

## Research Article

# Decentralized Model Predictive Control for Cooperative Multiple Vehicles Subject to Communication Loss

**Hojjat A. Izadi, Brandon W. Gordon, and Youmin Zhang**

*Department of Mechanical and Industrial Engineering, Concordia University, 1515 St. Catherine W, Montreal, QC, Canada H3G 2W1*

Correspondence should be addressed to Youmin Zhang, ymzhang@encs.concordia.ca

Received 1 July 2010; Revised 24 November 2010; Accepted 28 February 2011

Academic Editor: Yu Gu

Copyright © 2011 Hojjat A. Izadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The decentralized model predictive control (DMPC) of multiple cooperative vehicles with the possibility of communication loss/delay is investigated. The neighboring vehicles exchange their predicted trajectories at every sample time to maintain the cooperation objectives. In the event of a communication loss (packet dropout), the most recent available information, which is potentially delayed, is used. Then the communication loss problem changes to a cooperative problem when random large communication delays are present. Such large communication delays can lead to poor cooperation performance and unsafe behaviors such as collisions. A new DMPC approach is developed to improve the cooperation performance and achieve safety in the presence of the large communication delays. The proposed DMPC architecture estimates the tail of neighbor's trajectory which is not available due to the large communication delays for improving the performance. The concept of the tube MPC is also employed to provide the safety of the fleet against collisions, in the presence of large intervehicle communication delays. In this approach, a tube shaped trajectory set is assumed around the trajectory of the neighboring vehicles whose trajectory is delayed/lost. The radius of tube is a function of the communication delay and vehicle's maneuverability (in the absence of model uncertainty). The simulation of formation problem of multiple vehicles is employed to illustrate the effectiveness of the proposed approach.

## 1. Introduction

To address the intervehicle communication loss/delay in a network of cooperative multiple vehicles, a previously developed delay-dependent decentralized model predictive control (DMPC) architecture in [1] is modified and extended to subsystems with generic class of dynamics, subject to intervehicle communication packet loss. The proposed decentralized model predictive control (DMPC) features two main contributions. The first contribution is the development of a new DMPC approach that estimates the trajectory of the neighboring vehicles for the tail of the prediction horizon which would otherwise not be available due to the communication losses leading to random communication delays. In this approach, the tail of the cost function is estimated by adding extra decision variables in the cost function.

A relatively small amount of existing works have investigated the implementation issues such as communication loss associated with the exchange of trajectory information, but

so far a few works have proposed a systematic tail estimation process to compensate for large delays. For example, in [2–5], it is assumed that the neighboring vehicles remain stationary at the last delayed states broadcasted by them. Such assumptions may yield poor performance for large communication delays, because the stationary state vector is not necessarily an accurate estimate of a time-varying state vector. Similar issues are investigated in [6], where for the hardware implementation of a robust DMPC to wheeled robots, both computation and communication delays are considered. To account explicitly for the time delays a model of the vehicle is used to estimate their state vector when required. Then the uncertainty arising from this estimation is taken into account by accommodating that into the effective disturbance force used for constraint tightening. Also, in [7], the stability, feasibility, and computation issues of a distributed MPC are investigated. The main practical attraction of the proposed architecture in [7] is that it allows terminating the computations at the end of each sampling

period even if convergence is not achieved; this feature facilitates the real-time implementation of DMPC.

The second contribution of this paper is an extension of the tube-based model predictive control (MPC) approach [8, 9] for the case of the large communication delays in order to guarantee the safety of the fleet against possible collisions during formation control problems. The concept of the tube MPC in the existing works [8, 9] is normally used to calculate a robust bound on the states due to model uncertainty, whereas in this paper the approach is used to calculate bounds on the exchanged neighbor trajectories when large intervehicle communication delays exist.

In [10], the wireless communication packet loss/delay is considered; once the packet loss/delay occurs, the previous available trajectory of the faulty vehicle—whose trajectory is delayed—is extrapolated to predict the future reference trajectory. Also, in [11], the communication failure in formation flight of multiple vehicles leads to break in the communicated messages that force the fleet to redefine the communication graph.

This paper is organized as follows. Section 2 presents a general formulation of the delay-free decentralized model predictive controller and the corresponding algorithm. In Section 3 a new delay-compensated DMPC (DC-DMPC) is developed to address the communication delays arising from communication losses. A safety guarantee method is also developed based on the concept of tube MPC. In Section 4 the proposed algorithms are tested through simulation of a leader-follower formation problem for a fleet of unmanned vehicles.

## 2. Decentralized Model Predictive Control

Consider a team of cooperative vehicles with uncoupled dynamics. Each vehicle is equipped with three main components: (1) measurement sensors, (2) communication device, and (3) computation resource. The measurement sensors of each vehicle measure its own states assuming full state measurement is available. The communication device is used to gather the information from the neighboring vehicles and communicate with human operators. Using the computation resource, each vehicle solves a decentralized optimization problem at each sampling time based on its current measured states (from sensors) and the neighbor's predicted state trajectories (provided through communication). It is further assumed that there is no sensor error, model uncertainty, or communication noise. These assumptions allow focusing on the main problem concerning intervehicle communication loss/delay. The developed algorithms do not rely on such assumptions; in fact, these assumptions are not restrictive for extending the proposed algorithms to nonideal situations. It is worth mentioning that even if a perfect model of neighboring vehicle is available still there is a possibility of mismatch between the predicted neighboring vehicle's trajectory and the actual ones, because the neighboring vehicles may use a particular communication topology which leads to different DRHC optimization problems and hence different solutions. This problem can be overcome only by a centralized approach which is beyond the approach of this

paper, because in a centralized approach only one communication topology which connects all the team members exists. Then, it is intended to focus on such mismatches arising from decentralized nature of the problem rather than mismatches coming from model uncertainty, and so forth.

Moreover, the terms *agent*, *vehicle*, *team*, *member*, and *subsystem* bear the same meaning. And so do the terms *group*, *team*, *fleet*, and *network*.

**2.1. Interaction Graph Topologies.** *Graph theory* is a relevant mathematical tool for modeling and analyzing the interactions among the cooperative vehicles (see for instance, [12, 13]). *Graph topology* is described by two basic elements: *nodes* and *edges*. *Nodes* represent the vehicles, and an *edge* between two nodes denotes the interaction between those vehicles. The *interaction graph* is denoted by  $\mathbb{G}$  and represented as follows:

$$\mathbb{G} = \{\mathbb{V}, \mathbb{E}\}, \quad (1)$$

where  $\mathbb{V}$  is the set of nodes (vehicles) and  $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$  is the set of edges  $(i, j)$ , with  $i, j \in \mathbb{V}$ . Hence, considering a set of  $N_v$  vehicles cooperating to perform a common mission, the  $i$ th vehicle in the team is assigned to the  $i$ th node of the graph. If an edge  $(i, j)$  connecting the  $i$ th and  $j$ th node is present in  $\mathbb{E}$ , it means that the  $i$ th and  $j$ th vehicles have an interaction. This relation is termed as neighborhood for  $i$ th and  $j$ th vehicles, and it is said that

- (i)  $i$ th and  $j$ th vehicles are *neighboring* vehicles;
- (ii)  $i$ th vehicle is a *follower* of the  $j$ th vehicle;
- (iii)  $j$ th vehicle is a *leader* of the  $i$ th vehicle.

Also, it is assumed that the interaction graph is *directed*, that is,  $(i, j) \in \mathbb{E}$  does not necessarily imply  $(j, i) \in \mathbb{E}$ , unless it appears explicitly in  $\mathbb{E}$ . Further, let  $N_l^i$  and  $N_f^i$  denote the number of the *leaders* and *followers* of vehicle  $i$ , respectively.

In this paper, the information exchange structure coincides the control structure; in fact, an edge  $(i, j)$  indicates that (1) vehicle  $j$  sends information on its planned trajectory to  $i$  and (2) the cost function of vehicle  $j$  is coupled to states of vehicle  $i$ . Also, it is assumed that the *interaction graphs* have a particular fixed structure and is set by the human operator prior to the mission.

**2.2. DMPC Notation and Terminology.** With model predictive control, a cost function is optimized over a finite time window called the *prediction horizon*,  $N$ . The first of the computed optimal inputs is applied to the plant. The reader is referred to [14] for a comprehensive review of MPC schemes.

In this paper, the DMPC architecture is formulated based on the quasi-infinite-RHC formulation [15] where the key is to tune the *terminal cost* so that the closed-loop MPC has the property of an infinite horizon controller. In this approach the states are driven to a neighborhood of the origin where it is a positively invariant set under a feedback *terminal controller* for linearized system.

