2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Failure

Hojjat A. Izadi, *Student Member, IEEE*, Brandon W. Gordon, and Youmin Zhang, *Senior Member, IEEE*

***Abstract*—In this paper, the decentralized receding horizon control (DRHC) of multiple cooperative vehicles with the possibility of communication failure is investigated. The neighboring vehicles exchange their computed trajectories at each sample time to maintain cooperation objectives. It is assumed that the communication failure is partial in nature, which in turn leads to large communication delays. A new reconfigurable DRHC approach is developed that guarantees the safety of the entire fleet in the presence of inter-vehicle communication failures. The concept of tube RHC is introduced to guarantee the safety of the fleet against collisions during faulty conditions. In this approach, a tube shaped trajectory set is used instead of a trajectory for the neighboring vehicles experiencing the communication failure.**

I. Introduction

Fault tolerant controllers and algorithms have recently become an active area of new research in the field of cooperative control of multiple vehicles [1-5]. Two general classes of faults may be considered: the low level faults such as actuator or sensor faults which directly affect the inner loop controller and the high level faults which affect the teaming objectives such as communication or GPS failures. In this paper, we are interested in the second class of failures. It is desired to study the effect of high level faults on the teaming behavior and designing new reconfigurable fault tolerant controllers that can handle the fault and provide safe behavior and satisfactory performance. The vehicles use a Decentralized Receding Horizon Control (DRHC) scheme for both path planning and inner loop control.

Only a few research works address the fault diagnosis subject for multiple vehicles. In a very close work [1], it is desired to manage the communication failure in formation control of multiple vehicles. Two main fault categories are considered in [1]: 1) TX and/or RX do not work; 2) One aircraft is lost. When the TX of an aircraft in the formation doesn't work, all the followers of faulty aircraft must put another eligible aircraft as reference. Also, the faulty aircraft is moved to a position in the formation where there is no need to transmit the information (e.g. a leaf in the tree). When the RX of an aircraft is faulty, it either uses the trajectory of a virtual leader or leaves the formation. To keep all aircrafts informed about all operational aircrafts in the group, a backup broadcasting communication channel is used. Whenever an aircraft is lost the formation must be reconfigured to a predetermined allowable formation.

In another related work [2], two high level faults for formation control are considered: 1) GPS sensor failure and 2) wireless communication packet losses. To detect the GPS sensor failure a state/output observer is used which monitors the behavior of vehicle. If the difference between the output of observer and GPS data is larger than some threshold, then a GPS fault is identified. Once the fault is identified, the distance and position information given by other vehicles (at least three vehicles) in fault-free condition is used to estimate the position of faulty vehicle using the distance formula. However, it is not mentioned how the information on distance is provided in faulty situation. To detect the communication packet loss/delay fault in [2] the packets are numbered sequentially and the number of packet is also transmitted. A mismatch between the expected packet number and the received packet number implies the occurrence of packet loss. Once, the packet loss/delay is occurred, depending on whether the lost information is from leader or neighbor, the previous available trajectory of neighbor is extrapolated to predict the future reference trajectory.

In [3] two main faults are defined for a cooperative leader-follower formation control of multi-robots: 1) Communication channel failure (High Level Fault) and 2) Robot machine failure (Low Level Fault). To diagnose these faults, the faulty robot should leave the formation for formation safety and the other team members have to reconstruct the formation. In all above works no discussion on guaranteed collision avoidance is made.

In this paper a faulty condition is defined and then the fault detection and control reconfiguration method is presented, it involves the failure of a high performance communication channel which leads to large communication delays; and the proposed reconfigurable DRHC architecture is supposed to handle the large communication delays and maintain a safe formation. The safety guarantee method for the faulty condition is developed based on the concept of tube RHC, *i.e.* using a tube instead of a trajectory for neighboring vehicles experiencing the communication failure.

[1] Manuscript received Sep. 15, 2008. This work was supported in part by National Science and Engineering Research Council of Canada (NSERC).

