Safe Path Planning in the Presence of Large Communication Delays Using Tube Model Predictive Control

Hojjat A. Izadi[1], Brandon W. Gordon[2] and Youmin Zhang[3]
Concordia University, Montreal, Quebec, H3G 1M8, Canada

In this paper, a conflict resolution problem for multiple vehicles subject to large communication delays is investigated in a decentralized model predictive control (DMPC) framework. Using model predictive control, each vehicle plans its own future path towards its assigned target and predicts the possible collisions with neighboring vehicles. The planned trajectories are then redefined to resolve possible collisions. Conflict prediction and resolution is accomplished by providing cooperation and coordination among the vehicles. The cooperation part involves exchanging the planned trajectories at each sample time among the vehicles. The coordination part consists of imposing a maneuverability constraint in each vehicle's optimization problem. The possible collisions are determined by calculating the neighboring vehicle's reachable set consisting of a tube shaped trajectory around the neighbor's trajectory. To resolve potential conflicts each vehicle predicting a possible collision restricts its maneuverability so that its calculated tube does not intersect with those of its neighbors. Computer simulations illustrate the effectiveness of the proposed approach.

Nomenclature

A, B	=	state transition matrices
d	=	discrete communication time delay
$\mathbb{E}$	=	set of edges in graph topology
$\mathbb{G}$	=	graph topology
$\mathbb{H}^i$	=	tube around the trajectory of vehicle i
$J^{\boldsymbol{i}}$	=	cost function of i^{th} vehicle
N_v	=	number of vehicles
N_n^i	=	number of neighbors of vehicle i
$\mathscr{P}^i(t_k)$	=	decentralized model predictive control (DMPC) problem
P	=	final state penalty
Q	=	state penalty
R	=	input penalty
$\boldsymbol{r}^i$	=	state vector of target point of vehicle i
S	=	coupling state penalty
$\boldsymbol{x}$	=	state vector
$\boldsymbol{x}^{\boldsymbol{i}}$	=	state vector of i^{th} vehicle
$\boldsymbol{x}_k^i(t)$	=	the predicted state vector of i^{th} vehicle at time t, computed at time step t_k

[1] Postdoctoral Fellow, Department of Mechanical and Industrial Engineeringt, 1515 St Catherine West, ha_izadi@encs.concordia.ca.

[2] Associate Professor, Department of Mechanical and Industrial Engineeringt, 1515 St Catherine West, bwgordon@encs.concordia.ca.

[3] Associate Professor, Department of Mechanical and Industrial Engineeringt, 1515 St. Catherine West, ymzhang@encs.concordia.ca, Senior Member AIAA.

$\mathbb{X}^i$	=	set of admissible states of vehicle *i*
$\mathbb{U}^i$	=	set of admissible inputs of vehicle *i*
$\mathbb{X}^i_f$	=	set of final states of vehicle *i*
T	=	prediction horizon
$\boldsymbol{u}$	=	input vector
$\boldsymbol{u}^{\mathbf{i}}$	=	input vector of *i*th vehicle
$\mathbb{V}$	=	set of vehicles
φ	=	the state transition matrix
$\Gamma^i(t_k)$	=	information set of *i*th vehicle at time t_k
$\boldsymbol{\mu}^i$	=	the *maneuverability parameter* of vehicle *i*
Λ^i	=	reachable set of vehicle *i*
δ	=	execution horizon (sampling time)
τ	=	communication delay

I. Introduction

The demands in decentralized path planning and conflict resolution span a wide range of application including automated air traffic control,[1-3] automated road traffic control,[4,5] mobile robots,[6,7] and cooperative UAVs.[8,9] Many approaches have been used to design safe trajectory planner for such applications.[10-13] Among them model predictive control (MPC) -also known as receding horizon control- has found some attention. However, one problem concerns its weakness in handling non-convex constraints arising from collision avoidance objectives. To address this problem, in Ref. 9 a hybrid rule-based extension of the decentralized receding horizon control (DRHC) is proposed to avoid possible collisions. In Ref. 8 a mixed integer linear programming (MLIP) approach is utilized to handle the non-convex collision avoidance constraint using a decentralized model predictive control architecture. Safety is provided in Ref. 6 by seeking new maneuvers such that all conflicts are avoided. In Ref. 14, a DRHC methodology is proposed for trajectory tracking with safe conflict resolution for multiple autonomous vehicles. The collision avoidance is performed by including the avoidance function in the cost function. In order to address the symmetric cases (as a special form of non-convexity) a novel limit cycle method is also implemented in Ref. 14 to modify segments of the desired trajectories. This leads to feasible solutions for the optimization problem performed by each vehicle. Most recently, in Ref. 15, using invariant sets a set of emergency maneuvers is computed to avoid collisions whenever the feasibility is lost.

The attempts in mentioned references[8, 9, 14, 15] imply a new interest for using MPC for safe path planning in spite of its weakness in handling non-convex constraints and its computational complexity. Using MPC for developing the conflict resolution algorithms is motivated by three main property of MPC. First, its *predictive* nature allows predicating the possible collisions. Second, its unique advantage for *handling the constraints* potentially provides a significant advantage over *pursuit-evasion* game approaches where a backward reachable set is approximated to be avoided in order to avoid some unsafe target set. Since the mentioned approaches try to calculate a backward reachable set corresponding to entire input set, this may lead to large computed backward reachable sets in some cases. This does not allow vehicles to operate in the vicinity of each other safely. However, using MPC it is possible to restrict the maneuverability by imposing some constraints on the inputs and then calculate the reachability set for smaller set of inputs. This leads to smaller enforced protection zone around each vehicle, and hence, implies a larger set of feasible solutions. More precisely, the maneuverability can be restricted so as to resolve the conflicts. The third property is that it is easy to provide *cooperation* through the cost function and constraint among the vehicles to avoid collisions. In this paper, all these three advantages of MPC are employed to develop a novel Tube DMPC approach with conflict resolution capability in presence of large communication delays. Such large communication delays may arise due to failure in communication devices,[16-18] or limited communication bandwidth,[19] which should be taken into account for practical considerations.

In Ref. 20 and Ref. 21 a tube-based DMPC is proposed to provide formation safety under communication failures where the desired relative distances in the cost function are robustified to be larger than the radii of the reachable sets; although it has shown great efficiency to provide safe formation in simulations, in general it may not guarantee the collision avoidance as achieving the local minimum of the coupling cost -providing cooperation- is computationally expensive. Also, the overshoots in transient response may lead to collisions. In this paper, the methodology presented in Ref. 20 and Ref. 21 is extended to a general set of non-formation scenarios.

The approach presented in this paper for collision avoidance in a decentralized framework is based on a simple idea: "*compute my reachable set and avoid that by restricting your maneuverability*". Technically speaking, a tube is calculated around the delayed trajectory of the neighboring vehicles; since the neighboring vehicles may not stay on the delayed paths, the radius of the tube is non-zero; the radius of the tube depends on the communication delay and the maneuverability (and uncertainties which are not considered here). Then the neighboring vehicles are assumed to avoid each others tube.

This paper is organized as follows. In Section II a delayed-DMPC is formulated. Section III presents the general approach for collision avoidance. Then in Section IV a LMI approach for calculation of the reachable set of vehicles with linear dynamics is introduced and the tube is calculated by connecting all the reachable sets over the prediction horizon. Finally, an algorithm is proposed by which each vehicle determines its maneuverability so that its tube does not intersect with neighbor's tube and hence the collision is avoided. Section V demonstrates the proposed approach through computer simulations.