The discrete timing is then shown by  $k$  where  $k \in \mathbb{N}$ . As mentioned for the DMPC scheme presented in this paper the neighboring vehicles exchange their predicted state trajectories to provide cooperation. However, as we will see later for the time periods when the updated leader's state trajectory is not available, it is estimated by the followers. Hence, considering the trajectory of vehicles, those calculated by themselves and those estimated by followers, an appropriate notation is required to present all these possible trajectories (actual, predicted, and estimated). The possible state vectors are introduced as follows:

- (i)  $\mathbf{x}^i(k)$ : the *actual* state vector of the  $i$ th vehicle at time step  $k$ ;
- (ii)  $\mathbf{x}_k^i(p)$ : the state vector of the  $i$ th vehicle at time  $p$ , *predicted* by the  $i$ th vehicle at time step  $k$ , where  $p = 0, 1, 2, \dots, N$ ;
- (iii)  $\mathbf{x}_k^{i,j}(p)$ : the state vector of the  $i$ th vehicle at time step  $k$ , *estimated* by the  $j$ th vehicle at time step  $k$  where  $p = 0, 1, 2, \dots, N$ .

Further, the sequence of states over the prediction horizon is called the state trajectory of vehicle  $i$  calculated by itself and is represented by  $\mathbf{x}_k^i(\cdot)$ :

$$\mathbf{x}_k^i(\cdot) = \{\mathbf{x}_k^i(p) \mid p = 0, 1, 2, \dots, N\}, \quad (2)$$

$$\mathbf{u}_k^i(\cdot) = \{\mathbf{u}_k^i(p) \mid p = 0, 1, 2, \dots, N-1\}.$$

Then let the following represent the concatenated state and input trajectories of the leader(s) of the  $i$ th vehicle at time  $k$ :

$$\mathbf{x}_k^{-i}(\cdot) = \{\mathbf{x}_k^j(\cdot) \mid j \in \mathbb{V}, (i, j) \in \mathbb{E}\}, \quad (3)$$

$$\mathbf{u}_k^{-i}(\cdot) = \{\mathbf{u}_k^j(\cdot) \mid j \in \mathbb{V}, (i, j) \in \mathbb{E}\}.$$

If the trajectory is defined on an interval which is different from  $[k, k+N]$  by other vehicles, then the beginning and end times are indicated as  $\mathbf{x}_k^{i,j}(b : e)$ , that is,

$$\mathbf{x}_k^{i,j}(b : e) = \{\mathbf{x}_k^{i,j}(p) \mid p = b, b+1, \dots, e, j \in \mathbb{V}, (i, j) \in \mathbb{E}\}, \quad (4)$$

where  $[b : e]$  is the interval on which the trajectory is defined.

**2.3. Delay-Free DMPC Formulation.** In this section, a brief overview of a delay-free DMPC problem and its implementation are described. More details can be obtained from [6, 12, 16]. For the DMPC scheme presented in this paper, the predicted trajectories are exchanged instead of being estimated, thereby reducing the online computational time. Figure 1 shows the delay-free intervehicle communication between two neighboring vehicles and the information exchange at time step  $k$ .

In Figure 1, it is assumed that the current leader's predicted trajectories are always available instantly to their followers. However, at least one step delay has to be considered as the current predicted trajectories are not available

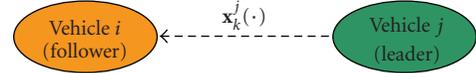


FIGURE 1: The delay-free intervehicle communication between two neighbors.

instantly, even if an infinite communication bandwidth is used. Then, the *information set* of the  $i$ th vehicle for the case of communication delay-free DMPC is introduced as follows:

$$\Gamma^i(k) = \{\mathbf{x}^i(k), \mathbf{x}_{k-1}^{-i}(\cdot)\}, \quad (5)$$

where set  $\Gamma^i(k)$  contains the updated information available to the  $i$ th vehicle at time step  $k$  and is referred to the *information set* in this paper. This collects (1) the instant state vector of the  $i$ th vehicle and (2) the concatenated state trajectory of neighbors calculated at the previous time step,  $\mathbf{x}_{k-1}^{-i}(\cdot)$ .

For the particular case of formation control, the delay-free decentralized cost function for the  $i$ th vehicle in the team at time  $k$  is defined as follows:

$$\begin{aligned} J^i(\Gamma^i(k), \mathbf{u}_k^i(\cdot)) &= \sum_{p=0}^{N-1} \left( \|\mathbf{x}_k^i(p) - \mathbf{x}^{T,i}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_k^i(p)\|_{\mathbf{R}}^2 \right) + \|\mathbf{x}_k^i(N) - \mathbf{x}^{T,i}\|_{\mathbf{P}}^2 \\ &+ \sum_{j|(i,j) \in \mathbb{E}} \left( \sum_{p=0}^{N-1} \|\mathbf{x}_k^i(p) - \mathbf{x}_{k-1}^j(p) - \mathbf{r}^{i,j}\|_{\mathbf{S}}^2 \right), \end{aligned} \quad (6)$$

where  $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}'\mathbf{Q}\mathbf{x}$  and  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$ , and  $\mathbf{S}$  are positive definite and symmetric matrices,  $\mathbf{x}^{T,i}$  is the state vector of target (waypoint) of vehicle  $i$ , and  $\mathbf{r}^{i,j}$  is the vector of desired relative position between vehicles  $i$  and  $j$ .

**2.3.1. Delay-Free DMPC Problem.** Assume that the following describes the discrete-time dynamics of the (homogeneous) vehicle  $i$ :

$$\mathbf{x}^i(p+1) = f(\mathbf{x}^i(p), \mathbf{u}^i(p)), \quad \mathbf{x}^i(0) = \mathbf{x}^i(0). \quad (7)$$

Then, the delay-free DMPC problem  $\mathcal{P}^i(k)$  is defined for any vehicle  $i \in \mathbb{V}$  at time step  $k$  as follows.

**Problem 1. Delay-Free DMPC Problem  $\mathcal{P}^i(k)$ .** Calculate

$$J^i(\Gamma^i(k)) = \text{Min}_{\{\mathbf{u}_k^i(\cdot), \mathbf{x}_k^i(\cdot)\}} J^i(\Gamma^i(k), \mathbf{u}_k^i(\cdot)) \quad (8)$$

subject to (for  $p = 0, 1, 2, \dots, N-1$ )

$$\mathbf{x}_k^i(p+1) = f(\mathbf{x}_k^i(p), \mathbf{u}_k^i(p)); \quad \mathbf{x}_k^i(0) = \mathbf{x}^i(k), \quad (9a)$$

$$\mathbf{x}_k^i(p) \in \mathbb{X}^i, \quad \mathbf{u}_k^i(p) \in \mathbb{U}^i, \quad (9b)$$

$$\mathbf{x}_k^i(N) \in \mathbb{X}_f^i, \quad (9c)$$

where  $\mathbb{X}^i$ ,  $\mathbb{U}^i$ , and  $\mathbb{X}_f^i \subseteq \mathbb{X}^i$  denote the set of admissible states, inputs, and final states (terminal region), respectively, for the  $i$ th vehicle.

Given  $\mathbb{E}$  and initial condition of vehicle  $i$ , do:

- (1)  $k \leftarrow 0$ .
- (2) Measure  $\mathbf{x}^i(k)$  and update the information set  $\Gamma^i(k)$  based on (4).
- (3) Solve  $\mathcal{P}^i(k)$  and predict the control and state trajectories  $\mathbf{u}_k^i(\cdot)$  and  $\mathbf{x}_k^i(\cdot)$ .
- (4) Send the predicted state trajectory  $\mathbf{x}_k^i(\cdot)$  to all followers  $j$  (where  $(j, i) \in \mathbb{E}$ ).
- (5) Apply the control action  $\mathbf{u}_k^i(0)$  to vehicle  $i$ .
- (6) Receive the trajectory  $\mathbf{x}_k^j(\cdot)$  from leader(s)  $j$  (where  $(i, j) \in \mathbb{E}$ ).
- (7)  $k \leftarrow k + 1$ . GOTO step 2.

ALGORITHM 1: Delay-free DMPC.

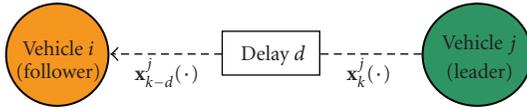


FIGURE 2: Delayed intervehicle communication between two neighboring vehicles.