H. Izadi, B. W. Gordon and Y. Zhang are with Department of Mechanical and Industrial Engineering, Concordia University, 1515 St. Catherine, Montreal, Quebec, H3G 1M8, Canada (e-mails: ha_izadi@encs.concordia.ca, bwgordon@encs.concordia.ca, ymzhang@encs.concordia.ca).

II. Decentralized Receding Horizon Control (DRHC) Formulation

Consider a team of N_v vehicles with uncoupled dynamics. The measurement sensors of each vehicle measure only its own states. The communication channel is used to gather the information from the neighbors and communicate with human operators. Using a computation resource, each vehicle solves an optimization DRHC problem at each sampling time based on its instant states (from sensors) and the trajectory of its neighboring vehicles (from communication channel). Moreover, each vehicle has the dynamical model of its neighboring vehicles available to predict their trajectory when required. In this paper, it is assumed that there is no measurement sensor error, no model uncertainty, no communication noise and a perfect optimization is performed. However, the main problem arises from possible failure in the communication channels.

A. Interaction and Information Exchange Graphs

The interaction between cooperative vehicles is usually represented by an "*interaction graph*" including two main elements: nodes and arcs. The nodes represent the vehicles and an arc between two nodes denotes a coupling term in the objectives and/or in the constraints associated to the nodes. Also, it is assumed that the *information exchange graph* is fixed and that each vehicle can communicate information with only a subset of the other vehicles in the team. Furthermore, it is assumed that only the vehicles which have physical interactions such as collision avoidance, will exchange information.

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 arc (i, j) connecting the i^{th} and j^{th} node is present, it means that the i^{th} and j^{th} vehicles have a coupling term in their cost function and/or in their constraints (interaction), and communicate with each other. This relationship is termed as a *neighborhood* for the i^{th} and j^{th} vehicles. This leads to the interaction graph as follows:

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

where V is the set of nodes (vehicles) and $E \subseteq V \times V$ the set of arcs (i,j), with $i, j \in V$. The interaction graph is indirect i.e. $(i,j) \in E$ implies $(j,i) \in E$ even though it does not appear in E. Also, let N_n^i denotes the number of neighbors of vehicle i.

B. DRHC Notation and Terminology

With Receding Horizon Control (RHC) - also known as model predictive control (MPC) - a cost function is optimized over a 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 [6] for a comprehensive review of RHC schemes.

The *execution horizon* δ is assumed to be equal to the *communication period*; thus this provides synchronization between the communication rate and the sampling rate of RHC. Then, the discrete timing is shown by t_k where $t_{k+1} = t_k + \delta$ (or $t_k = k.\delta$) and $t_0 = 0$.

The possible state vectors are introduced as follows:

- $x^i(t)$: the actual state vector of i^{th} vehicle at time *t.*
- $x_{t_k}^{j,i}(t)$: the state vector of j^{th} vehicle at time *t*, computed by i^{th} vehicle at time step t_k.

Then, the state of vehicle i calculated by itself at time t_k is represented by $x_{t_k}^{i,i}(t)$. Also the sequence of these states over the prediction horizon is called the state trajectory of vehicle i calculated by itself and is represented by $\hat{x}^i(t_k)$:

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

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

$$\hat{\bar{x}}^i(t_k) = [\ldots, \hat{x}^j(t_k), \ldots]^T \qquad ; j \in V,\ (i,j) \in E, \tag{3}$$

The same notation will be used for input vector.

C. Fault-Free DRHC Formulation

In some previous works [7, 8] a DRHC scheme is used where the vehicles need to exchange only their instant states. However, for the scheme presented in this paper the trajectories are exchanged instead of states to reduce the computation time. Figure 1 shows the inter-vehicle communication between two neighboring vehicles and the information exchanged at time t_k for fault-free condition. As seen the information exchange is not subject to communication delay.

