2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Dynamic Scheduling of Multiple RHC Systems with Coupling and Computational Delay

Ali Azimi, Brandon W. Gordon, and C. A. Rabbath, Senior *Member, IEEE*

***Abstract*—This paper develops a new algorithm for the dynamic scheduling of multiple receding horizon controllers running on a single processor. The subsystems are coupled and the formulation is adapted for decentralized RHC. The proposed formulation accounts for bounded model uncertainty, sensor noise and computation delay. A cost function appropriate for control of multiple vehicle systems is proposed and an upper bound on the cost as a function of the execution horizon is developed. The upper bound is optimized to obtain the execution horizon of all subsystems subject to the computation constraints. To study computational delay, two methods that applied on experimental apparatus, are used and compared theoretically. Based on this comparison, the method called retarded actuation with prediction presented better performance and the dynamic scheduling algorithm was updated based on this method to overcome computational delay. The new approach is illustrated through formation control of three radio controlled hovercraft system.**

I. INTRODUCTION

RECEDING horizon control (RHC) is a repeated online solution of a finite horizon open-loop optimal control problem [1].

Application of RHC to control problems with multiple subsystems is considered in this paper, which is addressed by applying RHC to the individual subsystems. These subsystems are dynamically decoupled. However, they are coupled in their cost index. This approach, results in multiple RHC processes that must be scheduled in an appropriate manner to achieve optimal performance in the presence of computing resource limitations.

In real-time implementation of control applications, normally each control task is considered as a periodic function [2]. Therefore, closed-loop implementation of RHC can be regarded as a periodic function with the period equal to the execution horizon of the system. Besides, it should be selected carefully to obtain a suitable trade-off between computational expense and controller performance.

In real-time implementation of a single RHC system, the execution horizon, which is equal to the sampling period, is selected based on the worst-case execution (computation) time. However, the computation time is highly varying [4], so considering worst-case computation time, leads to a conservative design, which requires high computational capacity. Besides, when multiple RHC systems are processed on the limited and shared computational resources (i.e., a single processor or a cluster of finite number of processors), distributing the computational resources between different systems (computational scheduling) is not a trivial task and needs careful considerations.

In recent years, some attempts have been made to use control theories in scheduling of computational systems [7]. However, systematic methods for scheduling multiple RHC systems are rarely discussed in the literature (See [9] for a review on this field).

In the present paper, we consider the problem of controlling multiple uncertain nonlinear systems by means of concurrent decentralized RHC schemes. A new scheduling approach is proposed by combining the results from continuous time nonlinear systems theory and the concept of scheduling theory like Rate Monotonic Priority Assignment (RM) [3], [2]. The problem of multiple subsystems on a single processor without any coupling between different subsystems was studied by the authors in [9]. The approach was further extended to multiple processors with coupling between subsystems [10]. Here, the approach is extended to the case of coupled RHC systems while considering computational delay on a single processor.

Franz et al. [6] and Milam et al. [5] studied the application of RHC to Caltech Ducted Fan experimental setup. To compensate computational delay, they used two retarded actuation methods (with and without state prediction). Their experimental studies presented the effectiveness of this approach. However, based on the study presented here, this method with prediction has better performance than the method without prediction, and is used in this paper.

II. PROBLEM STATEMENT

Consider a network of n dynamically decoupled subsystems using the RHC approach. The subsystems are supposed to have connection with each other by exchanging their information. Furthermore, consider the following nominal equation for the i^{th} subsystem:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i\left(\mathbf{x}_i(t), \mathbf{u}_i(t), t\right), \qquad i = 1,\ldots,n \tag{1}$$

which serves as a model for the actual subsystem described by:

Manuscript received September 21, 2007.

A. Azimi and B. W. Gordon are with Department of Mechanical and Industrial Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, H3G 1M8, CANADA (phone: (514) 848-2424 ext. 7058; fax: (514) 848-3175; (e-mail: a_azi@encs.concordia.ca, bwgordon@encs.concordia.ca).