**2.3.2. Delay-Free DMPC Algorithm.** Each vehicle  $i$  at each sampling period solves the decentralized problem  $\mathcal{P}^i(k)$ . The output of the optimization problem  $\mathcal{P}^i(k)$  is the predicted input and state trajectories of vehicle  $i$ . The generated state trajectory is sent to followers for cooperation purposes, and the first optimal input is applied to the individual vehicle  $i$ . The following algorithm is presented for the online implementation of the delay-free DMPC problem above. The algorithm is formulated for the  $i$ th vehicle as in Algorithm 1.

*Initialization.* For  $k = 0$  at steps 2 and 3, assume  $N_i^j = 0$  (because no information from leader(s) is available).

Algorithm 1 is repeated until the assigned targets are reached. The targets are assumed to be known and assigned to each agent *a priori*.

### 3. Delay-Compensated DMPC (DC-DMPC)

This section develops a new delay-compensated DMPC (DC-DMPC) approach. The safety issues are also discussed.

**3.1. Delayed Cost Function and Information Set.** Assume at time step  $k$  the vehicle  $i$  receives the information from neighbor  $j$  with time-delay  $d$ , where  $d \in \mathbb{N}$  and  $d \geq 1$ , see Figure 2. Then, the term large communication delays denotes the time delays which are larger than the sampling period, that is,  $d > 1$ .

When a vehicle receives the delayed information from neighbors the information set is updated as follows (compare with (5)):

$$\Gamma_D^i(k) = \{\mathbf{x}^i(k), \mathbf{x}_{k-d}^{-i}(\cdot)\}. \quad (10)$$

The *information set*  $\Gamma_D^i(k)$  represents updated information available to the  $i$ th vehicle at time step  $k$ . It implies at time step  $k$  that each vehicle  $i$  has access to its own delay-free

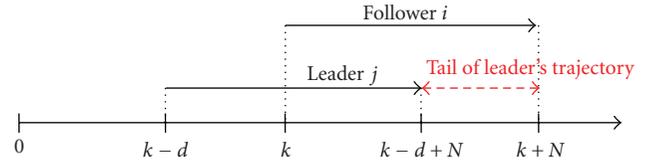


FIGURE 3: Tail of the leader's trajectory in the presence of large communication delay  $d$ .

information and delayed information from neighbors. Then, the decentralized delayed cost function for all  $i \in \mathbb{V}$  at time step  $k$  is defined as follows (compare with (6)):

$$\begin{aligned} J_D^i(\Gamma_D^i, \mathbf{u}_k^i(\cdot)) &= \sum_{p=0}^{N-1} \left( \|\mathbf{x}_k^i(p) - \mathbf{x}^{T,i}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_k^i(p)\|_{\mathbf{R}}^2 \right) \\ &+ \|\mathbf{x}_k^i(N) - \mathbf{x}^{T,i}\|_{\mathbf{P}}^2 \\ &+ \sum_{j|(i,j) \in \mathbb{E}} \left( \sum_{p=0}^{N-d} \|\mathbf{x}_k^i(p) - \mathbf{x}_{k-d}^j(p) - \mathbf{r}^{i,j}\|_{\mathbf{S}}^2 \right). \end{aligned} \quad (11)$$

The communication delay affects the cooperation cost for the periods when the leader's trajectory is not available. In fact, for the tail of the cost function during  $[N-d : N]$  the leader's trajectory is not available due to communication delay to calculate the cost function. More precisely, assume at time step  $k$  the vehicle  $i$  receives the information from neighbor  $j$  with time-delay  $d$ , where  $d \in \mathbb{N}$  and  $d \geq 1$ . Then, the trajectory of neighbor  $j$  for only the time interval  $[k-d, k+N-d]$  is available to vehicle  $i$ , while according to cost function of (6) vehicle  $i$  needs the trajectory of neighbor  $j$  for the entire interval  $[k, k+N]$ . Hence for the portion  $[k+N-d, k+N]$  the trajectory of  $j$  is not available due to the delay (see Figure 3). When the time delay is small this lack of information is not important, but for large communication delays the tail of cost function during  $[k+N-d, k+N]$  becomes large and as shown by simulations in Section 4 it can lead to poor performance and unsafe behaviors (see also [13, 17]). One remedy to this problem is proposed here by estimating the tail of the cost function by including extra decision variables in the cost function.

**3.2. Delay-Compensated Cost Function.** The main idea with the proposed cost function is that it replaces the missing information due to communication loss, with the decision variables. Then, the decision variables are used to approximate the missing information. More specifically, communication losses prevent access to updated leader's trajectory, and then the available delayed neighbor's trajectory is utilized. However, the delayed leader's trajectories do not cover the entire time slot required by follower to calculate its cost function; hence, a delay-compensated cost function is proposed as follows (for the  $i$ th vehicle in the team at time step  $k$ ):

$$\begin{aligned} J_{DC}^i & \left( \Gamma_D^i(k), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right) \\ & = J_D^i \left( \Gamma_D^i(k), \mathbf{u}_k^i \right) \\ & \quad + J_{\text{Tail}}^i \left( \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right), \end{aligned} \quad (12)$$

where  $J_D^i$  is calculated from (11) and

$$\begin{aligned} J_{\text{Tail}}^i & \left( \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right) \\ & = \sum_{j|(i,j) \in \mathbb{E}} \left[ \sum_{p=N-d+1}^N \left( \left\| \mathbf{x}_k^{j,i}(p) - \mathbf{x}_k^{j,i}(p) - \mathbf{r}^{i,j} \right\|_S^2 \right. \right. \\ & \quad \left. \left. + \left\| \mathbf{x}_k^{j,i}(p) - \mathbf{x}^{T,j} \right\|_Q^2 + \left\| \mathbf{u}_k^{j,i}(p) \right\|_R^2 \right) \right. \\ & \quad \left. + \left\| \mathbf{x}_k^{j,i}(N) - \mathbf{x}^{T,j} \right\|_P^2 \right]. \end{aligned} \quad (13)$$

The subscript "D" stands for "Delay". The delay-compensated decentralized cost function  $J_{DC}^i$  includes two main parts.

- (1) The delayed cost  $J_D^i$  which includes the cost of local vehicle  $i$  and the cost of neighboring vehicles calculated from the delayed information. Literally, this part is calculated using the information available from communication (and measurement sensors for local vehicle).
- (2) The tail cost  $J_{\text{Tail}}^i$  (13) which calculates the cost associated with the neighbors over the tail of the cost function during  $[N-d+1 : N]$  where information is not available due to the communication loss/delay. In fact, the added extra decision variables are used to calculate this part.

Although it is assumed that there is model uncertainty and communication noise, the lack of updated information due to communication loss will impose an uncertainty on the updated leader's trajectory, because, in the event of communication loss, the followers will utilize the leader's delayed trajectory, which may differ from the updated unavailable trajectories. Such uncertainty which is derived from the mismatch between updated and delayed trajectories may lead to poor performance of the decentralized DMPC.

**3.3. Safety Guarantee Using Tube DMPC.** For the particular case of formation control, consider a group of vehicles that are required (1) to keep certain relative positions (local objective) and (2) to visit a set of waypoints (global objective). In general, incorporating the collision avoidance constraint into the optimization problem of DMPC is not trivial because of the nonconvexity nature of the distance constraint. To address this problem, in [18] a hybrid rule-based extension of the decentralized model predictive control (DMPC) is proposed to avoid possible collisions. Also, in [19] a mixed integer linear programming (MLIP) approach is utilized to handle the nonconvex collision avoidance constraint using a DMPC architecture.

The problem becomes even more serious when the large communication delays exist because communication delay leads to a lack of updated information on the trajectory of neighboring vehicles and this can make the formation unsafe and put the team in jeopardy. Hence, a discussion on the collision avoidance is required.

In practice, normally the desired distance in formation control is chosen large enough to ensure safety against overshoots and uncertainties. In this paper another approach which is applicable to both delay-free and delayed situations is proposed. In this approach, instead of using an assumed trajectory for neighboring vehicles, a tube-shaped trajectory is assumed around the trajectory of leader, where the tube radius is the radius of the reachable set of leaders. This way, the leader is put in a protection zone where the followers cannot enter. The radius of tube is a function of maneuverability and communication delay; the smaller the communication delays the thinner the tube. Note that at the end the desired relative distance in the formation is chosen to be larger than the tube radius. This is the main distinction between this work and the cooperative or noncooperative approaches aiming at online estimation of the reachable set. In contrast, here the tube is calculated offline and is used for the online purposes.