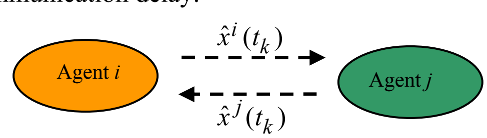

Figure 1: The inter-vehicle communications between two neighbors in fault-free condition

Equation (3) represents the information set that vehicle i receives from its neighbors. However, the vehicle i needs its own instant states as well to solve the optimization problem as we will see later; then the *information vector* of i^{th} vehicle for the case of fault-free DRHC is introduced as follows:

$$\tilde{x}^i(t_k) = [x^i(t_k), \hat{\bar{x}}^i(t_k)]^T \tag{4}$$

vector $\tilde{x}^i(t_k)$, the *information vector,* collects the state vector of i^{th} vehicle and the concatenated state trajectory of neighbors $\hat{\bar{x}}^i(t_k)$. The former is provided through on-board sensors and the latter is provided through communication.

For the particular case of formation control, the fault-free decentralized cost function for the i^{th} vehicle in the team at time t_k is defined as follows:

$$J^i(\tilde{x}^i(t_k),\hat{u}^i(t_k)) = \int_{t_k}^{t_k+T}\left(\left\|x_{t_k}^{i,i}(t)\right\|_Q^2 + \left\|u_{t_k}^{i,i}(t)\right\|_R^2\right)dt + \left\|x_{t_k}^{i,i}(t_k+T)\right\|_P^2 + \sum_{j|(i,j)\in E}\int_{t_k}^{t_k+T}\left\|x_{t_k}^{i,i}(t) - x_{t_k}^{j,j}(t) - r^{i,j}\right\|_S^2 dt \quad (5)$$

where $\|x\|_Q^2 = x^T Qx$ and *P, Q, R* and *S* are positive definite and symmetric matrices. Also, $r^{i,j}$ is the vector of desired position between agent *i* and *j*. Such approach is used extensively in the literature [9, 10].

1. Fault-Free DRHC Problem

Assume the following represents the dynamics for a general class of homogeneous vehicles:

$$\dot{x}(t) = f(x(t),u(t)); \qquad x(t_0) = x_0 \quad (6)$$

Then, the Fault-Free DRHC problem $P^i(t_k)$ is defined for the i^{th} vehicle at time t_k as follows:

Problem 1: Fault-Free DRHC Problem $P^i(t_k)$ ($i \in V$):

$$\underset{\hat{u}^i(t_k),\hat{x}^i(t_k)}{\text{Min}} \quad J^i(\tilde{x}^i(t_k),\hat{u}^i(t_k)) \quad (7)$$

Subject to:

$$\begin{aligned}&\dot{x}_{t_k}^{i,i}(t) = f(x_{t_k}^{i,i}(t),u_{t_k}^{i,i}(t));\\ &x_{t_k}^{i,i}(t_k) = x^i(t_k);\ t\in[t_k,t_k+T]\end{aligned} \quad (8a)$$

$$x_{t_k}^{i,i}(t)\in \mathrm{X}^i,\ u_{t_k}^{i,i}(t)\in \mathrm{U}^i;\ t\in[t_k,t_k+T] \quad (8b)$$

$$x_{t_k}^{i,i}(t_k+T)\in \mathrm{X}_f^i \quad (8c)$$

In Eq. (7), J^i comes from Eq. (5), vectors X^i, U^i and X_f^i denote the set of admissible states, inputs and final states (terminal region), respectively, for the i^{th} vehicle.

2. Fault-Free DRHC Algorithm

Each vehicle *i* at each sampling time solves the decentralized problem $P^i(t_k)$ using its own state information and those information from its neighbors; the output of this optimization problem is the input and state trajectory of itself on the interval $[t_k,t_k+T]$. After generating these trajectories the DRHC controller applies only the first portion of its own trajectory during $[t_k,t_{k+1}]$ to the vehicle, and sends the trajectory generated for itself to each neighbor for collision avoidance purposes. The following algorithm is presented for the on-line implementation of the proposed fault-free DRHC. The algorithm is formulated for the i^{th} vehicle; in fact, all vehicles run this algorithm during the mission simultaneously:

Algorithm 1: Fault-Free DRHC

1- Let *k=0*.