C. A. Rabbath is with Defence R&D Canada – Valcartier, 2459 Pie-XI Blvd North, Quebec City, QC, G3J 1X5, Canada (e-mail: Camille-Alain.Rabbath@drdc-rddc.gc.ca).

$$\dot{\hat{\mathbf{x}}}_i = \mathbf{f}_i\left(\hat{\mathbf{x}}_i(t), \mathbf{u}_i(t), t\right) + \mathbf{g}_i(\hat{\mathbf{x}}_i(t), \mathbf{u}_i(t), t), \ i = 1, \ldots, n \tag{2}$$

where $\mathbf{x}_i(t) \in \Re^n$ and $\hat{\mathbf{x}}_i(t) \in \Re^n$ are the nominal and actual states of the i^{th} system, respectively. The input vector $\mathbf{u}_i(t) \in \Re^m$ satisfies the constraints $\mathbf{u}_i(t) \in U_i$ ($\forall t \geq 0$), where U_i is the allowable set of inputs for system *i*.

Definition 1. The set A_i is called the neighboring set of subsystem *i*, and consists of any subsystem in the network that has direct interconnection with subsystem *i*.

The finite horizon cost associated to i^{th} subsystem is defined as follows:

$$\begin{aligned} J_i\left(t, T, \mathbf{x}_i, \mathbf{u}_i, \tilde{\mathbf{x}}_i\right) &= \int_t^{t+T} q_i\left(\mathbf{x}_i(\tau), \mathbf{u}_i(\tau)\right) d\tau \\ &+ \int_t^{t+T} \sum_{j \in A_i} g_{ij}\left(\mathbf{x}_i(\tau), \mathbf{x}_j(\tau)\right) d\tau \end{aligned} \tag{3}$$

where $\tilde{\mathbf{x}}_i$ is a vector containing the states of all neighbors of the i^{th} subsystem. g_{ij} is a function which defines the interaction between two nodes of the overall system $\mathbf{x}_i$ and $\mathbf{x}_j$. *T* is the optimization horizon of the RHC controller. The optimal cost is given by:

$$J_i^*\left(t, T, \mathbf{x}_i^*, \mathbf{u}_i^*, \tilde{\mathbf{x}}_i\right) = \inf_{\mathbf{u}_i(\cdot)} J_i\left(t, T, \mathbf{x}_i, \mathbf{u}_i, \tilde{\mathbf{x}}_i\right) \tag{4}$$

The optimized trajectory resulting from (4) is defined as $(\mathbf{x}_{T,i}^*(\tau;t), \mathbf{u}_{T,i}^*(\tau;t))$, $\tau \in (t, t+T]$. In the closed loop RHC the calculated input $\mathbf{u}_{T,i}^*(\tau;t)$ is applied to the actual system (2), and $\tau \in (t, t+\delta_i]$, while δ_i is called the *Execution Horizon* of system *i* ($\delta_i < T$).

Remark 1. In the presented decentralized RHC formulations, the estimated trajectories of neighbors are communicated and the i^{th} subsystem is only estimating its own trajectory [8]. Therefore, the dimension of the optimization problem associated to each subsystem is reduced, which decreases the computation expenses, as well.

A. Real-time scheduling of multiple RHC systems with coupling on a single processor

Suppose that the systems described in (1) and (2) are connected to a single computer for feedback control. From a computer control point of view, each control system can be handled as a periodic task in the real-time programming. The period of each periodic task is equal to the *execution horizon* of its related subsystem.

For the proposed approach, the execution horizons of all subsystems should be defined such that the overall performance of the system is maximized. In order to evaluate the performance of the system, the following cost function is proposed as the cost of the closed loop system from time *t* to $t + T_{sc}$, where T_{sc} is the period that calculated execution horizons would be applied to.

$$\hat{J}_{sc} = \sum_{i=1}^{n} \left(\begin{array}{l} \sum_{k=1}^{d_i} \left(\int_{t_k^i}^{t_k^i + \delta_i} \eta_i\left(\tau, t_k^i\right) d\tau \right) + \\ \int_{t + d_i \delta_i}^{t + T_{sc}} \eta_i\left(\tau, t + d_i \delta_i\right) d\tau \end{array} \right) \tag{5}$$

where

$$\eta_i(\tau, t) = q_i\left(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau;t)\right) + \sum_{j \in A_i} g_{ij}\left(\hat{\mathbf{x}}_i(\tau), \mathbf{x}_{T,j}^*(\tau;t)\right) \tag{6}$$

In addition, $d_i = \lfloor T_{sc} / \delta_i \rfloor$, $t_k^i = t + (k-1)\delta_i$, and $\mathbf{u}_{T,i}^*\left(\tau; t_k^i\right)$ is the optimal input applied to subsystem *i*.