The idea of tube MPC was first used to calculate a robust bound on the states for uncertain systems [8, 9].

To avoid conservative reachability sets (thick tubes) the maneuverability of each vehicle is restricted, and then the reachable set of each vehicle is calculated; in fact, the leader imposes an input constraint to its optimization problem, that is, at any time instant the computed input trajectories of leader do not deviate too far from the previous one.

Note that, although restricting the maneuverability can lead to less conservative results, it can be restrictive to application where an agile vehicle is needed. Hence, for applications where, for example, slow moving ground vehicles are used such as automated highway systems [20] or mobile robots [21, 22], this approach can be beneficial. In such application since aggressive maneuvers are not required, restricting the maneuverability is not prohibitive, and hence the proposed approach is well suited.

**3.3.1. Tube Formulation and Calculation.** This section presents a numerical method for offline calculation of the tube for subsystems with generic dynamics. Any other computational or analytical reachability set calculation method

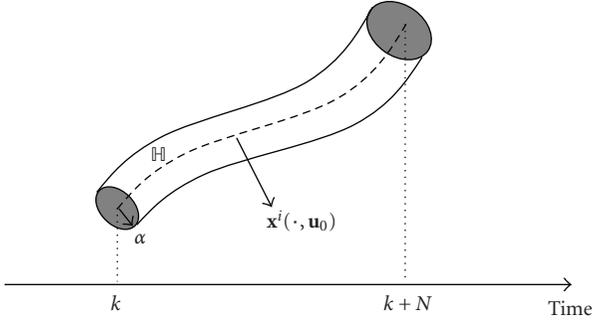


FIGURE 4: A tube  $\mathbb{H}$  around a nominal trajectory, of vehicle  $i$ .

may be used for this purpose. The main advantage of the presented tube calculation algorithm is that it can be computed offline and does not impose significant online computation load. Besides, the available analytical bounds are usually too conservative and are not in the desired format for tube analysis (often a bound on the norm of states is available).

Figure 4 shows the tube  $\mathbb{H}$  around a nominal trajectory  $\mathbf{x}^i(\cdot, \mathbf{u}_0)$ . The tube  $\mathbb{H}$  is formulated as follows (see also [23]):

$$\mathbb{H}_k^i = \left\{ (p, \hat{\mathbf{x}}^i) \in [0 : N] \times \mathbb{R}^n \mid \left| \hat{\mathbf{x}}^i(p, \mathbf{u}) - \mathbf{x}^i(p, \mathbf{u}_0) \right| < \alpha_k^i, \mathbf{u} \in \mathbb{U} \right\}, \quad (14)$$

where  $|\cdot|$  represents the component-wise absolute value of vector and  $\alpha_k^i$  denotes the radius of tube  $\mathbb{H}$  around the trajectory of vehicle  $i$  at time step  $k$ .

The following theorem represents a method for calculating the tube for a generic class of systems. Roughly speaking, tube is a generalization of reachability set; the reachability set is normally calculated around an initial state; however, tube is calculated around a trajectory of state vector. In other words, the deviation from some nominal trajectory over a given time is called tube.

**Theorem 1.** Assume that at time step  $k$  the  $d$  step delayed trajectory of leader  $j$  (i.e.,  $\mathbf{x}_{k-d}^j(\cdot)$ ) is available to vehicle  $i$ . If

$$\left| \mathbf{u}_k^j(p) - \mathbf{u}_{k-d}^j(p+d) \right| \leq \beta^j, \quad p = 0, 1, 2, \dots, N-d, \quad (15)$$

then the trajectory of vehicle  $j$  at time step  $k$  belongs to the tube around  $\mathbf{x}_{k-d}^j(\cdot)$ ; the tube is formulated as:

$$\mathbb{H}_k^{j,i} = \left\{ (p, \hat{\mathbf{x}}_{k-d}^j) \in [0 : N-d] \times \mathbb{R}^n \mid \left| \hat{\mathbf{x}}_{k-d}^j(p) - \mathbf{x}_{k-d}^j(p+d) \right| < \alpha_k^{j,i}(p) \right\}, \quad (16)$$

where  $\mathbb{H}_k^{j,i}$  represents the tube around the trajectory of vehicle  $j$ , calculated by vehicle  $i$  at time step  $k$ . Also,  $\alpha_k^{j,i}$  denotes the tube radius and is the solution of the following maximization problem.

**Problem 2.** Given  $\beta^j$ , calculate for all  $p = 0, 1, 2, \dots, N$

$$\alpha_k^{j,i}(p) = \text{Max}_{\{\hat{\mathbf{x}}_{k-d}^j(\cdot), \hat{\mathbf{u}}_{k-d}^j(\cdot), \mathbf{x}_{k-d}^j(\cdot), \mathbf{u}_{k-d}^j(\cdot)\}} \left[ \sum_{p=0}^{N-d} \left[ f(\hat{\mathbf{x}}_{k-d}^j(p), \hat{\mathbf{u}}_{k-d}^j(p)) - f(\mathbf{x}_{k-d}^j(p), \mathbf{u}_{k-d}^j(p)) \right] \right]$$

subject to

$$\hat{\mathbf{x}}_{k-d}^j(p+1) = f(\hat{\mathbf{x}}_{k-d}^j(p), \hat{\mathbf{u}}_{k-d}^j(p)),$$

$$\hat{\mathbf{x}}_{k-d}^j(t_{k-d}) = \mathbf{x}^j(t_{k-d}) \in \mathbb{X},$$

$$\mathbf{x}_{k-d}^j(p+1) = f(\mathbf{x}_{k-d}^j(p), \mathbf{u}_{k-d}^j(p)),$$

$$\mathbf{x}_{k-d}^j(t_{k-d}) = \mathbf{x}^j(t_{k-d}) \in \mathbb{X},$$

$$\hat{\mathbf{x}}_{k-d}^j(p) \in \mathbb{X}; \quad \mathbf{x}_{k-d}^j(p) \in \mathbb{X},$$

$$\hat{\mathbf{u}}_{k-d}^j(p) \in \mathbb{U}; \quad \mathbf{u}_{k-d}^j(p) \in \mathbb{U},$$

$$\left| \hat{\mathbf{u}}_{k-d}^j(p) - \mathbf{u}_{k-d}^j(p) \right| \leq \beta^j. \quad (17)$$

*Proof.* At time step  $k$  vehicle  $j$  uses the input trajectory  $\mathbf{u}_k^j(\cdot)$  which yields the state trajectory  $\mathbf{x}_k^j(\cdot)$ , then the solution of the differential equation (8) is calculated as:

$$\mathbf{x}_k^j(p) = \mathbf{x}^j(k) + \sum_{s=0}^{p-1} f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) \quad p = 0, 1, \dots, N. \quad (18)$$

But if vehicle  $j$  uses the assumed input  $\hat{\mathbf{u}}_k^j(\cdot)$ , then the trajectory will be as follows:

$$\hat{\mathbf{x}}_k^j(p) = \mathbf{x}^j(k) + \sum_{s=0}^{p-1} f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \quad p = 0, 1, \dots, N. \quad (19)$$

Subtracting (18) from (19) yields

$$\hat{\mathbf{x}}_k^j(p) - \mathbf{x}_k^j(p) = \sum_{s=0}^{p-1} \left[ f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) - f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \right] \quad p = 0, 1, \dots, N. \quad (20)$$

By finding  $\alpha_k^{j,i}$  the tube  $\mathbb{H}_k^{j,i}$  is determined.  $\alpha_k^{j,i}$  is the upper bound on  $|\hat{\mathbf{x}}_k^j(p) - \mathbf{x}_k^j(p)|$  and then is calculated by solving the following maximization problem.