II. Decentralized Model Predictive Control (DMPC)

Consider a team of vehicles with uncoupled dynamics cooperating to resolve potential collisions. Each vehicle is equipped with three main units: measurement sensors, communication device and computation resource. The measurement sensors of each vehicle measure only its own states. The communication device is used to gather the information from the neighbors and communicate with human operators through a shared communication channel. Using a computation resource, each vehicle solves an optimization DMPC problem at each sampling time based on its instant states (from sensors) and the trajectory of its neighboring vehicles (from communication channel). Furthermore, it is assumed that only the vehicles which have physical interactions such as collision avoidance, will exchange the information.

A. Interaction and Information Exchange Graphs

The interaction between cooperative vehicles is usually represented by an "*interaction graph*" including two main elements: nodes and edges. Considering a set of N_v vehicles cooperating to perform a common mission, the i^{th} vehicle corresponds to the i^{th} node of the graph. If an edge (i, j) connecting the i^{th} and j^{th} node is present, it means that the i^{th} and j^{th} vehicles have an interaction and hence they may have coupling term in their cost function and/or in their constraints, and communicate with each other. This relationship is termed as *neighborhood* for the i^{th} and j^{th} vehicles. This leads to the interaction graph as follows:

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

where $\mathbb{V}$ is the set of nodes (vehicles) and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ the set of edges (i,j), with $i, j \in \mathbb{V}$. The interaction graph is *undirected* i.e. $(i, j) \in \mathbb{E}$ implies $(j, i) \in \mathbb{E}$ even though it does not appear in $\mathbb{E}$. Also, let N_n^i denotes the number of neighbors of vehicle *i*. It is assumed that the *interaction graph* is set *a priori* using an efficient mission manager.

B. DMPC Notation and Terminology

With Model Predictive Control (MPC) a cost function is minimized over a future finite time called *prediction horizon T*, or in short *horizon*. The first portion of the computed optimal input is applied to the plant during a period of time called the *execution horizon*, δ, or *sampling period*. The reader is referred to Ref. 22 for a comprehensive review of the MPC schemes.

The *execution horizon* δ is assumed to be equal to the *communication period*; this provides synchronization between the communication rate and the sampling rate of MPC. Then, the discrete timing is denoted by t_k where $t_{k+1} = t_k + \delta$, $k \in \mathbb{N}$ and $t_0 = 0$.

The possible state vectors are introduced as follows:

- $\boldsymbol{x}^i(t)$: the actual state vector of i^{th} vehicle at time *t*.
- $\boldsymbol{x}_k^i(t)$: the predicted state vector of i^{th} vehicle at time *t*, computed at time step t_k .

Also the sequence of these states over the prediction horizon is called the state trajectory of vehicle $i \in \mathbb{V}$ and is represented by:

$$\boldsymbol{x}_k^i(\cdot) = \left\{ \boldsymbol{x}_k^i(t) \mid t \in [t_k, t_k + T] \right\} \tag{2}$$

Further, let the following represents the concatenated state trajectories of the neighbors of i^{th} vehicle at time t_k :

$$\boldsymbol{x}_k^{-i}(\cdot) = \left\{ \boldsymbol{x}_k^j(\cdot) \mid j \in \mathbb{V},\ (i, j) \in \mathbb{E} \right\} \tag{3}$$

The same notation will be used for input vector.

C. DMPC Formulation

Figure 1 shows the inter-vehicle communication between two neighboring vehicles and the information exchanged at any time t_k for delay-free condition.

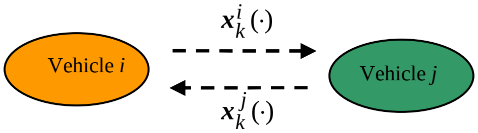

Figure 1. *The inter-vehicle communications between two neighbors in delay-free condition*

Equation (3) presents the information that vehicle *i* receives from its neighbors. The vehicle *i* needs its own instant states as well to solve the DMPC optimization problem; then the *information set* of the *i*th vehicle for the case of delay-free DMPC is introduced as follows:

$$\Gamma^i(t_k) = \left\{ \boldsymbol{x}^i(t_k),\ \boldsymbol{x}_k^{-i}(\cdot) \right\} \tag{4}$$

The *information set* $\Gamma^i(t_k)$, collects the state vector of *i*th vehicle and the concatenated state trajectory of the neighbors, $\boldsymbol{x}_k^{-i}(\cdot)$. The former is provided through on-board sensors and the latter is provided through communication.
With the DMPC algorithm, always the communicated messages are subject to at least one step delay as the computed trajectories are not available instantly even though an infinite communication bandwidth is available. Then, assume that the information communicated among the vehicles is subject to time-delay τ , it is assumed that $(d-1)\delta \le \tau \le d\delta$ where $d \in \mathbb{N}_1 = \mathbb{N} - \{0\}$ by which *d* denotes the discrete communication delay (see Figure 2). Hence, the delayed *information set* is updated as follows (compare with Eq. (4)):

$$\Gamma^i(t_k) = \left\{ \boldsymbol{x}^i(t_k),\ \ \boldsymbol{x}_{k-d}^{-i}(\cdot) \right\} \tag{5}$$

Set $\Gamma^i(t_k)$ represents the updated information available to the *i*th vehicle at time t_k . It implies at time t_k each vehicle *i* has access to its own delay-free information but the delayed information of its neighbors.

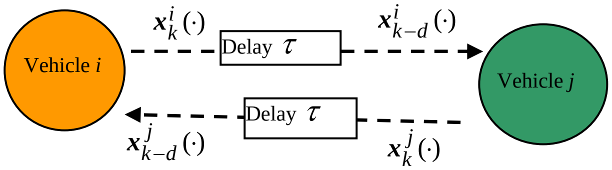
$\boldsymbol{x}_k^i(\cdot)$
Delay τ
$\boldsymbol{x}_{k-d}^i(\cdot)$
Vehicle *i*
Vehicle *j*
Delay τ
$\boldsymbol{x}_{k-d}^j(\cdot)$
$\boldsymbol{x}_k^j(\cdot)$

Figure 2. *Synchronization of communication delay with MPC timing*

The decentralized cost function for the *i*th vehicle in the team at time t_k is defined as follows:

$$J^i(\boldsymbol{x}^i(t_k), \boldsymbol{u}_k^i(\cdot)) = \int_{t_k}^{t_k+T} \left(\left\| \boldsymbol{x}_k^i(t) - \boldsymbol{r}^i \right\|_Q^2 + \left\| \boldsymbol{u}_k^i(t) \right\|_R^2 \right) dt + \left\| \boldsymbol{x}_k^i(t_k+T) - \boldsymbol{r}^i \right\|_P^2 \tag{6}$$

where $\| \boldsymbol{x} \|_Q^2 = \boldsymbol{x}^T Q \boldsymbol{x}$, $P>0$, $Q>0$, and $R>0$ are symmetric matrices. Also, $\boldsymbol{r}^i$ is the state vector of target of vehicle *i*.