The idea is to find the execution horizon of each subsystem, such that $\hat{J}_{sc}$ is minimized.

Remark 2. as stated in Remark 1, the *estimated trajectories* of neighbors are used in the optimization of i^{th} system; therefore, the optimized neighbor trajectory is used in (6) instead of actual neighbor trajectory.

For computation scheduling algorithm, which is presented in this paper, the concept of Rate Monotonic Priority Assignment (RM) is used [3]. However, RM is useful in static scheduling only; it is adapted for schedulability condition in our dynamic scheduling algorithm. The system is schedulable using RM, for a set of *n* tasks, if the following inequality is valid [3]:

$$\mu = \sum_{i=1}^{n} \frac{\delta_{c,i}}{p_i} \leq n(2^{\frac{1}{n}} - 1) \tag{7}$$

where μ is CPU Utilization factor, $\delta_{c,i}$ is the computation time of subsystem *i*, and p_i is the period of task *i*, which is equal to the execution horizon of subsystem *i* ($p_i = \delta_i$).

III. Real-time implementation of a single RHC

In the case of real-time implementation of RHC, an optimization problem must be solved online. The time required to solve this optimization problem, should be considered in the problems with fast dynamics. Thereby, retarded actuation method [5] is used in this paper.

In this section, a discussion is presented in the performance of retarded actuation method with both approaches of with and without state prediction and it is shown that this method with prediction has a better performance. Besides, the result of this analysis is used in the dynamic scheduling approach.

In the case of dynamic scheduling, the execution horizons are not constant. This concept is depicted in Figure 1 for *one subsystem* and $\delta^{(1)}$, $\delta^{(2)}$, and $\delta^{(3)}$ present the change in execution horizon.

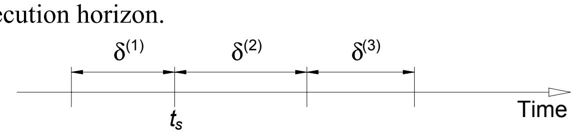

Figure 1- schematic diagram for dynamic scheduling; t_s is the start time of optimization and is equal to the sampling time.

We discuss both cases of *without prediction* and *with prediction*, in the dynamic scheduling case, by finding upper bounds on the state estimation errors. Furthermore, these upper bounds will be used in proposing the dynamic scheduling cost index. Besides, based on the upper bounds on the state estimation errors, the performance of both methods can be discussed considering the fact that smaller upper bound on the state estimation error, indicates more chances of gaining better performance.

Lemma 1. Consider the assumptions A.1 to A.4 of [11, Theorem 1] hold true, along with the following assumption:

A.5. $\|\mathbf{u}_1(t)-\mathbf{u}_2(t)\| \le \varepsilon$

where ε is a constant scalar. Then

$$\|\hat{\mathbf{x}}(t)-\mathbf{x}(t)\| \le b_s e^{L_x(t-t_0)} + \frac{b+\varepsilon L_u}{L_x}\left(e^{L_x(t-t_0)}-1\right) \tag{8}$$

where b and b_s represent the upper bounds on uncertainty and measurement error, respectively. L_u and L_x are Lipschitz constants. (See A.1 to A.4 of [11, Theorem 1] for details)

Proof: Taking integration from the result of [11, Theorem 1], results in (8). □

Lemma 2. If RHC method applies *without prediction*, in the case of dynamic scheduling (Figure 1), the optimization starts at time t_s and calculates the optimized input (which referred to as $\mathbf{u}_2$). This input is applied from $t_s+\delta^{(2)}$ to $t_s+\delta^{(2)}+\delta^{(3)}$. Since the new optimized input ($\mathbf{u}_2$) is different from the applied input to the actual system from t_s to $t_s+\delta^{(2)}$ (which is $\mathbf{u}_1$ and is the result of previous optimizations), the following relations are valid for the actual and nominal systems:

