\bar{z}_j + T_j \alpha_j) \right\} \quad (14)$$

$$\text{subject to} \quad g(z_i) \leq 0$$

$$g(\bar{z}_j + T_j \alpha_j) \leq 0, \quad j \in \mathcal{A}(i)$$

$$h(z_i, \bar{z}_j) \leq 0, \quad j \in \mathcal{N}(i), j \notin \mathcal{A}(i)$$

$$h(z_i, \bar{z}_j + T_j \alpha_j) \leq 0, \quad j \in \mathcal{A}(i)$$

$$h(\bar{z}_k, \bar{z}_j + T_j \alpha_j) \leq 0, \quad k \in \mathcal{N}(j), k \notin \mathcal{A}(i)$$

$$h(\bar{z}_{j_1} + T_{j_1} \alpha_{j_1}, \bar{z}_{j_2} + T_{j_2} \alpha_{j_2}) \leq 0, \quad j_1, j_2 \in \mathcal{A}(i).$$

The parameterization is based on the active coupling constraints, but the optimization includes both the active and inactive constraints. The first two constraints are the local constraints for vehicle i and for its active coupling neighbors. The next four equations express different types of couplings shown in Figure 2. Type I is between vehicle i and its neighbors with no active couplings; Type II is between vehicle i and its neighbors with active couplings; Type III is between vehicle i ’s active coupling neighbors and their neighbors; Type IV is between vehicle i ’s two active coupling neighbors. Note that some constraints in $h(z_i, \bar{z}_j + T_j \alpha_j) \leq 0$ could be omitted if α_j has no impact on them because of the row rank deficiency of T_j .

The key points of this algorithm are 1) it does not freeze the other vehicle’s plan, so that it can avoid Nash

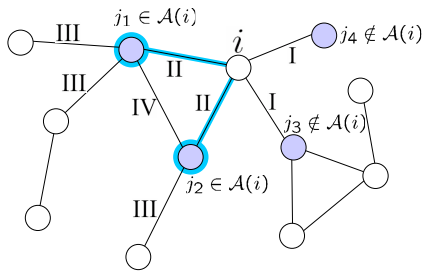


Fig. 2. Different types of coupling constraints. Each node represents a vehicle, and the arc connecting two nodes shows the existence of a coupling constraint between them.

equilibrium; and 2) it reduces the decision space of the other vehicles, so that the complexity of each local optimization remains low. For the problem of interest with a relatively large decision space (e.g., $\dim(z_i) = 20$, $n_v = 5$) and sparse active couplings (2 active coupling neighbors, $\dim(\alpha) = 2$), the reduction of the decision space of each optimization would be a factor of ~ 4 .

The algorithm iterates over the vehicles and the complete flow is summarized below:

- 1) For each vehicle i
 - a) Find the active coupling with this vehicle: $\mathcal{A}(i)$
 - b) Calculate the parameterization matrix T_j , $j \in \mathcal{A}(i)$
 - c) Solve local optimization Eq. 14 and obtain the solution (z_i^*, α_j^*)
 - d) Send the solution to other vehicles. Each vehicle updates the plan

$$z_i := z_i^*$$

$$z_j := z_j + T_j \alpha_j^*, \quad j \in \mathcal{A}(i)$$
- 2) Terminate if the maximum number of iterations is reached. Otherwise, increment the vehicle counter $i := i + 1$ and go to Step 1.

Note that in the step 1c) the vehicle i makes a decision on itself and its neighbors. Then, the step 1d) ensures that all of the vehicles have the same values for the decision variables z_i, z_j . The local optimization Eq. 14 requires the knowledge of other vehicles' cost function, but this depends only on the target score, target location, vehicle states etc., and is simple to communicate. The simulation results in Section V show that 2 iterations over the fleet produces a good performance that is comparable to the centralized approach.

IV. ALGORITHM PROPERTIES

A. Feasibility Over the Iteration

First, we show that feasibility is maintained over the iteration.