1. *DMPC Problem*

Assume the following equation describes the linear dynamics for homogeneous vehicles:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t); \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{7}$$

Then, the DMPC problem $\mathscr{P}^i(t_k)$ is defined for the *i*th vehicle at time t_k as follows:

Problem 1: DMPC Problem $\mathscr{P}^i(t_k)$ *(* $i \in \mathbb{V}$ *)*:

$$\underset{\{\boldsymbol{u}_k^i(\cdot),\,\boldsymbol{x}_k^i(\cdot)\}}{\text{Min}} \quad J^i(\boldsymbol{x}^i(t_k), \boldsymbol{u}_k^i(\cdot)) \tag{8}$$

subject to (during $t \in [t_k, t_k + T]$):

$$\dot{\boldsymbol{x}}_k^i(t) = A\boldsymbol{x}_k^i(t) + B\boldsymbol{u}_k^i(t) \; ; \; \boldsymbol{x}_k^i(t_k) = \boldsymbol{x}^i(t_k) \tag{9a}$$

$$\boldsymbol{x}_k^i(t) \in \mathbb{X}^i, \; \boldsymbol{u}_k^i(t) \in \mathbb{U}^i(t_k) \tag{9b}$$

$$\boldsymbol{x}_k^i(t_k + T) \in \mathbb{X}_f^i \tag{9c}$$

In Eq. (8), J^i is calculated from Eq. (6), vectors $\mathbb{X}^i \subseteq \mathbb{R}^n$, $\mathbb{U}^i \subseteq \mathbb{R}^m$ and $\mathbb{X}_f^i \subseteq \mathbb{X}^i$ denote the set of admissible states, inputs and terminal states (terminal region) respectively, for the i^{th} vehicle.

The admissible input set $\mathbb{U}^i(t_k)$ may change at each time step to answer the needs for collision avoidance. This is the key issue within this paper and is used as a degree of freedom to provide coordination among vehicles to restrict their maneuverability and avoid collisions. If the collision avoidance problem is seen as a pursuit-evasion game, then $\mathbb{U}^i(t_k)$ is maximized for evader and is minimized for pursuer.

2. DMPC Algorithm

Each vehicle $i \in \mathbb{V}$ at each sampling time solves the decentralized problem $\mathscr{P}^i(t_k)$ using its updated states and the neighbor's trajectories; the output of this optimization problem is the input and state trajectory of vehicle $i \in \mathbb{V}$ on the interval $[t_k, t_k + T]$. After generating these trajectories the DMPC controller applies only the first portion of its own computed input during $[t_k, t_{k+1}]$, and sends the generated state trajectory to all neighboring vehicles for collision avoidance purposes. The following algorithm presents the on-line implementation of the DMPC. The algorithm is formulated for the vehicle $i \in \mathbb{V}$ and all vehicles run this algorithm during the mission simultaneously:

Algorithm 1: DMPC

```
1-  Let $k$=0, and GOTO step 3.
2-  Receive the trajectory $\boldsymbol{x}_{k-d}^j(\cdot) \quad (i, j) \in \mathbb{E}$ (with appropriate $d$) from neighbors.
3-  Measure $\boldsymbol{x}^i(t_k)$ and update $\Gamma^i(t_k)$ .
4-  Solve $\mathscr{P}^i(t_k)$ and generate $\boldsymbol{u}_k^i(\cdot)$ and $\boldsymbol{x}_k^i(\cdot)$ .
5-  Send the trajectory $\boldsymbol{x}_k^i(\cdot)$ to the neighboring vehicles.
6-  Apply the control action for individual vehicle $i$ during $[t_k, t_{k+1}]$:
7-  $k$=$k$+1. Goto step 2.
```

For initialization during $t < d$, the 2nd and 3rd step are not executed and it is assumed that $\mathbb{E} = \varnothing$.

This algorithm is repeated until the assigned targets are reached. The targets are assumed to be known and assigned to each vehicle *a priori*.

III. General Problem Statement

The reachable set of vehicle *i* at time *t* is formulated as:

$$\Lambda^i\,(t, \boldsymbol{x}_0, \mathbb{U}^i) = \left\{ \boldsymbol{x}^i(t) \,|\, \dot{\boldsymbol{x}}^i(s) = A\boldsymbol{x}^i(s) + B\boldsymbol{u}^i(s),\; \boldsymbol{x}^i(t_0) = \boldsymbol{x}_0, \boldsymbol{u}^i \in \mathbb{U}^i, s \in [t_0, t] \right\} \tag{10}$$

Figure 3 shows a graphical sketch of the reachable set of 3 neighboring vehicles at time *t*:

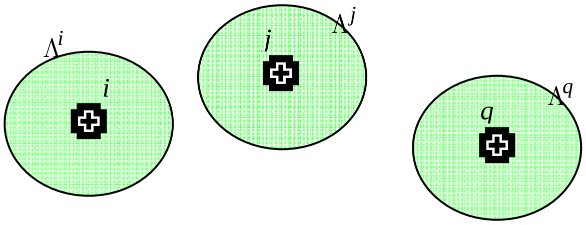

Figure 3. *The reachable sets for three vehicles*

It is obvious that if the reachable sets of neighboring vehicles have no intersection ($\Lambda^i \cap \Lambda^j \cap \Lambda^q = \varnothing$) then no collision will happen at time *t*. In order to take the advantage of the predictive nature of MPC and predict the possible collisions during the prediction horizon, all reachable sets are computed over the prediction horizon and connected together which results in the *tube* concept; in fact, *tube* is formed by connecting the reachable sets over the prediction horizon. Figure 4 shows how by connecting the reachable sets a *tube* is formed. From Figure 4 it is clear that if the tube of neighboring vehicles have intersection (*i.e.* $\mathbb{H}^i \cap \mathbb{H}^j \cap \mathbb{H}^q \neq \varnothing$) then a collision is possible and conflict resolution is required. In general, sufficient condition for collision avoidance is that: $\mathbb{H}^1 \cap \mathbb{H}^2 \cap \mathbb{H}^3 \cap ....\cap \mathbb{H}^{N_v} = \varnothing$.

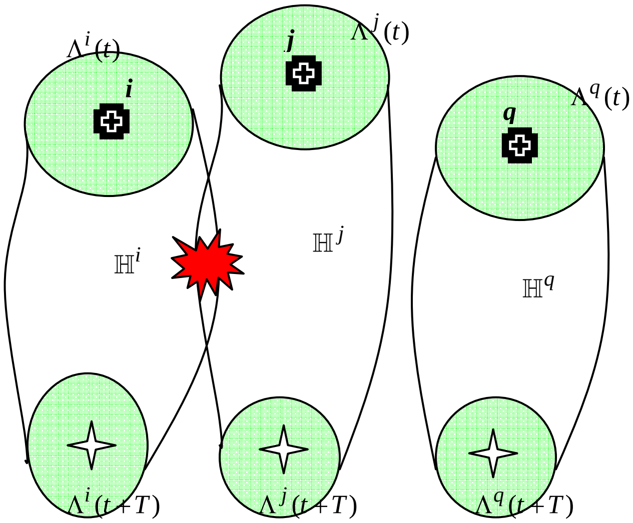

Figure 4. *The tube is formed by connecting the reachable sets over the prediction horizon*

The *tube* analysis allows each vehicle to predict the possible collisions and hence, change the plan to avoid the collisions; the collision avoidance policy is based on setting the admissible input set $\mathbb{U}^i$ so that the tubes do not intersect. Then at each time step each vehicle $i \in \mathbb{V}$:

1- *Calculates the neighbor's tube from delayed information by assuming limited maneuverability for neighbors.*

2- *Sets $\mathbb{U}^i$ (maneuverability) so that its tube does not intersect with neighbor's tube.*

Hence, it is required to obtain the tube radius as a function of delay and maneuverability; the tube is calculated around the delayed trajectories. It is also desired to use an algorithm for tube calculation which does not impose significant online computation and is implementable and scalable in a decentralized framework. Finally, the collision avoidance algorithm is integrated with the DMPC algorithm.