*Problem 3.* Given  $\mathbf{u}_k^j(\cdot)$ ,  $\mathbf{x}_k^j(\cdot)$  and  $\beta^j$  calculate for all  $p = 0, 1, 2, \dots, N$

$$\alpha_k^{j,i}(p) = \text{Max}_{\{\hat{\mathbf{x}}_k^j(\cdot), \hat{\mathbf{u}}_k^j(\cdot)\}} \left[ \sum_{s=0}^{p-1} \left[ f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) - f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \right] \right]$$

subject to

$$\begin{aligned} \hat{\mathbf{x}}_k^j(p+1) &= f(\hat{\mathbf{x}}_k^j(p), \hat{\mathbf{u}}_k^j(p)), \\ \hat{\mathbf{x}}_k^j(0) &= \mathbf{x}^j(0) \quad \hat{\mathbf{x}}_k^j(p) \in \mathbb{X} \quad \hat{\mathbf{u}}_k^j(p) \in \mathbb{U} \\ |\hat{\mathbf{u}}_k^j(p) - \mathbf{u}_k^j(p)| &\leq \beta^j. \end{aligned} \quad (21)$$

In optimization Problem 3 a nominal trajectory  $\mathbf{x}_k^j(\cdot)$  is given as input. In other words the optimal value depends on any nominal trajectory. On the other hand, at any time step DMPC generates a new trajectory which is served as nominal trajectory in Problem 3. Hence, Problem 3 should be modified to be independent of any nominal trajectory and be applicable for a general trajectory  $\mathbf{x}_k^j(\cdot)$ . Hence, considering  $\hat{\mathbf{x}}_k^j(\cdot)$  and  $\hat{\mathbf{u}}_k^j(\cdot)$  as new decision variables in the optimization Problem 3 and shifting the time from  $k$  to  $k-d$  will lead to Problem 2. This way, the tube can be computed offline and used for online applications, and no online computation is required.

**3.3.2. Tube Calculation Algorithm.** The application of Theorem 1 allows calculating the tube  $\mathbb{H}_k^{j,i}$  around the trajectory of each neighboring vehicle. Assume that the control input at each sample time for vehicle  $j$  is bounded as follows:

$$|\mathbf{u}_k^j(p) - \mathbf{u}_{k-1}^j(p+1)| \leq \mu, \quad p = 0, 1, \dots, N-2, \quad (22)$$

where  $\mu$  is a vector with appropriate dimension and is called the *maneuverability* vector. It allows restricting the maneuverability of each vehicle during the time segments when the actual predicted trajectory of vehicle may not be available to followers. The neighboring vehicles by considering such restrictions can compute a bound on the trajectories. Then, if at any time step  $k$  vehicle  $i$  receives the information from leader  $j$  with  $d$  steps time delay, that is,  $\mathbf{x}_{k-d}^j(\cdot)$ , then the results of Theorem 1 are applicable by finding  $\beta^j(\mu, d)$ . To find  $\beta^j(\mu, d)$  the input constraint (22) can be used sequentially as follows (the superscript  $j$  is dropped temporarily):

$$\begin{aligned} -\mu &\leq \mathbf{u}_k(p) - \mathbf{u}_{k-1}(p+1) \leq \mu, \\ -\mu &\leq \mathbf{u}_{k-1}(p+1) - \mathbf{u}_{k-2}(p+2) \leq \mu, \\ &\vdots \\ -\mu &\leq \mathbf{u}_{k-d+1}(p+d-1) - \mathbf{u}_{k-d}(p+d) \leq \mu, \\ &\vdots \\ -d \cdot \mu &\leq \mathbf{u}_k(p) - \mathbf{u}_{k-d}(p+d) \leq d \cdot \mu, \\ &p = 0, 1, 2, \dots, N-d. \end{aligned} \quad (23)$$

Hence,

$$\beta^j = d \cdot \mu, \quad p = 0, 1, 2, 3, \dots, N-d. \quad (24)$$

Therefore, we have

$$\beta^j(p, \mu, d) = \begin{cases} 0, & p = 0, \\ \mu, & p = 1, \\ 2\mu, & p = 2, \\ \vdots & \vdots \\ (d-1)\mu, & p = d-1, \\ d\mu, & p = d, d+1, \dots, N-d. \end{cases} \quad (25)$$

Then  $\alpha_k^{j,i}$ , the radius of tube  $\mathbb{H}_k^{j,i}$ , is calculated from Problem 2 after calculating  $\beta^j$  from (25). The output of Problem 2 is the trajectory of vector  $\alpha_k^{j,i}(\cdot)$  and will be used in the online DMPC algorithm.

The main benefit of using this procedure for tube calculation is that calculation of  $\alpha_k^{j,i}$  does not impose any online computation load as it can be computed offline. The only parameter which may be unknown prior to mission is the time delay  $d$ . Hence, the tube can be computed for different delays, and the results are given to the DMPC controller as some tabulated data, to be used in online applications. In this way no online computation is imposed.

**3.3.3. Nonconvexity Avoidance.** Using a tube instead of a trajectory in the cost function (12) can lead to nonconvexity for the optimization problem due to the nonconvex nature of the tube. Thus, in order to avoid the nonconvexity, in the cost function (12),  $\mathbf{x}_{k-d}^j(p)$  is not modified (or replaced by tube  $\mathbb{H}_k^{j,i}$ ), instead the desired relative position  $\mathbf{r}^{i,j}(p)$  will be modified as follows:

$$\mathbf{r}^{i,j}(p) \leftarrow \mathbf{r}^{i,j}(p) + \text{sign}(\mathbf{r}^{i,j}(p)) \cdot \alpha_k^{j,i}(p), \quad (26)$$

where  $\alpha_k^{j,i}$  is the tube radius at time step  $k$  and is calculated from Problem 2. In fact, the margin  $\alpha_k^{j,i}(p) \geq 0$  is added to the desired distance to ensure the safety. Since  $\mathbf{r}^{i,j}(\cdot)$  is the relative position vector,  $\alpha_k^{j,i}$  is multiplied by the diagonal matrix  $\text{sign}(\mathbf{r}^{i,j}(\cdot))$  to ensure that adding  $\alpha_k^{j,i}$  increases the magnitude of  $\mathbf{r}^{i,j}(\cdot)$ ; in fact  $\mathbf{r}^{i,j}(\cdot)$  is preserved, and extra margin is added.

**3.4. Delay-Compensated DMPC Problem.** The delay-compensated DMPC (DC-DMPC) problem  $\mathcal{P}_{\text{DC}}^i(k)$  is defined below at time step  $k$  for any  $i$ th vehicle. The outputs of this decentralized optimization problem are (1) the input and (2) the state trajectory of the local vehicle over the prediction horizon and (3) the trajectory of neighboring vehicles during the tail of the cost function.

*Problem 4.* DC-DMPC Problem  $\mathcal{P}_{\text{DC}}^i(k)$ . Given  $\Gamma_D^i(t_k)$  calculate

$$J_{\text{DC}}^i(\Gamma_D^i(k)) = \underset{\{\mathbf{x}_k^i(\cdot), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N)\}}{\text{Min}} J_{\text{DC}}^i \\ \times \left( \Gamma_D^i(k), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N), \right. \\ \left. \mathbf{u}_k^{-i}(N-d:N) \right) \quad (27)$$

subject to

(i) for  $p = 0, 1, 2, \dots, N-1$ : (9a)–(9c)

(ii) and for  $p = N-d, \dots, N$  and  $(i, j) \in \mathbb{E}$ :

$$\mathbf{x}_k^{j,i}(p+1) = f(\mathbf{x}_k^{j,i}(p), \mathbf{u}_k^{j,i}(p)); \quad (28a)$$

$$\mathbf{x}_k^{j,i}(N-d) = \mathbf{x}_{k-d}^j(N) \quad (28b)$$

$$\mathbf{x}_k^{j,i}(p) \in \mathbb{X}^j, \quad \mathbf{u}_k^{j,i}(p) \in \mathbb{U}^j \quad (28b)$$

$$\mathbf{x}_k^{j,i}(N) \in \mathbb{X}_f^j, \quad (28c)$$

(iii) and for  $p = 0, 1, 2, \dots, N-2$ :

$$\left| \mathbf{u}_k^i(p) - \mathbf{u}_{k-1}^i(p+1) \right| \leq \boldsymbol{\mu}, \quad (28d)$$

where constraints (9a)–(9c) are the same as in the delay-free DMPC problem  $\mathcal{P}^i(t_k)$  and correspond to the trajectory for calculating the delayed cost function  $J_D^i$ . Constraints (28a) and (28b) correspond to the tail cost function  $J_{\text{Tail}}^i$  (13). Constraint (28c) is the terminal constraint and is the same as (9c) in  $\mathcal{P}^i(t_k)$ . Moreover, constraint (28d) is added for safety guarantee purpose, where  $|\cdot|$  represents the component-wise absolute value of vector and  $\boldsymbol{\mu}$  is a vector with appropriate dimension and called the *maneuverability* vector. Constraint (28d) is called *maneuverability* constraints (such constraint is also known as *compatibility* constraints in similar literatures, [24, 25]). It allows restricting the maneuverability of each vehicle and hence calculating the tube as explained in Theorem 1. Roughly speaking, constraints (28a)–(28c) are added for cooperation performance improvement, and constraint (28d) is added for safety purpose.

*Remark 2.* Expressing equations (28d) in the norm form instead of component-wise absolute value leads to more conservative results in practice.