Theorem 1. Assume the fleet initially satisfies all of the constraints

$$\forall j : \quad g(z_j) \leq 0$$

$$h(z_j, z_k) \leq 0, \quad k \in \mathcal{N}(j).$$

Then, if the optimization of vehicle i is feasible, the optimization of the next vehicle $i + 1$ is also feasible.

Proof. Let the superscript $(\cdot)^\circ$ represents the decision variable prior to vehicle i 's optimization, and the superscript $(\cdot)^*$ represents the solution optimized by i . Initially,

$$\forall j : \quad g(z_j^\circ) \leq 0$$

$$h(z_j^\circ, z_k^\circ) \leq 0, \quad k \in \mathcal{N}(j).$$

The feasible solution of vehicle i 's optimization obtained in step 1c) satisfies the constraints

$$g(z_i^*) \leq 0$$

$$g(z_j^\circ + \delta z_j^*) \leq 0, \quad j \in \mathcal{A}(i)$$

$$h(z_i^*, z_j^\circ) \leq 0, \quad j \in \mathcal{N}(i), j \notin \mathcal{A}(i)$$

$$h(z_i^*, z_j^\circ + \delta z_j^*) \leq 0, \quad j \in \mathcal{A}(i)$$

$$h(z_k^\circ, z_j^\circ + \delta z_j^*) \leq 0, \quad k \in \mathcal{N}(j), k \notin \mathcal{A}(i)$$

$$h(z_{j_1}^\circ + \delta z_{j_1}^*, z_{j_2}^\circ + \delta z_{j_2}^*) \leq 0, \quad j_1, j_2 \in \mathcal{A}(i).$$

After communicating the solutions and updating them in step 1d), the variables satisfy all the constraints

$$g(\bar{z}_i) \leq 0, \quad \forall i$$

$$h(\bar{z}_i, \bar{z}_j) \leq 0, \quad \forall j \in \mathcal{N}(i) \quad (15)$$

where

$$\bar{z}_i = z_i^*$$

$$\bar{z}_j = z_j^\circ + \delta z_j^*, \quad j \in \mathcal{A}(i)$$

$$\bar{z}_j = z_j^\circ, \quad j \notin \mathcal{A}(i).$$

Given these results, the constraints for the problem for the next vehicle in the optimization sequence $(i + 1)$ are:

$$g(z_{i+1}) \leq 0$$

$$g(\bar{z}_j + \delta z_j) \leq 0, \quad j \in \mathcal{A}(i + 1)$$

$$h(z_{i+1}, \bar{z}_j) \leq 0, \quad j \in \mathcal{N}(i + 1), j \notin \mathcal{A}(i + 1)$$

$$h(z_{i+1}, \bar{z}_j + \delta z_j) \leq 0, \quad j \in \mathcal{A}(i + 1)$$

$$h(\bar{z}_k, \bar{z}_j + \delta z_j) \leq 0, \quad k \in \mathcal{N}(j), k \notin \mathcal{A}(i + 1)$$

$$h(\bar{z}_{j_1} + \delta z_{j_1}, \bar{z}_{j_2} + \delta z_{j_2}) \leq 0, \quad j_1, j_2 \in \mathcal{A}(i + 1). \quad (16)$$

By comparing Eqs. 15 and 16, it can be shown that reusing the solution from vehicle i 's optimization

$$z_{i+1} = \bar{z}_{i+1},$$

$$\delta z_j = 0, \quad \forall j \in \mathcal{A}(i + 1)$$

provides a feasible solution to the optimization of $i + 1$. ■

Note that because the feasibility of the entire fleet is maintained over the iteration, the algorithm could be terminated at any time. Some previous work [3], [5] also maintains this property but not the following property about the performance.

B. Global Cost Monotonically Decreasing

This subsection shows that the global objective value is monotonically decreasing along the iteration.

Theorem 2. The global cost function defined by

$$J(\mathbf{z}) = \sum_{i=1}^{n_v} J_i(z_i)$$

is monotonically decreasing over the iteration.