IV. Safety Guarantee Using Tube-DMPC

A large communication delay implies the lack of updated information on the trajectory of neighboring vehicles; the lack of information can make the trajectories unsafe and put the team in jeopardy. However, in such cases, if a constraint is imposed on the maneuverability of each vehicle, then the reachable set of neighboring vehicles can be estimated and restricted using the available, albeit delayed, information. Hence, instead of using an assumed trajectory for neighboring vehicles a tube is assumed, where the tube is the reachable set of neighbors when the input constraint is applied. The smaller the communication delays the thinner the tube. The idea of tube MPC is used normally for uncertain systems to calculate a robust bound on the states.[25-27]

A. LMI Approximation of Reachable Set

Several methods for calculating the reachable set of dynamical systems are developed in the literatures,30,31 which calculate the reachable set analytically or numerically. Any of them can be used to be integrated with DMPC presented in this paper. For the purpose of this paper, a method for approximation of the reachable set of LTI systems is presented in the following lemma:

Lemma 1: For LTI system of Eq. (7) if:

$$\mathbb{U}^i(t) = \left\{ \boldsymbol{u} \mid \int_{t_0}^{t} \boldsymbol{u}'\boldsymbol{u}\,dt \le \beta \right\} \tag{11}$$

then the reachable set $\Lambda^i(t, \boldsymbol{x}_0, \mathbb{U}^i)$ is bounded by the ellipsoid Υ centered at $\boldsymbol{x}_0$:

$$\Upsilon = \left\{ \boldsymbol{x} \mid (\boldsymbol{x} - \boldsymbol{x}_0)' M (\boldsymbol{x} - \boldsymbol{x}_0) \le \beta \right\} \tag{12}$$

where M is symmetric and diagonal solution of the following LMI:

$$M > 0 \qquad \begin{bmatrix} A'M + MA & MB \\ B'M & -I \end{bmatrix} \le 0 \tag{13}$$

Also if dynamics presented in Eq. (7) is controllable then M=W^{-1} where W is the *controllability Gramian*.

Proof: see Ref. 28, page 78.

B. Tube Formulation and Calculation

The state vector of each vehicle contains two types of variables: 1) the states which directly involve in physical collision such as positions and are denoted by vector $\boldsymbol{\xi}$, 2) the rest of the states such as velocities, and are denoted by vector $\boldsymbol{\upsilon}$; hence: $\boldsymbol{x} = [\boldsymbol{\xi}, \boldsymbol{\upsilon}]'$. In this paper, *tube* is referred to as an extraction of reachable set which includes only the position states $\boldsymbol{\xi}$. Figure 5 shows the tube $\mathbb{H}$ around a nominal trajectory $\boldsymbol{\xi}(t, \boldsymbol{u}_0)$. The *tube* $\mathbb{H}$ is formulated as follows (see also Ref. 24):

$$\mathbb{H} = \left\{ (t, \boldsymbol{\xi}) \in [t_0, T] \times \mathbb{R}^p \mid \left| \boldsymbol{\xi}(t, \boldsymbol{u}) - \boldsymbol{\xi}(t, \boldsymbol{u}_0) \right| < \boldsymbol{\alpha}(t) \right\} \tag{14}$$

where $|\cdot|$ represents the component wise absolute value and $\boldsymbol{\alpha}(t) \in \mathbb{R}^p$ is the radius of tube at time t; also, p is the dimension of ξ (for a 2-D motion p=2).

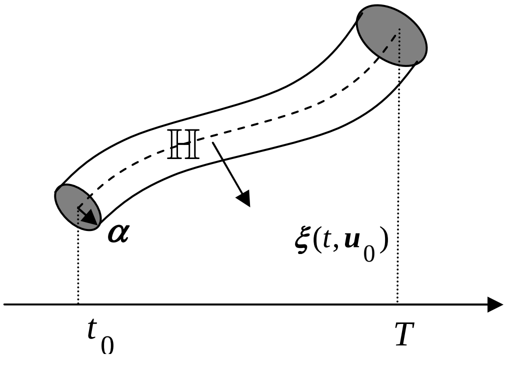

Figure 5. *A tube around a nominal trajectory*

The following theorem presents a method for calculating the tube around the delayed trajectories of neighboring vehicles obeying linear dynamics.

Theorem 1: Assume at time t_k the d step delayed trajectory of neighbor j (i.e. $\boldsymbol{x}^j_{k-d}(\cdot)$) is available to vehicle i. If:

$$\int_{t_{k-d}}^{t_k} \left[\boldsymbol{u}_k^j(t) - \boldsymbol{u}_{k-d}^j(t)\right]' \left[\boldsymbol{u}_k^j(t) - \boldsymbol{u}_{k-d}^j(t)\right] dt \le \beta^j \tag{15}$$

then the trajectory of vehicle *j* at time t_k belongs to the tube around the delayed trajectory of neighbor *j* calculated by vehicle *i* as:

$$\mathbb{H}_k^{j,i} = \left\{(t,\boldsymbol{\xi}) \in [t_k, t_{k-d} + T] \times \mathbb{R}^p \mid \left|\boldsymbol{\xi}(t,u) - \boldsymbol{\xi}_{k-d}^j(t)\right| < \boldsymbol{\alpha}^{j,i}(t)\right\} \tag{16}$$

where $\mathbb{H}_k^{j,i}$ represents the tube around the trajectory of vehicle *j*, calculated by vehicle *i* at time t_k, also:

$$\boldsymbol{\alpha}^{j,i}(t) = \left[\sqrt{\frac{\beta^j}{m_{11}}}, \sqrt{\frac{\beta^j}{m_{22}}}, \ldots\ldots, \sqrt{\frac{\beta^j}{m_{pp}}}\right]' \tag{17}$$

where $M = diag(m_{11}, m_{22}, \ldots\ldots m_{nn})$, is a solution of the LMI presented in Eq. (13).

Proof: Assume at time t_{k-d} vehicle *j* uses the input trajectory $\boldsymbol{u}_{k-d}^j(\cdot)$ which yields the state trajectory $\boldsymbol{x}_{k-d}^j(\cdot)$; then:

$$\boldsymbol{x}_{k-d}^j(t) = \varphi(t, t_{k-d})\boldsymbol{x}^j(t_{k-d}) + \int_{t_{k-d}}^{t} \varphi(t,s)B\boldsymbol{u}_{k-d}^j(s)ds \qquad t \in [t_{k-d}, t_{k-d} + T] \tag{18}$$

where φ is the state transition matrix. But if vehicle *j* uses the input trajectory $\boldsymbol{u}_{k-d}^j(\cdot) + \Delta\boldsymbol{u}$ -which is assumed by neighbors of *j*- then the trajectory will be different and as shown below:

$$\hat{\boldsymbol{x}}_{k-d}^j(t) = \varphi(t, t_{k-d})\boldsymbol{x}^j(t_{k-d}) + \int_{t_{k-d}}^{t} \varphi(t,s)B[\boldsymbol{u}_{k-d}^j(s) + \Delta\boldsymbol{u}]ds \tag{19}$$

Then the difference between these two trajectories is:

$$\Delta\boldsymbol{x}_{k-d}^j(t) = \hat{\boldsymbol{x}}_{k-d}^j(t) - \boldsymbol{x}_{k-d}^j(t) = \int_{t_{k-d}}^{t} \varphi(t,s)\,B\,\Delta\boldsymbol{u}\,ds \tag{20}$$