$$\begin{aligned}&\dot{\mathbf{x}}(t)=\mathbf{f}\left(\tau,\mathbf{x}(\tau),\mathbf{u}_2(\tau)\right)\\ &\dot{\hat{\mathbf{x}}}(t)=\mathbf{f}\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)+\mathbf{g}\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)\\ &\tau\in\left[t_s,t_s+\delta^{(2)}\right)\end{aligned} \tag{9}$$

$$\begin{aligned}&\dot{\mathbf{x}}(t)=\mathbf{f}\left(\tau,\mathbf{x}(\tau),\mathbf{u}_2(\tau)\right)\\ &\dot{\hat{\mathbf{x}}}(t)=\mathbf{f}\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_2(\tau)\right)+\mathbf{g}\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_2(\tau)\right)\\ &\tau\in\left[t_s+\delta^{(2)},t_s+\delta^{(2)}+\delta^{(3)}\right)\end{aligned} \tag{10}$$

where $\mathbf{u}_1$ is the result of previous optimizations. In addition, consider the assumptions of Lemma 1 hold true; then:

$$\begin{aligned}\|\hat{\mathbf{x}}(\tau)-\mathbf{x}(\tau)\| &\le b_s e^{L_x(\tau-t_s)} + \frac{b}{L_x}\left(e^{L_x(\tau-t_s)}-1\right)\\ &+\frac{\varepsilon L_u}{L_x}\left(e^{L_x(\tau-t_s)}-e^{L_x\left(\tau-t_s-\delta^{(2)}\right)}\right)\end{aligned} \tag{11}$$

where $\tau\in\left[t_s+\delta^{(2)},t_s+\delta^{(2)}+\delta^{(3)}\right)$.

Remark 3. b_s presents the bound on initial condition and initial conditions are the state of the systems at the sampling time. Therefore, $\|\hat{\mathbf{x}}(t_s)-\mathbf{x}(t_s)\| \le b_s$.

Proof: See [11]. □

Lemma 3. Consider RHC method applies *with prediction*, in the case of dynamic scheduling (Figure 1). The procedure for calculating new inputs starts at time t_s and the following steps are done:

- From t_s to $t_s+\delta^{(1)}$, the prediction of states are done based on the input $\mathbf{u}_1$ which is known from previous optimizations.
- The predicted states $\mathbf{x}^p(t_s+\delta^{(1)})$ are used as initial condition of optimization and the new input $\mathbf{u}_2$ is calculated. It is assumed that at time t_s, the new execution horizon is not known and the last execution horizon ($\delta^{(1)}$) is used for prediction.
- The new input $\mathbf{u}_2$ is applied to the system from $t_s+\delta^{(2)}$ to $t_s+\delta^{(2)}+\delta^{(3)}$

Therefore, if $\delta^{(1)} \le \delta^{(2)}$, from $t_s+\delta^{(1)}$ to $t_s+\delta^{(2)}$, in the nominal model ($\mathbf{x}$) that is used in the controller, new input ($\mathbf{u}_2$) is applied but in the actual system ($\hat{\mathbf{x}}$) the previous input ($\mathbf{u}_1$) was applied. This procedure is explained in the following:

$$\begin{aligned}&\dot{\mathbf{x}}(t)=f\left(\tau,\mathbf{x}(\tau),\mathbf{u}_1(\tau)\right)\\ &\dot{\hat{\mathbf{x}}}(t)=f\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)+g\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)\\ &\tau\in\left[t_s,t_s+\delta^{(1)}\right)\end{aligned} \tag{12}$$

$$\begin{aligned}&\dot{\mathbf{x}}(t)=f\left(\tau,\mathbf{x}(\tau),\mathbf{u}_2(\tau)\right)\\ &\dot{\hat{\mathbf{x}}}(t)=f\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)+g\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_1(\tau)\right)\\ &\tau\in\left[t_s+\delta^{(1)},t_s+\delta^{(2)}\right)\end{aligned} \tag{13}$$