Proof. The local optimization of vehicle i results in

$$J_i(z_i^\circ) + \sum_{j \in \mathcal{A}(i)} J_j(z_j^\circ) \geq J_i(z_i^*) + \sum_{j \in \mathcal{A}(i)} J_j(z_j^\circ + \delta z_j^*)$$

Let $J(z^\circ)$ denote the global cost prior to the optimization by vehicle i and $J(\bar{z})$ denote the global cost prior to the optimization by the next vehicle $i + 1$. Then,

$$\begin{aligned} J(z^\circ) &= J_i(z_i^\circ) + \sum_{j \in \mathcal{A}(i)} J_j(z_j^\circ) + \sum_{j \notin \mathcal{A}(i)} J_j(z_j^\circ) \\ &\geq J_i(z_i^*) + \sum_{j \in \mathcal{A}(i)} J_j(z_j^\circ + \delta z_j^*) + \sum_{j \notin \mathcal{A}(i)} J_j(z_j^\circ) \\ &= J_i(\bar{z}_i) + \sum_{j \in \mathcal{A}(i)} J_j(\bar{z}_j) + \sum_{j \notin \mathcal{A}(i)} J_j(\bar{z}_j) = J(\bar{z}) \end{aligned}$$

Therefore, each local optimization decreases the global cost. Because this algorithm maintains feasibility, it monotonically decreases the global cost over the iteration. ■

V. SIMULATION RESULTS

A. Simulation Setup

In this simulation, all n_v vehicles are assumed to have the same linear dynamics which are described by a simple double integrator model: $\forall i = 1, \dots, n_v$

$$\begin{aligned} \mathbf{x}_i(k+1) &= \begin{bmatrix} O & I \\ O & O \end{bmatrix} \mathbf{x}_i(k) + \begin{bmatrix} 0.5I \\ I \end{bmatrix} \mathbf{u}_i(k), \\ \mathbf{x}_i(k) &= [\mathbf{r}_i(k)^T \quad \mathbf{v}_i(k)^T]^T. \end{aligned}$$

Constraints are imposed on the position, the speed, and the control input of each vehicle at each time step $k = 0, \dots, N$

$$\|\mathbf{r}_i(k)\|_2 \leq 1, \quad \|\mathbf{v}_i(k)\|_2 \leq 0.35, \quad \|\mathbf{u}_i(k)\|_2 \leq 0.18.$$

The systems are coupled with two neighbors through the following position constraints.

$$\begin{aligned} \|\mathbf{r}_i(k) - \mathbf{r}_{i+1}(k)\|_2 &\leq 0.8, \quad i = 1, \dots, (n_v - 1) \\ \|\mathbf{r}_{n_v}(k) - \mathbf{r}_1(k)\|_2 &\leq 0.8 \end{aligned}$$

These two-norm constraints are expressed as a combination of linear constraints. The cost direction for the i th vehicle is

$$c_i = \left[\cos\left(\frac{i-1}{n_v}\right) \quad \sin\left(\frac{i-1}{n_v}\right) \right]^T.$$

The overall cost function to minimize is

$$\begin{aligned} \sum_{i=1}^{n_v} \sum_{k=0}^{N-1} \left\{ \mathbf{x}_i(k)^T R_1 \mathbf{x}_i(k) + \mathbf{u}_i(k)^T R_2 \mathbf{u}_i(k) \right\} \\ + c_i^T \mathbf{r}_i(N) + \mathbf{r}_i(N)^T H \mathbf{r}_i(N) \end{aligned}$$

where the weights on the states R_1 and inputs R_2 in the stage cost are chosen to be much smaller than the weight H on the terminal position. Both the centralized and the local optimization are quadratic programming, and CPLEX 9.1 is used as a solver.

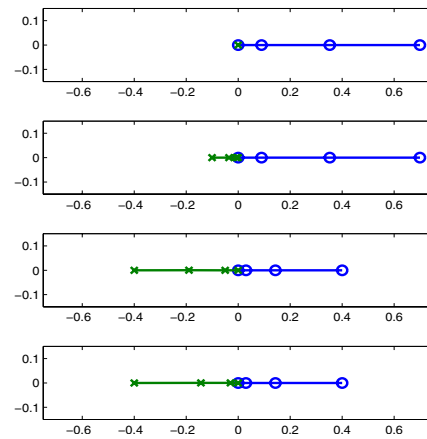


Fig. 3. The evolution of plans over the iteration for the simple two vehicle example.