Considering Eq. (20), $\Delta\boldsymbol{x}_{k-d}^j(t)$ is the solution of the following LTI:

$$\Delta\dot{\boldsymbol{x}}^j = A\Delta\boldsymbol{x}^j + B\Delta\boldsymbol{u}^j \;\; ; \;\; \Delta\boldsymbol{x}^j(t_{k-d}) = 0 \tag{21}$$

From the assumption of theorem, we have:

$$\int_{t_{k-d}}^{t_k} \left[\Delta\boldsymbol{u}^j(t)\right]' \left[\Delta\boldsymbol{u}^j(t)\right] dt \le \beta^j$$

then using *Lemma 1* the reachable set of LTI system presented in Eq. (21) is bounded by the following ellipsoid:

$$\Upsilon^j(t) = \left\{\boldsymbol{x} \mid [\Delta\boldsymbol{x}_{k-d}^j(t)]' M [\Delta\boldsymbol{x}_{k-d}^j(t)] \le \beta^j \right\} \qquad t \in [t_{k-d}, t_{k-d} + T] \tag{22}$$

Substituting Eq. (20) into Eq. (22) yields:

$$\Upsilon^{j}(t) = \{ \boldsymbol{x} \,|[\hat{\boldsymbol{x}}^{j}_{k-d}(t) - \boldsymbol{x}^{j}_{k-d}(t)]' M [\hat{\boldsymbol{x}}^{j}_{k-d}(t) - \boldsymbol{x}^{j}_{k-d}(t)] \le \beta^{j} \} \quad t \in [t_{k-d}, t_{k-d} + T] \tag{23}$$

Using the ellipsoid formula the radius of ellipsoid of Eq. (23) for vehicle $j \in \mathbb{V}$ in each direction is calculated as follows:

$$\boldsymbol{r}_{\Upsilon^{j}} = \left[\sqrt{\frac{\beta^{j}}{m_{11}}}, \sqrt{\frac{\beta^{j}}{m_{22}}}, ..., \sqrt{\frac{\beta^{j}}{m_{nn}}} \right]' \tag{24}$$

Since the ellipsoid of Eq. (23) over $t \in [t_{k-d}, t_{k-d} + T]$ is equivalent to the tube of Eq. (16); then, abstracting the tube $\mathbb{H}^{j,i}$ from the ellipsoid Υ^{j} is straightforward. For each component of $\boldsymbol{\xi}$ (p^{th} component) one should find the corresponding component in $\boldsymbol{r}_{\Upsilon^{j}}$.

C. Combining Tube and DMPC

Theorem 2: Assume in the DMPC problem the control input varies as follows:

$$\left| \boldsymbol{u}^{j}_{k}(t) - \boldsymbol{u}^{j}_{k\text{-}1}(t) \right| \le \boldsymbol{\mu}^{j} \quad t \in [t_{k}, t_{k\text{-}1} + T] \;\&\; k \in \mathbb{N} \tag{25}$$

where $\boldsymbol{\mu}^{j} \ge 0$ is called the *maneuverability parameter* of vehicle *j*. Then $\beta^{j}(d, \boldsymbol{\mu}^{j})$ for $\forall j \in \mathbb{V}$ in *Theorem 1* can be chosen as follows:

$$\beta^{j}(d, \boldsymbol{\mu}^{j}) = d^{3}\delta[\boldsymbol{\mu}^{j}]'[\boldsymbol{\mu}^{j}] \tag{26}$$

Proof:

$$\begin{aligned} \Delta\boldsymbol{u} &= [\boldsymbol{u}^{j}_{k} - \boldsymbol{u}^{j}_{k-d}] = [(\boldsymbol{u}^{j}_{k} - \boldsymbol{u}^{j}_{k-1}) + (\boldsymbol{u}^{j}_{k-1} - \boldsymbol{u}^{j}_{k-2}) + ... + (\boldsymbol{u}^{j}_{k-d+1} - \boldsymbol{u}^{j}_{k-d})] \\ &\le [\boldsymbol{\mu}^{j} + \boldsymbol{\mu}^{j} + ... + \boldsymbol{\mu}^{j}] = d.\boldsymbol{\mu}^{j} \end{aligned} \tag{27}$$

Hence:

$$\begin{aligned} &\int_{t_{k-d}}^{t_k} \Delta\boldsymbol{u}'\Delta\boldsymbol{u}\,dt = \int_{t_{k-d}}^{t_k} [d.\boldsymbol{\mu}^{j}]'[d.\boldsymbol{\mu}^{j}]dt \le d^{2}\boldsymbol{\mu}^{j\,'}\boldsymbol{\mu}^{j} \int_{t_{k-d}}^{t_k} dt = d^{2}\boldsymbol{\mu}^{j\,'}\boldsymbol{\mu}^{j}(t_{k-d} - t_{k}) = d^{3}\delta[\boldsymbol{\mu}^{j}]'[\boldsymbol{\mu}^{j}] \\ &\Rightarrow \quad \beta^{j}(d, \boldsymbol{\mu}^{j}) = d^{3}\delta[\boldsymbol{\mu}^{j}]'[\boldsymbol{\mu}^{j}] \end{aligned} \tag{28}$$

Then each vehicle *i* uses the formula $\beta^{j}(d, \boldsymbol{\mu}^{j}) = d^{3}\delta[\boldsymbol{\mu}^{j}]'[\boldsymbol{\mu}^{j}]$ in Eq. (26) where $(i, j) \in \mathbb{E}$ to calculate the bound on the inputs of its neighbor *j* as it sees a *d* step delay from *j*; in fact, this is the bound calculated by vehicle *i* then the second superscript is added to indicate that vehicle *i* calculates this bound for neighbor *j*, *i.e.* $\beta^{j,i}(d, \boldsymbol{\mu}^{j}) = d^{3}\delta[\boldsymbol{\mu}^{j}]'[\boldsymbol{\mu}^{j}]$.

Furthermore, the vehicle *i* has access to its updated trajectory with only one step delay and then set *d*=1 which yields $\beta^{i,i}(d, \boldsymbol{\mu}^{i}) = \delta[\boldsymbol{\mu}^{i}]'[\boldsymbol{\mu}^{i}]$ to be used by vehicle *i* for calculation of its own tube. Then the tube of vehicle *j* calculated at time t_k by neighbor *i* from the delayed trajectory $\boldsymbol{\xi}^{j}_{k-d}$, is denoted by $\mathbb{H}^{j,i}_{k}$ in Eq. (16). Also the tube of vehicle *i* calculated at time t_k by itself from one step delayed trajectory $\boldsymbol{\xi}^{i}_{k-1}$ is denoted by $\mathbb{H}^{i,i}_{k}$ and calculated from Eq. (16) by setting *d*=1 and hence using $\beta^{i,i}(d, \boldsymbol{\mu}^{i}) = \delta[\boldsymbol{\mu}^{i}]'[\boldsymbol{\mu}^{i}]$. Then, each vehicle $i \in \mathbb{V}$ chooses its maneuverability $\boldsymbol{\mu}^{i}$ so that the following collision avoidance condition holds:

$$\mathbb{H}^{i,i} \cap \mathbb{H}^{i,j} = \varnothing \quad i, j \in \mathbb{V} \;\;\&\; (i, j) \in \mathbb{E} \tag{29}$$

In fact, each vehicle i calculates the tube around the delayed trajectory of each neighbor and then define its maneuverability $\boldsymbol{\mu}^i$ so that there is no intersection between its tube and the tube of neighboring vehicles. Thanks to the capability of DMPC it is easy to enforce the maneuverability condition of Eq. (25) in the optimization problem via input constraints in Eq. (9b). Hence the input set in DMPC problem $\mathscr{P}^i(t_k)$ is updated as follows:

$$\mathbb{U}^i(t_k)=\left\{\boldsymbol{u}(t)\,|\left|\boldsymbol{u}(t)-\boldsymbol{u}_{k\text{-}1}^j(t)\right|\le\boldsymbol{\mu}^j\ ,\ t\in[t_k,t_{k-1}+T]\right\} \tag{30}$$

If the graph topology is well-connected then the decentralized condition of Eq. (29) can imply condition $\mathbb{H}^1\cap\mathbb{H}^2\cap\mathbb{H}^3\cap.......\cap\mathbb{H}^{N_v}=\varnothing$ and hence collision avoidance satisfaction (the definition of a well-connected graph topology is left for a future research in order to find a condition on the graph topology so that the condition of Eq. (29) implies collision free paths).