$$\begin{aligned}&\dot{\mathbf{x}}(t)=f\left(\tau,\mathbf{x}(\tau),\mathbf{u}_2(\tau)\right)\\ &\dot{\hat{\mathbf{x}}}(t)=f\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_2(\tau)\right)+g\left(\tau,\hat{\mathbf{x}}(\tau),\mathbf{u}_2(\tau)\right)\\ &\tau\in\left[t_s+\delta^{(2)},t_s+\delta^{(2)}+\delta^{(3)}\right)\end{aligned} \tag{14}$$

where $\delta^{(1)} \le \delta^{(2)}$. In addition, consider the assumptions of Lemma 1 hold true. The state estimation error can be bounded as follows:

$$\begin{aligned}\|\hat{\mathbf{x}}(\tau)-\mathbf{x}(\tau)\| &\le b_s e^{L_x(\tau-t_s)} + \frac{b}{L_x}\left(e^{L_x(\tau-t_s)}-1\right)\\ &+\frac{\varepsilon L_u}{L_x}\left(e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}-1\right)e^{L_x\left(\tau-t_s-\delta^{(2)}\right)}\end{aligned} \tag{15}$$

where $\tau\in\left(t_s+\delta^{(2)},t_s+\delta^{(2)}+\delta^{(3)}\right]$.

Proof: From (12) and Lemma 1:

$$\left\|\hat{\mathbf{x}}(t_s+\delta^{(1)})-\mathbf{x}(t_s+\delta^{(1)})\right\| \le b_s e^{L_x\delta^{(1)}} + \frac{b}{L_x}\left(e^{L_x\delta^{(1)}}-1\right) \tag{16}$$

In addition, using Lemma 1 and (13) result in:

$$\begin{aligned}\left\|\hat{\mathbf{x}}(t_s+\delta^{(2)})-\mathbf{x}(t_s+\delta^{(2)})\right\| &\le \frac{b+\varepsilon L_u}{L_x}\left(e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}-1\right)\\ &+\left\|\hat{\mathbf{x}}(t_s+\delta^{(1)})-\mathbf{x}(t_s+\delta^{(1)})\right\| e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}\end{aligned} \tag{17}$$

Combination of (16) and (17) results in:

$$\left\|\hat{\mathbf{x}}(t_s+\delta^{(2)})-\mathbf{x}(t_s+\delta^{(2)})\right\|\le\frac{b+\varepsilon L_u}{L_x}\left(e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}-1\right)$$
$$+\left(b_s e^{L_x\delta^{(1)}}+\frac{b}{L_x}\left(e^{L_x\delta^{(1)}}-1\right)\right)e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)} \tag{18}$$

which leads to:

$$\left\|\hat{\mathbf{x}}(t_s+\delta^{(2)})-\mathbf{x}(t_s+\delta^{(2)})\right\|\le b_s e^{L_x\delta^{(2)}}$$
$$+\frac{b}{L_x}\left(e^{L_x\delta^{(2)}}-1\right)+\frac{\varepsilon L_u}{L_x}\left(e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}-1\right) \tag{19}$$

Furthermore, using Lemma 1 and (14) result in:

$$\left\|\hat{\mathbf{x}}(\tau)-\mathbf{x}(\tau)\right\|\le\frac{b}{L_x}\left(e^{L_x\left(\tau-t_s-\delta^{(2)}\right)}-1\right)$$
$$+\left\|\hat{\mathbf{x}}(t_s+\delta^{(2)})-\mathbf{x}(t_s+\delta^{(2)})\right\|e^{L_x\left(\tau-t_s-\delta^{(2)}\right)} \tag{20}$$

From combination of (19) and (20):

$$\left\|\hat{\mathbf{x}}(\tau)-\mathbf{x}(\tau)\right\|\le\frac{b}{L_x}\left(e^{L_x\left(\tau-t_s-\delta^{(2)}\right)}-1\right)$$
$$+\begin{pmatrix} b_s e^{L_x\delta^{(2)}}+\frac{b}{L_x}\left(e^{L_x\delta^{(2)}}-1\right)\\ +\frac{\varepsilon L_u}{L_x}\left(e^{L_x\left(\delta^{(2)}-\delta^{(1)}\right)}-1\right)\end{pmatrix}e^{L_x\left(\tau-t_s-\delta^{(2)}\right)} \tag{21}$$