2- Receive the trajectory $\hat{x}^j(t_k)$; $(i,j)\in E$ from neighbors.

3- Solve $P^i(t_k)$ and generate the control action $\hat{u}^i(t_k)$ for $[t_k,t_k+T]$.

4- Send the trajectory $\hat{x}^i(t_k)$ to the neighboring vehicles.

5- Execute the control action for individual vehicle *i* over the time interval $[t_k,t_{k+1}]$.

$$u^i(t) = u_{t_k}^{i,i}(t);\ t\in[t_k,t_{k+1}] \quad (9)$$

6- *k=k+1*. Goto step 2.

This algorithm is repeated until the assigned targets (*e.g.* origin) are reached. The targets are assumed to be known and assigned to each agent *a priori*.

Remark 1: to alleviate the formulation complexity the above algorithm and cost function (5) assumes zero computation time and that the generated trajectories from neighbors are available instantly. However, a one step delay has to be imposed as the exchanged trajectories are subject to at least one step delay,

III. Fault Tolerant DRHC

This section deals with fault detection and isolation (FDI) and fault tolerant controller (FTC) reconfiguration design. The safety guarantee in faulty conditions is also discussed. It is assumed each vehicle is equipped with: 1) high performance communication channel and 2) low performance communication channel as backup. In the fault-free condition the high performance communication channel is used which leads to *small communication delays,* typically smaller than sampling time. In the faulty condition the low performance communication channel is used as backup which leads to *large communication delays*. The delay in faulty condition is applied to both received and sent information from/to faulty vehicle. Then the following faulty condition is defined:

Faulty Condition: The high performance communication channel fails.

A. Fault Detection and Isolation (FDI)

In the normal condition all the vehicles receive/send the information from/to their neighbors with no delay (or a small delay as less than sampling time). If the communication delay of received messages is larger than some threshold, which is the limit between *small communication delays* and *large communication delays*, the occurrence of high performance communication failure is concluded (See

Figure 2 and compare with Figure 1). Both faulty vehicle and its neighbors can use this sign to detect the fault. However, this idea needs to be more expanded to find which vehicle is faulty in the team.

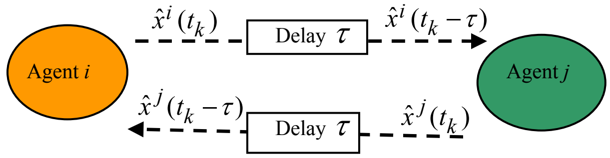

Figure 2: The inter-vehicle communication between two faulty vehicles

Assume at some time vehicle i does not hear from its neighbors; how does this vehicle determine that the break in messages is due to failure in its own communication channel or that of its neighbors? The approach presented here needs each vehicle in the group to have at least two neighbors i.e. $N_n^i \geq 2\ ;\ i \in V$. Hence, if the vehicle i does not hear after a reasonable time (T_c) from all its neighbors it concludes that its high performance communication channel is faulty. Accordingly, once the vehicle i hears from some neighbors without delay and does not hear from the others it concludes its communication channel is not faulty and it is the communication channel of one (or more) of its neighbor that is faulty, this FDI algorithm is summarized in.

B. Fault Tolerant Controller (FTC)

Once the fault is detected and the faulty vehicle is identified in the team, all vehicles involving the fault, construct the set of faulty neighbors, which is denoted by V_F^i, the set of faulty neighbors of vehicle i. The vehicles which have a faulty neighbor assign the faulty neighbor to this set, and the faulty vehicle assign all of its neighbors to this set. Then the faulty vehicle switches to the backup low performance communication channel. This will cause the neighboring vehicles to receive the messages from faulty vehicle with a *large communication delay*. Then the DRHC controller of neighbors of faulty vehicle and faulty vehicle have to be reconfigured to account for *large communication delays i.e.* to use delayed information instead of delay-free information; the next subsection presents the DRHC formulation for faulty condition.