The results of this subsection and the previous subsection are summarized in the following tube calculation algorithm. This algorithm is used by $\forall i\in\mathbb{V}$ to calculate a tube around the delayed trajectory of neighbor j:

Algorithm 2: Tube Calculation Algorithm

Given $\boldsymbol{\mu}^j\ge 0$ and d, where $(i,j)\in\mathbb{E}$ and d is the communication delay between i and j:

```
1- Calculate $\beta^j(d,\boldsymbol{\mu}^j)$ from Eq. (26).
2- Calculate $\boldsymbol{\alpha}^{j,i}(t)$ from Eq. (17).
3- Calculate $\mathbb{H}_k^{j,i}$ from Eq. (16).
```

The vehicle i can also use *Algorithm 2*, to calculate the tube around trajectory of itself by setting d=1.

D. Conflict Resolution DMPC Algorithm

Initial Feasibility Assumption: assume at time t_0 for all $i\in\mathbb{V}$ there exist $\boldsymbol{\mu}_0^i$ so that a feasible solution satisfying collision avoidance condition Eq. (29) exists.

The following algorithm is presented for the on-line implementation of the DMPC with the proposed conflict resolution scheme, $i\in\mathbb{V}$:

Algorithm 3: DMPC with Conflict Resolution

```
1- Let $k$=0, and GOTO step 4.
2- Receive the trajectory $\boldsymbol{x}_{k-d}^j(\cdot)$ ; $(i,j)\in\mathbb{E}$ (with appropriate $d$).
3- Calculate admissible $\mathbb{U}^i$ :
    a. Choose $\boldsymbol{\mu}^i=\boldsymbol{\mu}_0^i$ .
    b. For $\forall j\in\mathbb{V}$ where $(i,j)\in\mathbb{E}$
        I. Choose $\boldsymbol{\mu}^j=\boldsymbol{\mu}_0^j$ and calculate $\mathbb{H}_k^{j,i}$ for $(i,j)\in\mathbb{E}$ by running Algorithm 2.
        II. Calculate $\mathbb{H}_k^{i,i}$ by running Algorithm 2.
        III. If $\mathbb{H}_k^{i,i}\cap\mathbb{H}_k^{j,i}\neq\varnothing$ then reduce $\boldsymbol{\mu}^i$ by 10% and goto step 3.b.II.
    End For
    c. Compute the admissible input set $\mathbb{U}^i$ from Eq. (30).
4- Measure $\boldsymbol{x}^i(t_k)$ and calculate $\Gamma^i(t_k)$ .
5- Solve $\mathscr{P}^i(t_k)$ and generate: $\boldsymbol{u}_k^i(\cdot)$ and $\boldsymbol{x}_k^i(\cdot)$ .
6- Send the trajectory $\boldsymbol{x}_k^i(\cdot)$ to the neighboring vehicles.
7- Apply the control action for individual vehicle $i$ over the time interval $[t_k,t_{k+1}]$ .
8- $k$=$k$+1. Goto step 2.
```

The only difference between this algorithm and Algorithm 1 is step 3 which provides collision avoidance. In step 3.b. III, for checking the condition $\mathbb{H}_k^{i,i} \cap \mathbb{H}_k^{j,i} = \varnothing$, it is required that the distance between trajectories $\boldsymbol{\xi}_{k-d}^{j}(\cdot)$ and $\boldsymbol{\xi}_{k-1}^{i}(\cdot)$ at any time $t \in [t_k, t_{k-d} + T]$ be larger than $\boldsymbol{\alpha}^{j,i} + \boldsymbol{\alpha}^{i,i}$ (radius of tubes); this is a practical method used in the simulations to check whether tubes intersect or not.

V. Simulation Results

Collision avoidance of a fleet of unmanned vehicles in the 2D plane is considered, where $\boldsymbol{\xi} \in \mathbb{R}^2$, $\boldsymbol{\upsilon} \in \mathbb{R}^2$, $\boldsymbol{u} \in \mathbb{R}^2$ and $\dot{\boldsymbol{\xi}} = \boldsymbol{\upsilon}$, $\dot{\boldsymbol{\upsilon}} = -\boldsymbol{\upsilon} + [1,\ 0.4]\boldsymbol{u}$.

A. Off-line Calculations

The presented dynamics for vehicles is marginally stable; hence, to take advantage of the Controllability *Gramain* for stable systems an inner loop feedback controller is first designed to stabilize the dynamics by placing the poles at [-0.0513 -0.0513 -1.9487 -1.9487] by feedback controller: *K*= [0.1, 0, 1.0, 0; 0, 0.25, 0.0, 2.5]. Then the solution of the LMI of Eq. (13) is $M = W^{-1}$ = diag(12.0, 120.0, 75.0, 750.0). To verify this ellipsoid bounds the trajectories, the ellipsoid along with 100 random simulations for four different β (see Eq. (11)) are depicted in Figure 6. As seen the ellipsoid is a non-conservative bound on the reachable sets.

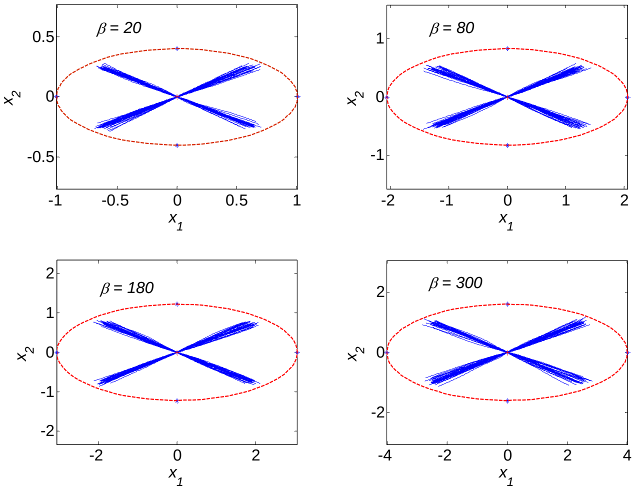

Figure 6. *The reachable sets corresponding to positions for different maneuverability*

B. Simulation of On-line Scenarios

For the simulations, CORA (Control Optimization and Resource Allocation) library developed in CIS (Control and Information Systems) laboratory of Concordia University is used. CORA is an object oriented library based on Microsoft C++ environment and uses the SNOPT optimization package [29] to solve the MPC and other optimization problems. For trajectory generation, each state trajectory is modeled by a B-Spline basis function; hence, for inter-vehicle communication, only the Spline control points are communicated rather than all points of the trajectory, in order to reduce the communication load.