B. Simple Two Vehicle Case

The first example involves two vehicles i and j that can move on a two dimension plane. The terminal position of the vehicle i has its local minimum at coordinates $(0, 0.7)$

$$\begin{bmatrix} 0 \\ 0.7 \end{bmatrix} = \arg \min_{\mathbf{r}_i(N)} \left\{ c_i^T \mathbf{r}_i(N) + \mathbf{r}_i(N)^T H \mathbf{r}_i(N) \right\},$$

and that of the vehicle j is at $(0, -0.7)$. Because the two vehicles must satisfy the separation constraint of 0.8, their separate objectives are conflicting. The planning horizon is three steps for both vehicles.

Figure 3 shows the evolution of plans over two iterations. The plans of vehicle i are marked with \circ , and those of vehicle j are marked with \times . Originally, both vehicles are at the origin. First, vehicle i solves its local optimization. Because no coupling constraints are active at this point, the plan reaches the local minimum $(0, 0.7)$. Vehicle j then solves its optimization, but given the separation constraint, this vehicle can only plan to move to $(0, -0.1)$, as shown in the second part of the figure.

The vehicle i solves the next optimization, but since a coupling constraint has become active, it uses a parameterized decision for j with a variable α_j of dimension $m = 1$. The bottom figure shows the plans after two iterations. The final plans are the same as the globally optimal centralized solution.

If the decentralized non-cooperative algorithm in Eq. 7 were used, it would produce a Pareto optimal solution shown in the second figure of Figure 3, which is clearly not the globally optimal solution. Note that if the vehicle j plans first followed by vehicle i , the non-cooperative algorithm results in a symmetric Pareto optimal solution, which again is not the globally optimal solution. This example clearly shows the performance improvement over the decentralized non-cooperative approach.

C. Five Vehicle Case

Figure 4 shows a more complex case with five vehicles. In this example, the local minimum for each vehicle is located

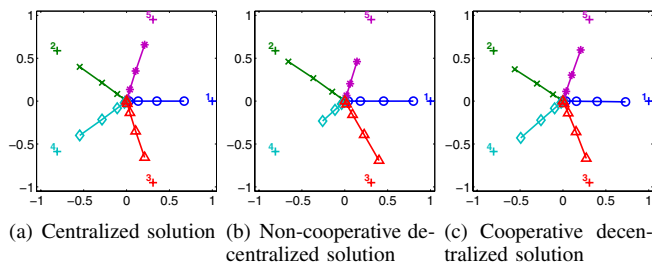


Fig. 4. Final plans for five vehicles

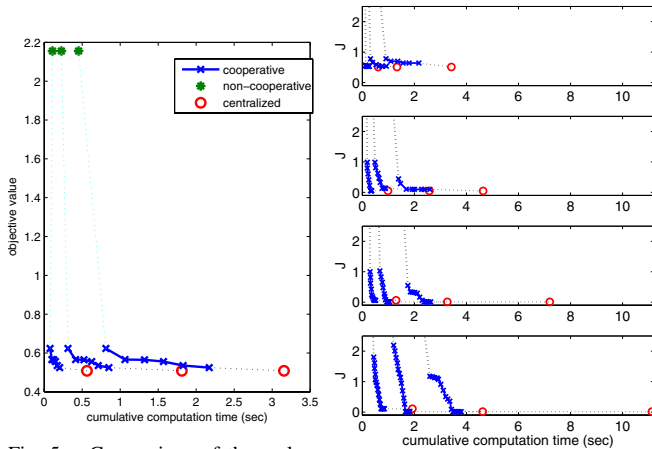


Fig. 5. Comparison of three algorithms in terms of performance and computation.