1. Reconfigurable DRHC Formulation

Assume that the information communicated among the vehicles in faulty condition is subject to time-delay τ ; Figure 2 illustrates how the vehicles receive the information with a time delay from their neighbors, it is assumed that $\tau \geq \delta$ in faulty condition. Further, assume $(d-1)\delta \leq \tau \leq d\delta$ where $d \in N$, see Figure 3.

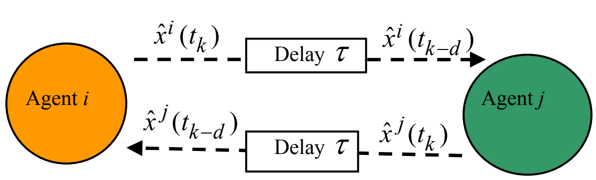

Figure 3: Synchronization of communication delay with RHC timing

Hence, the information vectors are updated as follows (compare with (4)):

$$\tilde{x}^i(t_k) = [x^i(t_k), \hat{x}^i(t_{k-d})]^T \tag{10}$$

vector $\tilde{x}^i(t_k)$ 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, *i.e.* $\hat{x}^i(t_{k-d})$. Consequently, the decentralized cost function of each vehicle i includes two parts: the first part is associated to the cost of individual vehicle i and the second part is associated to the neighboring vehicles and therefore relies on the delayed information. Hence, the cost function for faulty condition (large communication delay) is presented as following for the i^{th} vehicle in the team at time t_k :

$$\begin{aligned}
&J_F^i(\tilde{x}^i(t_k), \hat{u}^i(t_k)) = \\
&\int_{t_k}^{t_k+T} \left(\left\| x_{t_k}^{i,i}(t) \right\|_Q^2 + \left\| u_{t_k}^{i,i}(t) \right\|_R^2 \right) dt + \left\| x_{t_k}^{i,i}(t_k+T) \right\|_P^2 + \\
&\sum_{\substack{j|(i,j)\in E \\ j \notin V_F^i}} \left[\int_{t_k}^{t_k+T} \left\| x_{t_k}^{i,i}(t) - x_{t_k}^{j,j}(t) - r^{i,j} \right\|_S^2 dt \right] + \\
&\sum_{\substack{j|(i,j)\in E \\ j \in V_F^i}} \left[\int_{t_k}^{t_{k-d}+T} \left\| x_{t_k}^{i,i}(t) - x_{t_{k-d}}^{j,j}(t) - r^{i,j} \right\|_S^2 dt \right. \\
&\qquad \left. + \int_{t_{k-d}+T}^{t_k+T} \left\| x_{t_k}^{i,i}(t) - x_{t_k}^{j,i}(t) - r^{i,j} \right\|_S^2 dt \right] + \\
&\sum_{\substack{j|(i,j)\in E \\ j \in V_F^i}} \left[\int_{t_{k-d}+T}^{t_k+T} \left(\left\| x_{t_k}^{j,i}(t) \right\|_Q^2 + \left\| u_{t_k}^{j,i}(t) \right\|_R^2 \right) dt + \left\| x_{t_k}^{j,i}(t_k+T) \right\|_P^2 \right]
\end{aligned} \tag{11}$$

The subscript "F" stands for Faulty condition.