**3.5. DC-DMPC Algorithm.** Algorithm 2 is presented for the online implementation of the proposed DC-DMPC problem  $\mathcal{P}_{\text{DC}}^i(k)$ . The algorithm is formulated for the  $i$ th vehicle; in fact, all vehicles run this algorithm during the mission simultaneously.

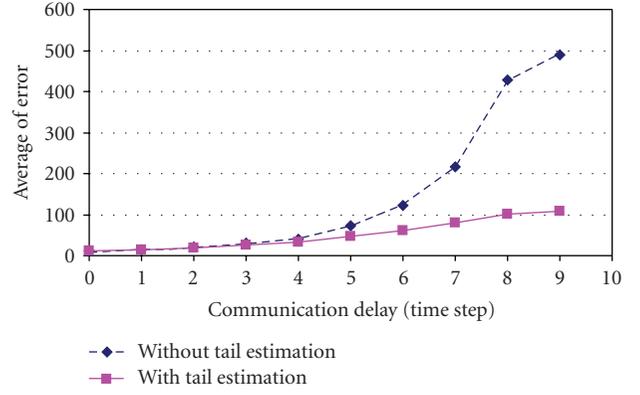


FIGURE 5: Average of error versus communication delay for a triangle formation of three vehicles.

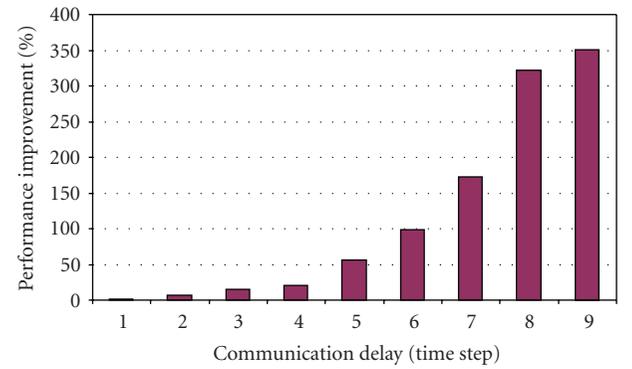


FIGURE 6: Percentage of average of error versus communication delay for a triangle formation of three vehicles.

*Initialization.* For  $k = 0$  to  $k = d-1$  at steps 2, 3, and 4 assume that  $N_k^i = 0$ . For  $k = 0$  to  $k = d-2$  at step 7 assume  $N_k^i = 0$  (because no information from leader(s) is available).

This algorithm is a modified version of Algorithm 1 and handles the large communication delays; it also provides the safety guarantee by executing the step 3 using the tube DMPC approach.

*Remark 3.* In step 2 of Algorithm 2, to measure the time delays, it is assumed that all the vehicles are equipped with synchronous clocks. Then at each sampling time, each vehicle, in addition to its predicted trajectory, broadcasts the time it is calculated the trajectory. Hence, the knowledge on the amount of delay is not required *a priori*.

## 4. Simulation Results

In this section, the proposed approach is tested on the formation problem of a fleet of unmanned vehicles with the following 2DOF dynamics. This dynamics can capture the motion of a mini-hovercraft equipped with two sets of thrusters to control the side and forward motions; also, a

Given  $\mu$ ,  $\mathbb{E}$  and initial condition of vehicle  $i$ , do:

- (1)  $k \leftarrow 0$ .
- (2) Measure  $\mathbf{x}^i(k)$ , calculate delay from leader and update the delayed *information vector* of (10).
- (3) **Calculate the tube of all leaders:**
  - For  $\forall j \in \mathbb{V}$  where  $(i, j) \in \mathbb{E}$ 
    - (i) Calculate  $\beta^j(\cdot, \mu, d)$  from (25).
    - (ii) Calculate  $\alpha_k^{j,i}(\cdot)$  from the data provided by offline solution of Problem 2.
    - (iii) Update  $\mathbf{r}^{i,j}(\cdot) \leftarrow \mathbf{r}^{i,j}(\cdot) + \text{sign}(\mathbf{r}^{i,j}(\cdot)) \cdot \alpha_k^{j,i}(\cdot)$  and update  $\mathbf{r}^{i,j}$  in the cost function (12).
- END
- (4) Solve  $\mathcal{P}_{DC}^i(\mathbf{k})$  and calculate the control action  $\mathbf{u}_k^i(\cdot)$  and  $\mathbf{x}_k^i(\cdot)$ .
- (5) Send the state trajectory  $\mathbf{x}_k^i(\cdot)$  to the followers where  $(j, i) \in \mathbb{E}$ .
- (6) Apply the control action  $\mathbf{u}_k^i(0)$  to individual vehicle  $i$ .
- (7) Receive  $\mathbf{x}_{k-d}^j(\cdot)$  from leaders where  $(i, j) \in \mathbb{E}$ .
- (8)  $k \leftarrow k + 1$ . Goto step 2.

ALGORITHM 2: DC-DMPC.

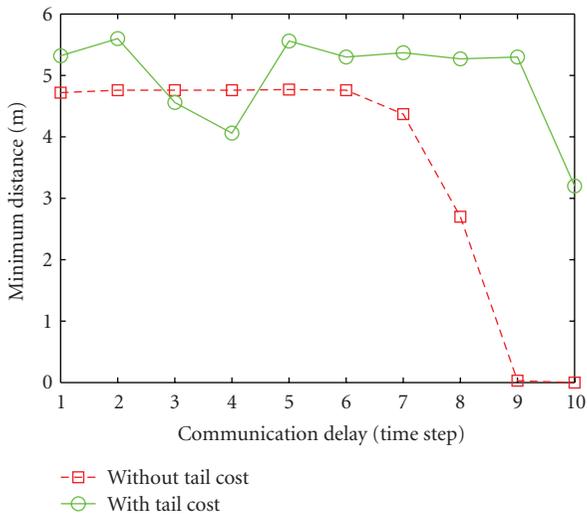


FIGURE 7: Minimum distance between vehicles in the presence of a communication delays.

small damping term is added as the typical resistance of water to hovercraft motion is small:

$$\begin{aligned}
 \dot{x}_1 &= x_2, \\
 \dot{x}_2 &= -0.1x_2 + u_1, \\
 \dot{x}_3 &= x_4, \\
 \dot{x}_4 &= -0.1x_4 + u_2,
 \end{aligned} \tag{29}$$

where  $x_1$  and  $x_2$  represent the components of position vector in  $x - y$  coordinate and  $x_3$  and  $x_4$  are their corresponding velocity components. The input vector is given by  $\mathbf{u} = [u_1, u_2]$ . This dynamics is discretized by sampling time  $h = 0.1$  sec. The MPC problem is solved using optimization toolbox of MATLAB. When Algorithm 1 is used a set of  $2N$  decision variables are used for each vehicle to model the control inputs (because each vehicle has only two control inputs). Once Algorithm 2 is used, two extra sets of decision

variables are used;  $2N + 2d$  decision variables are used. It means the computation grows with communication delay.

In the first simulation example, it is desired to examine the effect of the tail cost added to the cost function. The simulation was run for two cases.

- (1) Using the cost function *without the tail cost*. In this case the control input is set to  $u = 0$  for the tail of the cost function. The extra decision variables for tail cost estimation are not included in the optimization problem.
- (2) Using the cost function *with the tail cost*. In this case the tail of the cost function (12) is estimated using the extra decision variables in the optimization problem  $\mathcal{P}_{DC}^i(k)$ .

The matrix penalties in the cost function are chosen as follows:  $\mathbf{Q} = \mathbf{I}$  (where  $\mathbf{I}$  is identity matrix), and  $\mathbf{R} = 0.1 \mathbf{0I}$  then

$$\mathbf{P} = \begin{bmatrix} 0.8634 & 1.447 & 0.000 & 0.000 \\ 1.4471 & 2.800 & 0.000 & 0.000 \\ 0.0000 & 0.000 & 0.863 & 1.447 \\ 0.0000 & 0.000 & 1.447 & 2.800 \end{bmatrix} \tag{30}$$

and  $\mathbf{S} = 20\mathbf{I}$ . The final penalty matrix  $\mathbf{P}$  is calculated from the Lyapunov equation [15]. The optimization horizon is chosen to be  $N = 10$ . In all cases no disturbances, sensor noise, or model uncertainty is considered in the simulations in order to focus on the effect of the communication delay.