The prediction horizon and execution horizon (sampling time) of DMPC is set to *T*=1 sec and $\delta = 0.2$ sec, respectively. Also, it is assumed that the communicated messages are subject to a delay of *d*=3 steps (or $\tau = d.\delta = 0.6\,\text{sec}$). Further the initial maneuverability parameter is set to $\boldsymbol{\mu}_0^i = 40\,I$.

To evaluate the proposed collision avoidance algorithm, a scenario (see Figure 7) is considered where the vehicles start from some random positions (circles) and they have to visit their assigned targets (cross product sign). The target positions are assigned so that the vehicles potentially collide. This scenario can imitate the air traffic control scenarios near the airports.

Figure 7-Left shows the case where no collision avoidance constraint is used; in fact, the vehicles do not cooperate and hence two of the vehicles collide according to the corresponding distance time history in Figure 8-Left. For this scenario, the Figure 7-Right shows the effect of the proposed collision avoidance algorithm. As seen from the corresponding distance profile of Figure 8-Right the proposed algorithm can predict the possible collision and avoid collisions.

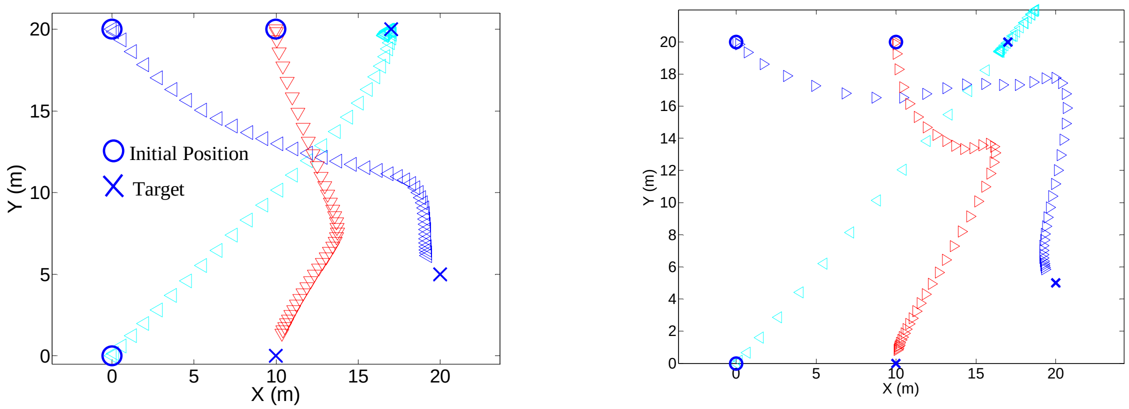

Figure 7. *The snapshot of the trajectories for 3 vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm*

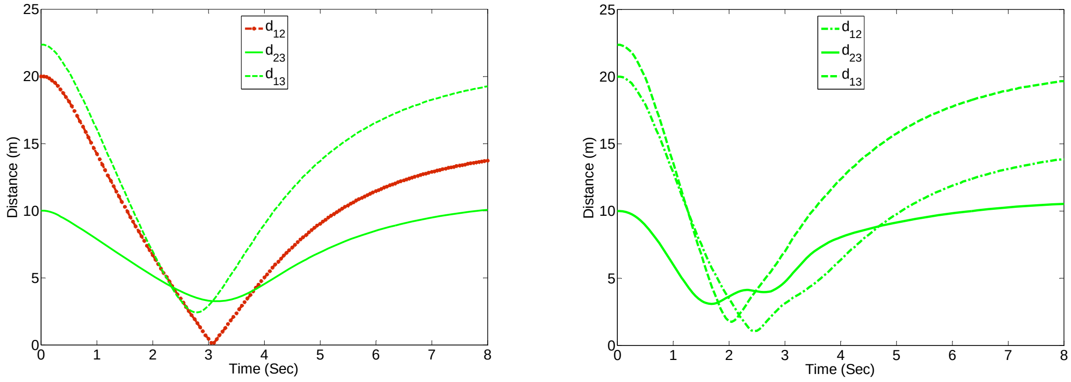

Figure 8. *The distance between each pair of vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm*

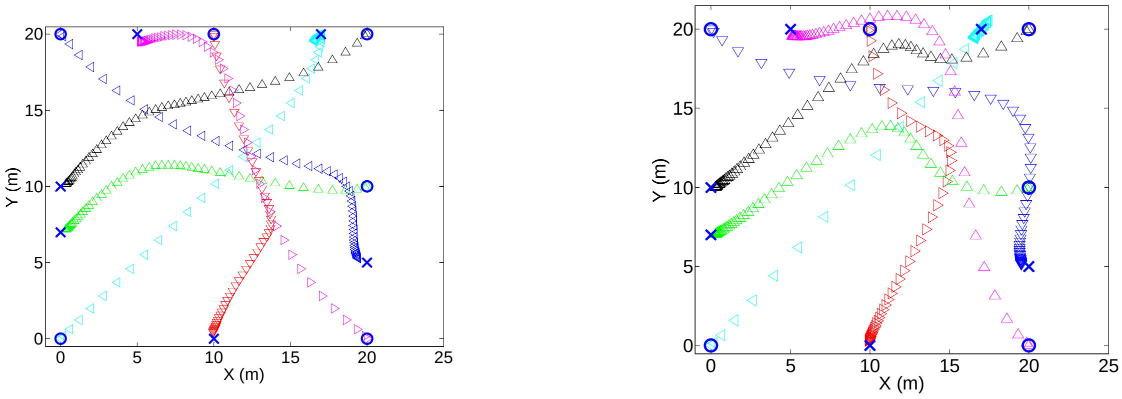

Figure 9. *The snapshot of the trajectories for 6 vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm*

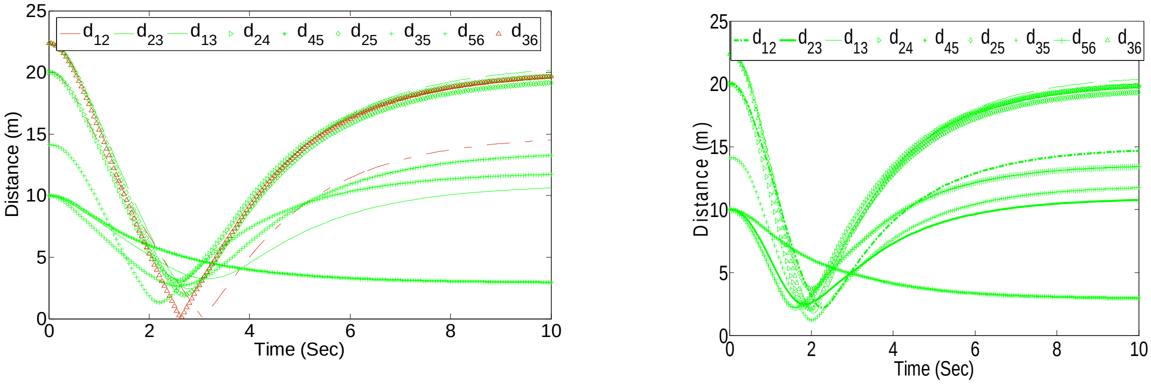

Figure 10. *The distance between each pair of vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm*

To consider a more complex scenario, the simulations are repeated for 6 vehicles and the simulation results are depicted in Figure 9 and Figure 10. For this case, if the collision avoidance algorithm is not used two pair of vehicles will collide according to Figure 10-Left. The snapshot of the trajectories for the case when the proposed conflict resolution algorithm is used, is depicted in Figure 9-Right. The corresponding distance profile, shown in Figure 10-Right, shows that the proposed algorithm can resolve the potential conflicts.