Rearranging right side of (21) results in (15) that completes the proof. □

Discussion 1. For the case of $\delta^{(1)}>\delta^{(2)}$, the upper bound on the state estimation error is smaller than (15). Therefore, for the case of general changing in the execution horizon, the following inequality can be used:

$$\left\|\hat{\mathbf{x}}(\tau)-\mathbf{x}(\tau)\right\|\le b_s e^{L_x(\tau-t_s)}+\frac{b}{L_x}\left(e^{L_x(\tau-t_s)}-1\right)$$
$$+\frac{\varepsilon L_u}{L_x}\left(e^{L_x\left|\delta^{(2)}-\delta^{(1)}\right|}-1\right)e^{L_x\left(\tau-t_s-\delta^{(2)}\right)} \tag{22}$$

Besides, comparing (11) and (22) shows that using *with prediction* method, the error on the state estimation will be limited by a smaller bound, which means more chance of gaining better performance. Therefore, this method is used in real-time application of RHC systems in this paper.

IV. Dynamic Scheduling of Multiple RHC Systems with Coupling and Computational Delay

Equation (5) required some data for evaluation, which are not available at time *t*. It needs the future states of the actual subsystems and the future optimized inputs (from time *t* to $t+T_{sc}$) for calculating $\hat{J}_{sc}$ which are not available at current time (*t*). Therefore, instead of calculating $\hat{J}_{sc}$, the following cost is proposed which is an estimation of the cost in (5):

$$\bar{J}_{sc}(\hat{\mathbf{x}}(t),\mathbf{u}_T^*(.))=\sum_{i=1}^{n}\left(\frac{T_{sc}}{\delta_i}\int_t^{t+\delta_i}\eta_i(\tau,t)\,d\tau\right) \tag{23}$$

Corollary 1. Consider (22) which is the result of Section III. If the dynamic scheduling method is applied such a way that $\delta^{(2)}=\delta^{(3)}\equiv\delta$ and $\delta^{(1)}\equiv\delta^{p}$, where δ^{p} is the previous execution horizon and δ is the new one, (22) can be presented in the following, if *i* indicates subsystem number:

$$\left\|\hat{\mathbf{x}}_i(\tau)-\mathbf{x}_i(\tau)\right\|\le b_{s,i}e^{L_{x,i}\left(\tau-t_{s,i}\right)}$$
$$+\frac{b_i}{L_{x,i}}\left(e^{L_{x,i}\left(\tau-t_{s,i}\right)}-1\right)+\frac{\varepsilon_i L_{u,i}}{L_{x,i}}\left(e^{L_{x,i}\left|\delta_i-\delta_i^{p}\right|}-1\right)e^{L_{x,i}\left(\tau-t_i\right)} \tag{24}$$

where t_i represents the time that new inputs are applied to subsystem *i* and $\tau\in\left(t_i,t_i+\delta_i\right]$.

Lemma 4. Consider the following assumptions:

1- If $\mathbf{x}_i$ and $\mathbf{x}_j$ are two $r\times1$ column vectors, $\mathbf{x}_{ij}$ is a $2r\times1$ column vector such that $\mathbf{x}_{ij}=\left[\mathbf{x}_i{}^T,\mathbf{x}_j{}^T\right]^T$.

2- Let $y(\mathbf{x}_{ij})=g_{ij}(\mathbf{x}_i,\mathbf{x}_j)$ be Lipschitz continuous with constant L_{ij}^{g}, and $y(\mathbf{x}_{ij})$ be a *positive scalar*.

3- *Retarded actuation* method is used *with prediction* as explained in the Section III.

Then:

$$g_{ij}\left(\hat{\mathbf{x}}_i(\tau),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right)\le g_{ij}\left(\mathbf{x}_{T,i}^*(\tau;t_{s,i}),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right)$$
$$+L_{ij}^{g}\begin{pmatrix} b_{s,i}e^{L_{x,i}\left(\tau-t_{s,i}\right)}+\frac{b_i}{L_{x,i}}\left(e^{L_x\left(\tau-t_{s,i}\right)}-1\right)\\ +\frac{\varepsilon_i L_{u,i}}{L_{x,i}}\left(e^{L_{x,i}\left|\delta_i-\delta_i^{p}\right|}-1\right)e^{L_{x,i}\left(\tau-t_i\right)}\end{pmatrix} \tag{25}$$

where $\mathbf{x}_{T,i}^*(\tau;t_{s,i})$ is the optimal trajectory of subsystem *i* resulted from optimization index presented in (3) and the sampled data at time $t_{s,i}$. Similarly, $\mathbf{x}_{T,j}^*(\tau;t_{s,j})$ is for subsystem *j*.