A triangular leader-follower formation of three vehicles is first considered, where the moving vehicles are controlled to form an isosceles triangle with edges of 10, 7.02, and 7.02 m. The actual cooperation cost is chosen as the cooperation performance index of any follower  $i \in \mathbb{V}$  as follows:

$$I^i(k) = \sum_{j|(i,j) \in \mathbb{E}} \left\| \mathbf{x}^i(k) - \mathbf{x}^j(k) - \mathbf{r}^{i,j}(k) \right\|_{\mathbf{S}}^2 \quad \forall i \in \mathbb{V}. \tag{31}$$

This performance index represents the deviation (error) from the desired formation shape; hence, a smaller  $I^i$  is

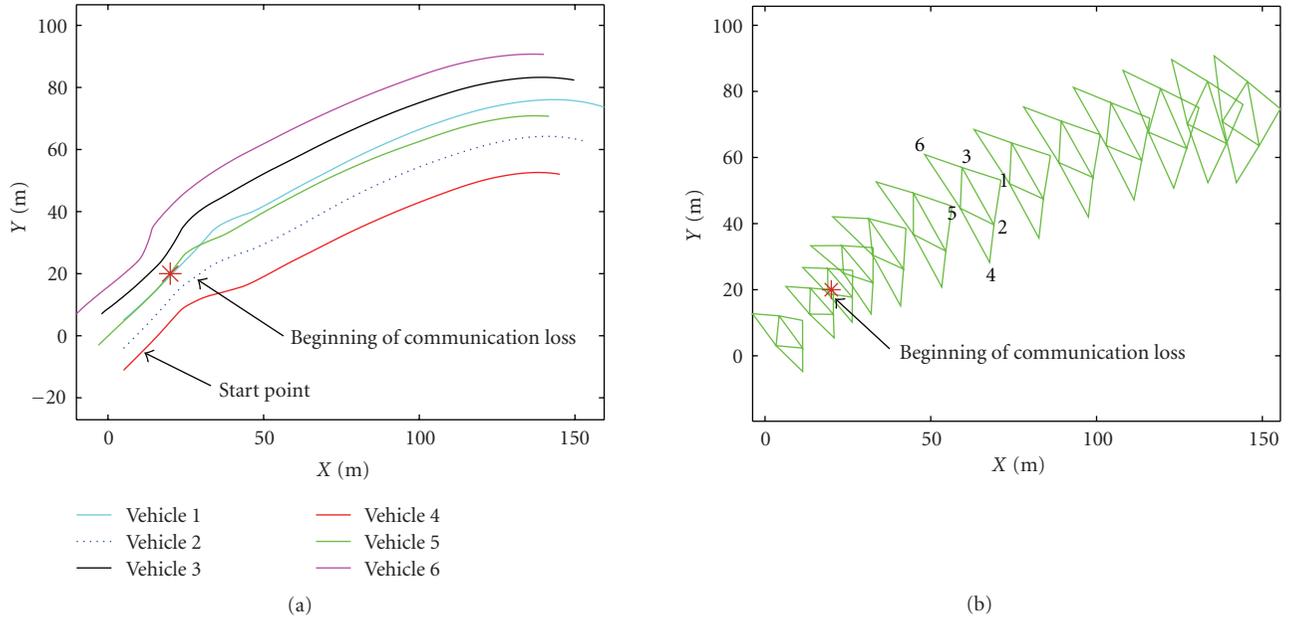


FIGURE 8: Trajectory (a) and formation snapshot (b) of six vehicles in triangular formations when Algorithm 2 is used.

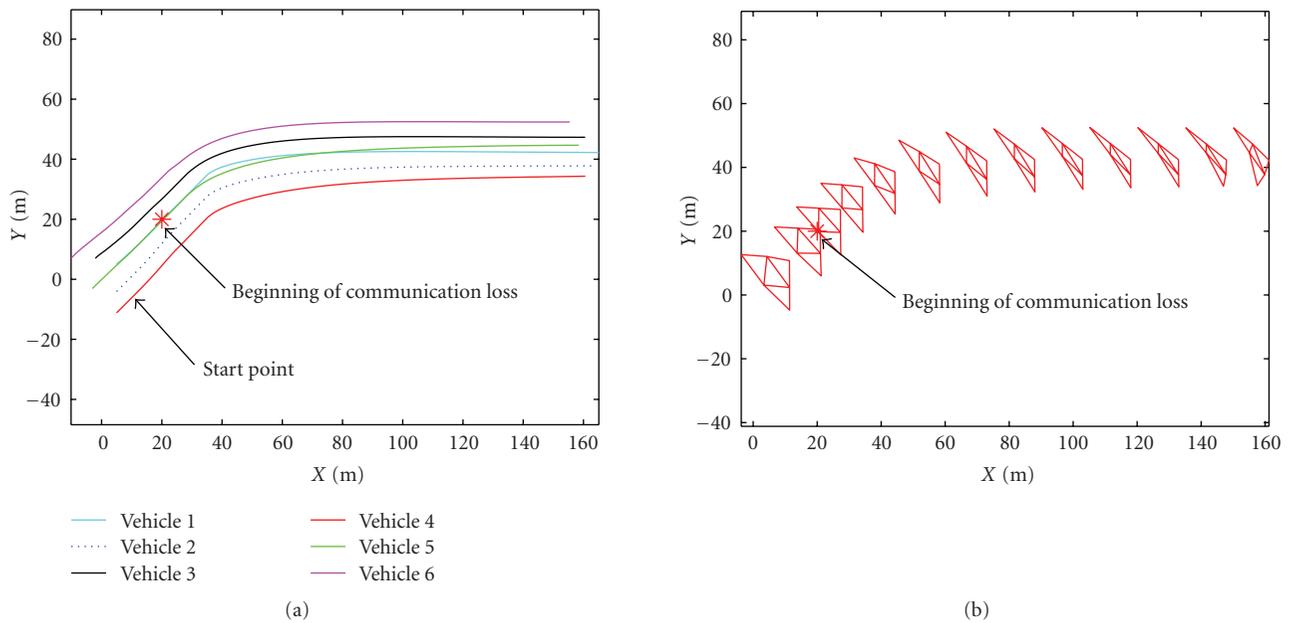


FIGURE 9: Trajectory (a) and formation snapshot (b) of six vehicles in triangular formations when Algorithm 1 is used.

desired. Then, the total error of the group at time step  $k$  is presented as follows:

$$I(k) = \sum_{i \in \mathcal{V}} I^i(k). \quad (32)$$

**4.1. Performance Improvement.** The simulation was repeated for cases with different communication delays, and the results are gathered in Figures 5 and 6. Figure 5 shows the average of the formation error (32) with each point representing a single simulation. It can be seen from

Figure 5 that estimating the tail of the cost function leads to smaller errors. Figure 6 shows the percentage of performance improvement (error reduction) when the tail estimation method is used comparing with the case where no tail estimation is performed. It shows also that for large communication delays the proposed approach can reduce the error by 350%. It can also enhance the stability of the formation; for this particular example, it is seen that if the communication delay is increased to around  $d = 10$  time steps, the formation becomes unstable when using the cost function without the