2. Safety Guarantee Using Tube RHC

A communication fault leads to large communication delays, which implies the lack of updated information on the trajectory of neighboring vehicles; this lack of information can make the formation 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 limited using the available, although delayed, information. The main idea is that whenever a communication failure occurs the faulty vehicle imposes an input constraint in its optimization problem: *i.e.* at any time instant the input trajectories do not deviate too far from the previous one. Consequently, 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 RHC (or tube MPC) is used normally for uncertain systems to calculate a robust bound on the states [11-13]. Using the tube instead of a trajectory in the formation flight leads also to the concept of tight (fault free) and loose (faulty) formations; because in faulty conditions the position of faulty vehicle is assumed to be a closed set (like a sphere), rather than a single point.

The following problem represents a method for calculating the reachable set for linear systems:

Problem 2: Consider the following describes the dynamics of each vehicle:

$$\dot{x} = Ax + Bu \;\; ; \;\; x(t_0) = x_0 \tag{12}$$

Also assume the control input is bounded as follows:

$$-\mu \le u_{t_k}(t) - u_{t_{k-1}}(t) \le \mu \;\; ; t \in [t_k, t_{k-1}+T] \;\; \& \;\; k \in N \tag{13}$$

where μ is a vector with appropriate length containing the bound on the control inputs. Then calculate after d steps, how far can each vehicle get? In other words, if at time t_k vehicle i receives the information from neighbor j with d steps time delay *i.e.* $\hat{x}^j(t_{k-d})$, then calculate the reachable set of j at time t_k. This reachable set is denoted by $\hat{\mathrm{X}}^j(t)$.

Solution: The updated trajectory of vehicle j is approximated from delayed trajectory, by vehicle i as follows:

$$\hat{x}^j(t_k) = \hat{x}^j(t_{k-d}) + \delta\hat{x}^j(t_k) \tag{14}$$

where, $\delta\hat{x}^j(t_k)$ is due to any variation in control input during $[t_{k-d}, t_k]$ and is calculated as follows: using the state transition method the solution of (12) is given as:

$$x(t) = \varphi(t,t_0)x(t_0) + \int_{t_0}^{t} \varphi(t,s)Bu(s)ds \tag{15}$$

Then assume a small perturbation in input as follows: $u = u + \delta u$, hence:

$$\begin{aligned} x + \delta x &= \varphi(t,t_0)x(t_0) + \int_{t_0}^{t} \varphi(t,s).B.(u(s)+\delta u(s)).ds \Rightarrow \\ \delta x &= \int_{t_0}^{t} \varphi(t,s).B.\delta u(s).ds \end{aligned} \tag{16}$$

To find $\delta u(s)$ after d step delay the input constraint (13) is written sequentially to end up with the following:

$$-d\mu \le \delta u_{t_k}(t) \le d\mu; \quad t \in [t_k, t_{k-d}+T] \tag{17}$$

By substituting all possible values for $\delta u(s)$ from (17) into (16) all possible δx can be found, and then the set of reachable states $\hat{\mathrm{X}}^j(t)$ can be calculated using (14).

For a safe trajectory, in the 3rd term of cost function (11), $x^{j,j}_{t_{k-d}}(t)$ must be chosen from tube $\hat{\mathrm{X}}^j(t)$ and not the trajectory $x^{j,j}_{t_{k-d}}(t)$. Using a tube instead of trajectory will raise this question that which point of tube can be used for $x^{j,j}$ in the cost function (11). Different approaches can be used to choose one of the points as the most important point, *i.e.* the point from the reachable set $\hat{\mathrm{X}}^j(t)$ that puts the system the most in jeopardy must be chosen and handled. For instance, to satisfy a collision avoidance constraint the nearest point of the tube is the most important point.