VI. Conclusions and Future Work

A new decentralized model predictive control (DMPC) approach is developed which provides the safety of cooperative vehicles against possible collisions in presence of large inter-vehicle communication delays. The method is based on calculation of some tube shaped sets around the delayed trajectory of neighboring vehicles by connecting the reachable sets over the prediction horizon, and then setting the maneuverability so that the tubes of neighboring vehicles do not intersect. A LMI approach is used to compute the reachable set and tubes for vehicles with linear dynamics, which does not impose any online computation load. The simulation results demonstrate that the proposed methodology can be used for conflict resolution problems. Future research involves extending the algorithm for vehicles with a nonlinear dynamics in the presence of disturbances and model uncertainties. Also, the feasibility of the proposed algorithm and tubes should be addressed.

Acknowledgement

This work was supported in part by National Science and Engineering Research Council of Canada (NSERC).

References

[1]Krozel, J. and Peters, M., "A Decentralized Control Strategy for Distributed Air/Ground Traffic Separation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, Aug. 14-17, 2000.

[2]Krozel, J. and M. Peters, "Decentralized Control Techniques for Distributed Air/Ground Traffic Separation," NASA Report: TR 99RTO36-03, NASA Ames Research Center, Moffett Field, CA 94035.

[3]Pallottino, L., Scordio, V. G., Frazzoli, E. and Bicchi, A., "Decentralized Cooperative Policy for Conflict Resolution in Multi-Vehicle Systems," *IEEE Transactions on Robotics*, Vol. 23, No. 6, 2007, pp. 1170-1183.

[4]Baskar, L. D., De Schutter, B. and Hellendoorn, H., "Decentralized Traffic Control and Management with Intelligent Vehicles," *the 9th Trail Congress*, Rotterdam, The Netherlands, Nov. 2006.

[5]Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. and Wang, Y., "Review of Road Traffic Control Strategies," *Proceedings of the IEEE*, Vol. 91, No. 12, Dec. 2003.

[6]Pallottino, L., Giovanni Scordio, V. and Bicchi, A., "Decentralized Cooperative Conflict Resolution among Multiple Autonomous Mobile Agents," *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 14-17, 2004, pp. 4758-4763.

[7]Yoon, Y., Kim, H., Shin, J., Choe, T. and Park, Y., "Communication in Distributed Model Predictive Collision Avoidance," *International Conf. on Robot Communication and Coordination*, Athens, Greece, 2007.

[8]Richards, A. and How, J. "Decentralized Model Predictive Control of Cooperating UAVs," *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 14-17, 2004.

[9]Keviczky, T., Vanek, B., Borrelli, F. and Balas, G. J., "Hybrid Decentralized Receding Horizon Control of Vehicle Formations," *Proceedings of the American Control Conference*, Minneapolis, Minnesota, USA, June 14-16, 2006, pp. 3358-3363.

[10]Swaroop D. and Hedrick, J. K., "String Stability of Interconnected Systems," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, 1996, pp. 349-357.

[11]Varaiya, P., “Smart Cars on Smart Roads: Problems of Control,” *IEEE Transactions on Automatic Control*, Vol. 38, No. 2, 1993, pp. 195-207.

[12]Tomlin, C.J., Pappas, G.J. and Sastry, S. S., “Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems,” *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509-521.

[13]Mastellone, S., Stipanovic, D. M., Graunke, C. R., Intlekofer K. A. and Spong, M. W., “Formation Control and Collision Avoidance for Multi-agent Non-holonomic Systems: Theory and Experiments,” *The International Journal of Robotics Research*, Vol. 27, 2008, pp. 107-126.

[14]Mejia J. S. and Stipanovic, D. M., “Computational Receding Horizon Approach to Safe Trajectory Tracking,” *Integrated Computer-Aided Engineering*, Vol. 15, 2008, pp. 149-161.

[15]Keviczky, T., Borrelli, F., Fregene, K., Godbole, D. and Balas, G. J., “Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations,” *IEEE Transactions on Control Systems Technology*, Vol. 16, No.1, 2008.

[16]Zhang, X., Xu, R., Kwan, C., Haynes, L., Yang, Y. and Polycarpou, M., “Fault Tolerant Formation Flight Control of UAVs,” *International Journal of Vehicle Autonomous Systems*, Vol. 2, No.3/4, 2004, pp. 217-235.

[17]Innocenti, M., and Pollini, L., “Management of Communication Failures in Formation Flight,” *Journal of Aerospace Computing, Information, and Communication,* Vol. 1, Jan. 2004, pp. 19-35.

[18]Cheng, L., Wang, Y. and Zhu, Q., “A Communication-Based Multi-Robot Rigid Formation Control System: Design and Analysis,” *Int. Journal of Modeling, Identification and Control*, Vol. 1, No. 1, 2006.

[19]Izadi, H., Gordon, B. W. and Rabbath, C. A., “Decentralized Receding Horizon Control using Communication Bandwidth Allocation,” the *47th IEEE Conf. on Decision and Control*, Cancun, Mexico, Dec. 9-11, 2008.

[20]Izadi, H., Gordon, B. W. and Zhang,Y., "Reconfigurable Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Delay," *AIAA Journal of Guidance, Control and Dynamics,* Vol. 32, No.6, 2009, pp 1959-1965.

[21]Izadi, H., Gordon, B. W. and Zhang,Y., “Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Failure”, the *American Control Conference* , St Louis, USA, June 10-12, 2009.

[22]Mayne, D. Q., Rawlings, J. B., Rao C. V. and Scokaert, P. O., “Constrained Model Predictive Control: Stability and Optimality”, *Automatica*, Vol. 36, 2000, pp. 789-814.

[23]Dunbar, W. B. and Murray, R. M. “Receding Horizon Control of Multi-vehicle Formations: A Distributed Implementation,” the *43rd IEEE Conf. on Decision and Control*, 2004.

[24]Khalil, H. K., *Nonlinear Systems*, 2nd ed. New Jersey: Prentice Hall Inc., 1996, pp. 80-81.

[25]Alvarado, I., Limon, D., Alamo, T. and Camacho, E. F., “Output Feedback Robust Tube Based MPC for Tracking of Piece-Wise Constant References,” *46th IEEE Conf. on Decision and Control,* Dec. 12-14 2007, pp. 2175-2180.

[26]Langson, W., Chryssochoos, L., Rakovic, S. V., and Mayne, D. Q., “Robust Model Predictive Control Using Tubes”, *Automatica,* Vol. 40, No. 1, Jan. 2004, pp. 125-133.

[27]Trodden, P. and Richards, A., “Robust Distributed Model Predictive Control Using Tubes,” *American Control Conference*, June 14-16 2006, pp. 2034-2039.

[28]Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of Studies in Applied Mathematics Society for Industrial and Applied Mathematics (SIAM), 1994.

[29]Gill, P. E., Murray, W. and Saunders, M. A., *User’s Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112, USA, May 30, 2006.

[30]Mitchell, M., Bayen, A. M. and Tomlin, C. J., “A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games,” *IEEE Transactions on Automatic Control*, Vol. 50, No. 7, 2005, pp. 947-957.

[31]Hwang, I., Stipanovic, D. and Tomlin, C., “Application of Polytopic Approximations of Reachable Sets to Linear Dynamic Games and a Class of Nonlinear Systems,” *In the Proceedings of the American Control Conference,* Denver, June 2003.