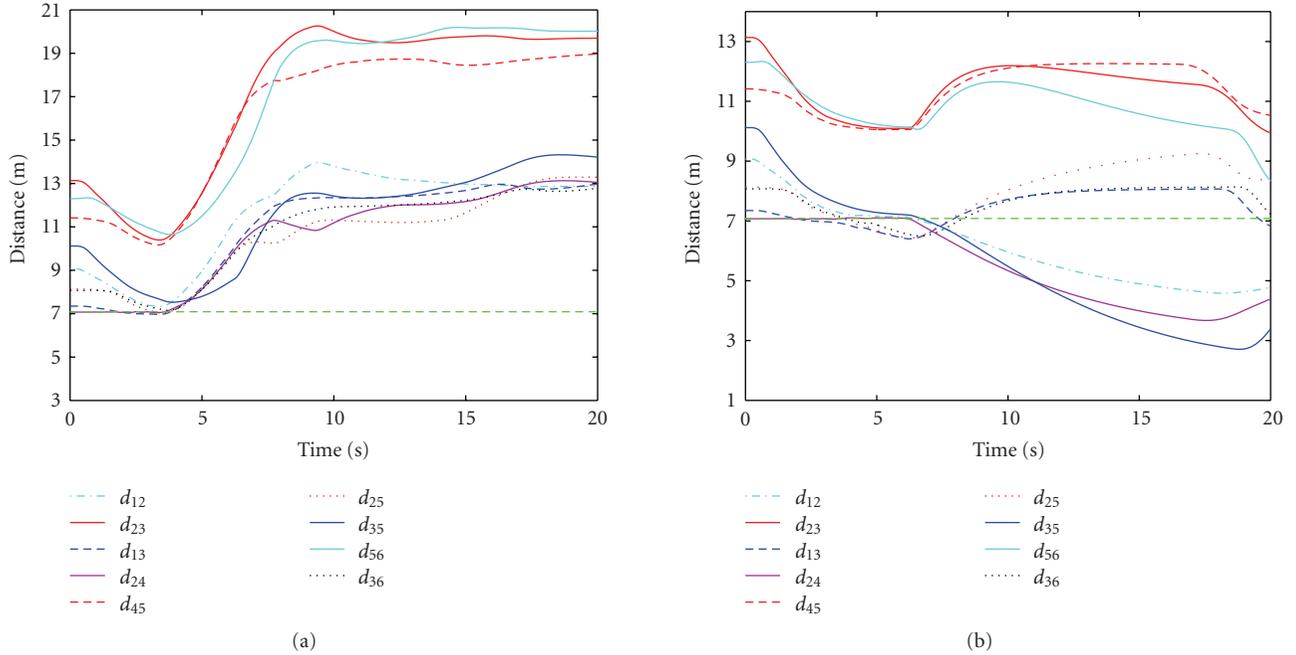


FIGURE 10: Distances between each pair of vehicles for Algorithm 2 (a) and Algorithm 1 (b).

tail cost. However, it is still stable with the proposed cost function including the tail cost. This result is consistent with that of [26, 27] where a final cost is added to the cost function for formation stability, although they did not consider communication delays. The overall trend of the graphs in Figure 5 shows that the error goes up with delay.

Note that, since we assume no uncertainty in the model, the main source of error arises from communication delay and decentralized nature of the problem. The communication delay forces the decentralized controllers to use delayed information instead of updated information, which adds some degree of uncertainty into the problem, leading to error and degraded cooperation performance. Also, the decentralized nature of the problem restricts the decentralized controllers to access all the information about neighboring vehicles intention and control calculation procedure. For example, vehicles do not take into account the interaction graph topology of neighboring vehicles, in order to reduce the computation load. The second source of error does not exist in a centralized framework.

**4.2. Safety Using Tube DMPC.** It is observed that in some simulations in presence of large communication delays although adding the final cost can lead to better performance and stable formation, the vehicles may get too close to each other and collide. For example, the minimum distance among the vehicles for a set of simulations is depicted in Figure 7. The desired distance between each pair of neighboring vehicles is 7.07 m. As seen even for the case where tail cost estimation is used the vehicles may get too close to each other.

Therefore, in the next simulation the proposed Algorithm 2 which includes the tube DMPC is used.

This case involves the triangular formation control of six vehicles. The communication graph topology is set as follows:

$$\begin{aligned} \mathbb{V} &= \{1, 2, 3, 4, 5, 6\}, \\ \mathbb{E} &= \{(2, 1), (3, 1), (4, 2), (5, 4), (6, 3)\}. \end{aligned} \quad (33)$$

The results are shown in Figure 8 through Figure 10. In this case, two sets of way points are considered to be visited by the fleet. At the beginning the intervehicle communication delay is  $d = 0$  but after 3 sec (around point (22, 22)) communication loss begins which leads to  $d = 7$  time step intervehicle communication delay. As seen from Figure 8 the vehicles start to keep a larger distance, and the formation is expanded for safety upon communication loss. Figure 9 shows the same scenario when Algorithm 1 is used.

The distances between each pair of neighboring vehicles are shown in Figure 10 for two cases: (1) Algorithm 1 and (2) Algorithm 2. It is desired that vehicles keep a 7.07 m distance from neighbors. As seen from Figure 10(b) in the case of Algorithm 1, vehicles get too close to each other and may collide. However, Algorithm 2 offers a loose (Figure 8) but safe formation (Figure 10(a)) as the consequence of using tube DMPC for safety.

For this case the minimum desired separation between two neighboring vehicle is 7.07 m. The corresponding required minimum relative distance, when Algorithm 2 is used for a delay of  $d = 7$  time-step, is about 10 m as measured from Figure 10(a). Hence, for this case the added margin due to using tube DRHC is calculated as  $10 - 7.07 \approx 3$  m; then for such relatively large delay, the added margin is about 42.8%, which does not seem conservative. In the presence of model uncertainty and communication noise this margin will grow.

## 5. Conclusions

A new delay-compensated decentralized model predictive controller (DC-DMPC) is proposed that can address communication delays arising from any source including packet losses. The proposed approach provides two key features. The first feature is that the tail of the cost function is estimated for performance improvement in the presence of the large intervehicle communication delays. Simulation results show that this method can lead to 350% improvement in the cooperation performance. The added online computation load depends on the communication delay and should be taken into account for any particular application. The second feature is the development of the tube-based MPC to provide guaranteed formation safety against possible collisions in the presence of large intervehicle communication delays. Together, these results provide a new approach to deal with the large intervehicle communication delays in DMPC problems that ensures safe formations and improved cooperation performance.

## Acknowledgment

This paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Strategic Project Grant (STPGP 350889-07) leading by the third author and a Discovery Project Grant by the second author.

## References

- [1] H. A. Azadi, B. W. Gordon, and Y. M. Zhang, "Decentralized receding horizon control for cooperative multiple vehicles subject to communication delay," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1959–1965, 2009.
- [2] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of discrete-time agents with delayed information exchange: a receding-horizon approach," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 4, pp. 4274–4279, Atlantis, Paradise Island, Bahamas, December 2004.
- [3] E. Franco, T. Parisini, and M. M. Polycarpou, "Stable receding-horizon cooperative control of a class of distributed agents," in *Proceedings of the American Control Conference (ACC '05)*, vol. 7, pp. 4673–4678, IEEE Press, June 2005.
- [4] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control (CDC-ECC '05)*, pp. 2206–2211, IEEE Press, December 2005.
- [5] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 324–338, 2008.
- [6] A. Richards and J. P. How, "Implementation of robust decentralized model predictive control," in *AIAA Guidance, Navigation, and Control Conference*, pp. 4929–4941, AIAA, August 2005.
- [7] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control (CDC-ECC '05)*, pp. 6680–6685, IEEE Press, December 2005.
- [8] I. Alvarado, D. Limon, T. Alamo, and E. F. Camacho, "Output feedback robust tube based MPC for tracking of piecewise constant references," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2175–2180, IEEE Press, December 2007.
- [9] P. Trodden and A. Richards, "Robust distributed model predictive control using tubes," in *Proceedings of the American Control Conference*, pp. 2034–2039, Minneapolis, Minn, USA, June 2006.
- [10] X. Zhang, R. Xu, C. Kwan, L. Haynes, Y. Yang, and M. M. Polycarpou, "Fault tolerant formation flight control of UAVs," *International Journal of Vehicle Autonomous Systems*, vol. 2, no. 3-4, pp. 217–235, 2004.
- [11] M. Innocenti, L. Pollini, and F. Giulietti, "Management of communication failures in formation flight," *Journal of Aerospace Computing, Information and Communication*, pp. 19–35, 2004.
- [12] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.
- [13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [15] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
- [16] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, 2007.
- [17] D. Garagić, "On delay-dependent stability of a swarm of networked autonomous vehicles under communication constraints," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 297–301, IEEE Press, June 2005.
- [18] T. Keviczky, B. Vanek, F. Borrelli, and G. J. Balas, "Hybrid decentralized receding horizon control of vehicle formations," in *Proceedings of the American Control Conference*, pp. 3358–3363, Minneapolis, Minn, USA, June 2006.
- [19] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 4, pp. 4286–4291, Atlantis, Paradise Island, Bahamas, December 2004.
- [20] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2065, 2003.
- [21] L. Pallottino, V. G. Scordio, and A. Bicchi, "Decentralized cooperative conflict resolution among multiple autonomous mobile agents," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 5, pp. 4758–4763, Atlantis, Paradise Island, Bahamas, December 2004.
- [22] Y. Yoon, H. Kim, J. Shin, T. Choe, and Y. Park, "Communication in distributed model predictive collision avoidance," in *Proceedings of the 1st International Conference on Robot Communication and Coordination*, Athens, Greece, 2007.

- [23] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1996.
- [24] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: a distributed implementation," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 1995–2002, Atlantis, Paradise Island, Bahamas, December 2004.
- [25] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
- [26] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 4, pp. 4631–4636, IEEE Press, December 2002.
- [27] D. Gu and E. Yang, "A suboptimal model predictive formation control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1295–1300, IEEE Press, 2005.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