3. Reconfigurable DRHC Problem Formulation

The reconfigurable DRHC problem $P^i_F(t_k)$ for faulty condition is defined for any i^{th} vehicle which involves in the fault, either itself is faulty or its neighbors are faulty (at time t_k):

Problem 3: Reconfigurable DRHC Problem $P^i_F(t_k)$ *:*

$$\underset{\hat{u}^i(t_k),\tilde{x}^i(t_k)}{\text{Min}} \quad J^i_F(\tilde{x}^i(t_k), \hat{u}^i(t_k)) \tag{18}$$

Subject to:

$$\begin{aligned} &\dot{x}^{i,i}_{t_k}(t) = f(x^{i,i}_{t_k}(t), u^{i,i}_{t_k}(t)); \\ &x^{i,i}_{t_k}(t_k) = x^i(t_k); \; t \in [t_k, t_k+T] \end{aligned} \tag{19a}$$

$$x^{i,i}_{t_k}(t) \in \mathrm{X}^i, \; u^{i,i}_{t_k}(t) \in \mathrm{U}^i; \; t \in [t_k, t_k+T] \tag{19b}$$

$$\begin{aligned} &\dot{x}^{j,i}_{t_k}(t) = f(x^{j,i}_{t_k}(t), u^{j,i}_{t_k}(t)); \\ &x^{j,i}_{t_k}(t_{k-d}+T) = x^{j,j}_{t_{k-d}}(t_{k-d}+T) \\ &t \in [t_{k-d}+T, t_k+T]; \qquad (i,j) \in E \;\&\; j \in V^i_F \end{aligned} \tag{19c}$$

$$\begin{aligned} &x^{j,i}_{t_k}(t) \in \mathrm{X}^j, \; u^{j,i}_{t_k}(t) \in \mathrm{U}^j; \\ &t \in [t_{k-d}+T, t_k+T]; \qquad (i,j) \in E \;\&\; j \in V^i_F \end{aligned} \tag{19d}$$

$$\begin{aligned} &x^{i,i}_{t_k}(t_k+T) \in \mathrm{X}^i_f \\ &x^{j,i}_{t_k}(t_k+T) \in \mathrm{X}^j_f \;\; ; \;\; (i,j) \in E \;\&\; j \in V^i_F \end{aligned} \tag{19e}$$

$$\left| u^{i,i}_{t_k}(t) - u^{i,i}_{t_{k-1}}(t) \right| \le \mu \;\; ; \;\; t \in \left[t_k, t_{k-1}+T \right] \tag{19f}$$

In Eq. (18), J^i_F is calculated from (11). Constraint (19f) is imposed for safety guarantee purposes (*Problem 2*).

Removing (19f) from problem $P^i_F(t_k)$ and setting $d=0$, the problem $P^i_F(t_k)$ reduces to $P^i(t_k)$, fault-free *Problem 1***.** Hence, this problem can be used even in the case of fault-free condition by all vehicles with appropriate consideration about constraint (19f); consequently in $P^i_F(t_k)$ it is perfectly valid to choose $i \in V$.

4. Reconfigurable DRHC Algorithm

The following algorithm is presented for the on-line implementation of the proposed reconfigurable DRHC problem $P_F^i(t_k)$. The algorithm is formulated for the i^{th} vehicle; in fact, all vehicles run this algorithm during the mission simultaneously:

Algorithm 2: Reconfigurable DRHC

1- Let $k=0$.
2- Receive the trajectory $\hat{x}^j(t_{k-d})$; $(i,j)\in E$ (with appropriate d).
3- Calculate the reachable set $\hat{\mathrm{X}}^j(t)$; $(i,j)\in E$ & $j\in V_F^i$ using (14).
4- Solve $P_F^i(t_k)$ and generate the control action $\hat{u}^i(t_k)$ for $[t_k, t_k+T]$.
5- Send the trajectory $\hat{x}^i(t_k)$ to the neighboring vehicles.
6- Execute the control action for individual vehicle i over the time interval $[t_k, t_{k+1}]$.
7- $k=k+1$. Goto step 2.

IV. Simulation Results

Formation of a fleet of unmanned vehicles with double integrator dynamics and velocity damping in the 2-dimensional plane is considered. For the simulations, the 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 [14] to solve the RHC and other optimization problems. As an example, for the triangular leaderless formation of 3 vehicles the distances between vehicles are depicted in Figure 4.