Fig. 6. Trade off between the performance and the computation time. From the top to the bottom, the number of vehicles are 5, 7, 10, and 15.

on a unit circle centered on the origin. The planning horizon is three steps for all vehicles, and the planning order was $1 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 4$ to highlight the effect of the planning order on the performance.

Two other algorithms described in Section II are used as benchmarks. These are 1) the centralized approach Eq. 6 that provides the globally optimal solution, and 2) the decentralized non-cooperative approach Eq. 7 that produces a locally optimized solution.

As shown in Figure 4(b), the decentralized non-cooperative approach produced a suboptimal solution, because the vehicles that plan earlier are less constrained and have more advantages. The decentralized cooperation algorithm produced the plans in shown Figure 4(c) that is similar to the centralized solution shown in Figure 4(a).

D. Performance and Computation

Figure 5 compares the global objective value and the cumulative computation time of three algorithms for the five vehicle example. Different lengths of planning horizon $N = 4, 6, 8$ were considered to investigate the scalability of the algorithms.

The solutions of the decentralized non-cooperative approach are marked with $*$. Although the computation time is small, the cost is fairly high. The centralized solutions are marked with \circ . The lines with \times show the evolution of the global cost of the decentralized cooperation algorithm. The plot starts from the end of the first iteration when

every vehicle has its solution and continues to the end of the second iteration. This proposed algorithm has objective values comparable to those of the centralized solution but scales better when the problem size increases.

Figure 6 shows cases with more vehicles ($n_v = 5, 7, 10, 15$). The decentralized non-cooperative approach has much higher cost and is out of the range of the plot. For the centralized and the proposed approach, the differences in the computation time scale up significantly for larger fleets. Note that in all the plots of Figures 5 to 6, the lines of the proposed approach are monotonically decreasing, which validates the result in Subsection IV-B by simulation.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a decentralized cooperation algorithm for systems coupled through the constraints. By exploiting the sparse structure of the active coupling constraints of trajectory optimization, the algorithm uses low-order parameterization of the neighbors' decisions. Simulation results showed that the proposed algorithm scales much better than the centralized approach and the performance is much better than that of the non-cooperative approach. Over the iteration, it is shown to maintain feasibility and monotonically decrease the global cost.

ACKNOWLEDGMENTS

Research funded by AFOSR Grant # FA9550-04-1-0458.

REFERENCES

- [1] R. Saber, W. Dunbar, and R. Murray, "Cooperative control of multi-vehicle systems using cost graphs and optimization," in *Proceedings of the IEEE American Control Conference*, 2003.
- [2] G. Inalhan, D. M. Stipanovic, and C. J. Tomlin, "Decentralized Optimization, with Application to Multiple Aircraft Coordination," in *Proceedings of the IEEE Conference on Decision and Control*, December 2002.
- [3] A. Richards and J. How, "Decentralized Algorithm for Robust Constrained Model Predictive Control," in *Proceedings of the IEEE American Control Conference*. Boston, MA: IEEE, 2004.
- [4] W. Dunbar and R. Murray, "Receding horizon control of multi-vehicle formation: A distributed implementation," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.
- [5] T. Schouwenaars, J. How, and E. Feron, "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2004.
- [6] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance," in *Accepted to appear in the proceedings of American Control Conference*, 2006.
- [7] A. Venkat, J. Rawlings, and S. Wright, "Stability and optimality of distributed model predictive control," in *IEEE Conference on Decision and Control, and the European Control Conference*, 2005.
- [8] U. Ozguner and W. R. Perkins, "Optimal control of multilevel large-scale systems," *International Journal of Control*, vol. 28, no. 6, pp. 967–980, 1978.
- [9] P. Heiskanen, "Decentralized method for computing Pareto solutions in multiparty negotiations," *European journal of Operational Research*, vol. 117, pp. 578–590, 1999.
- [10] R. Raffard, C. Tomlin, and S. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.
- [11] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [12] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, and A. Wozniak, *Control and coordination in hierarchical systems*. London: A Wiley – Interscience Publication, 1980, vol. 9.