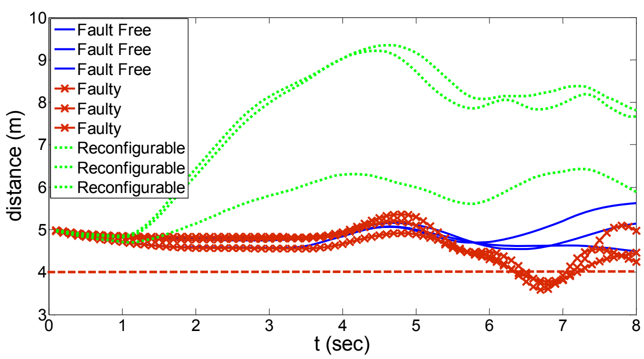

Figure 4: Distances between each pair of vehicles

The simulation was run for three cases: 1) fault free, 2) faulty with algorithm 1 and 3) faulty with reconfigurable algorithm 2. For the cases 2 and 3, after $t = 1sec$, a fault in high performance communication channel of vehicle 2 occurs which leads to d=5 time step delay in all information communicated to and from this vehicle. It is desired that vehicles keep a 5m distance and not less than 4m; as seen from Figure 4, in the case of fault, algorithm 1 violates this restriction that may lead to collision. However, the reconfigurable algorithm 2 offers a loose but safe formation (Figure 4) as the consequence of using tube RHC for safety.

V. Conclusions and Future Work

A new fault tolerant reconfigurable controller approach is developed which can address faults leading to large inter-vehicle communication delays. The fault detection and fault tolerant algorithms also perform in a decentralized fashion.

References

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

[2] 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 2004 Vol. 2, No.3/4, pp. 217 - 235.*

[3] L. Cheng, Y. Wang, and Q. Zhu, "A communication-based multi-robot rigid formation control system: design and analysis" *Int. J. Modeling, Identification and Control, Vol. 1, No. 1, 2006.*

[4] R. Tino, M. Henrique Terra, and Marcel Bergerman, "A fault tolerance framework for cooperative robotic manipulators", *Control Engineering Practice*, No 15, 2007.

[5] P. Molnar and J. Starke, "Communication fault tolerance in distributed robotic systems," in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds. New York: Springer-Verlag, 2000, pp. 99–108.

[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. Scokaert., "Constrained model predictive control: Stability and optimality", *Automatica*, 36, 2000, pp. 789-814.

[7] H. Izadi, B. W. Gordon, and C. A. Rabbath, "Decentralized Receding Horizon Control Using Communication Bandwidth Allocation", *the 47th IEEE Conf. on Decision and Control*, Cancun, Mexico, December 9-11, 2008.

[8] H. Izadi, B. W. Gordon, C. A. Rabbath, "Stability Improvement for Time Varying Decentralized Receding Horizon Control Systems", the *13th IEEE IFAC International Conf. on Methods and Models in Automation and Robotics (MMAR),* Szczecin, Poland, 27 - 30 August 2007.

[9] T. Keviczky , F. Borrelli and G. J. Balas, "Stability Analysis of Decentralized Receding Horizon Control for the decoupled systems ", *44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, December 12-15, 2005, pp.1689-1694.

[10] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: A distributed implementation", *43rd IEEE Conf. on Decision and Control*, 2004.

[11] I. Alvarado, D. Limon, T. Alamo, and E. F. Camacho, "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.

[12] W. Langson, I. Chryssochoos, S. V. Rakovic, and D. Q. Mayne, "Robust model predictive control using tubes", *Automatica,* Volume 40, Issue 1, January 2004, Pages 125-133.

[13] P. Trodden, and A. Richards, "Robust distributed model predictive control using tubes", *American Control Conference*, June 14-16 2006, pp 2034-2039.

[14] P. E. Gill, W. Murray, and M. A. Saunders, "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.