Proof: From Lipschitz continuity of positive scalar $y(.)$:

$$y\left(\left[\hat{\mathbf{x}}_i(\tau),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right]^T\right)\le L_{ij}^{g}\left\|\left(\hat{\mathbf{x}}_i(\tau)-\mathbf{x}_{T,i}^*(\tau;t_{s,i})\right)\right\|$$
$$+y\left(\left[\mathbf{x}_{T,i}^*(\tau;t_{s,i})^T,\mathbf{x}_{T,j}^*(\tau;t_{s,j})^T\right]^T\right) \tag{26}$$

Using Corollary 1, equation (24) results in the following:

$$\left\|\hat{\mathbf{x}}_i(\tau)-\mathbf{x}_{T,i}^*(\tau;t_{s,i})\right\|\le b_{s,i}e^{L_{x,i}\left(\tau-t_{s,i}\right)}$$
$$+\frac{b_i}{L_{x,i}}\left(e^{L_x\left(\tau-t_{s,i}\right)}-1\right)+\frac{\varepsilon_i L_{u,i}}{L_{x,i}}\left(e^{L_{x,i}\left|\delta_i-\delta_i^{p}\right|}-1\right)e^{L_{x,i}\left(\tau-t_i\right)} \tag{27}$$

Therefore, combination of (26) and (27) results in (25) which completes the proof. □

Lemma 5. Suppose the following assumptions hold true:

1- q_i is quadratic, so: $q_i(\mathbf{x}_i,\mathbf{u}_i)=\mathbf{x}_i^T Q_i\mathbf{x}_i+\mathbf{u}_i^T R_i\mathbf{u}_i$

2- $\bar{Q}_i(\mathbf{x}_i)=\mathbf{x}_i^T Q_i\mathbf{x}_i$ is Lipschitz continuous with constant P_i.

3- $\mathbf{f}_i(t,\mathbf{x}_i,\mathbf{u}_i)$ in (1) is piecewise continuous in *t* and

Lipschitz in $\mathbf{x}_i$ and $\mathbf{u}_i$ with Lipschitz constants $L_{x,i}$ and $L_{u,i}$, respectively.

4- $\mathbf{g}_i(t,\mathbf{x}_i,\mathbf{u}_i)$ in (2) is bounded and $\left\|\mathbf{g}_i(t,\mathbf{x}_i,\mathbf{u}_i)\right\| \le b_i$.

5- *Retarded actuation* method is used *with prediction* as explained in the Section III.

6- Assumptions of Corollary 1 are valid.

Using [9, Proposition 1], Corollary 1, and Lemma 4, it is straightforward to find the following upper bound for the overall closed loop cost, expressed in (23):

$$\sum_{i=1}^{n}\left(\frac{T_{sc}}{\delta_i}\int_{t_i}^{t_i+\delta_i}\left(\begin{array}{l} q_i\left(\hat{\mathbf{x}}_i(\tau),\mathbf{u}_{T,i}^*(\tau;t_{s,i})\right) \\ +\sum_{j\in A_i} g_{ij}\left(\hat{\mathbf{x}}_i(\tau),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right)\end{array}\right)d\tau\right)$$
$$\le \sum_{i=1}^{n}\left(\frac{T_{sc}}{\delta_i}\left(\begin{array}{l}\int_{t_i}^{t_i+\delta_i} q_i\left(\mathbf{x}_{T,i}^*(\tau;t_{s,i}),\mathbf{u}_{T,i}^*(\tau;t_{s,i})\right)d\tau \\ +P_i\left(B_i^{(1)}+B_i^{(2)}\right)\end{array}\right)\right) \quad (28)$$
$$+\sum_{i=1}^{n}\left(\frac{T_{sc}}{\delta_i}\sum_{j\in A_i}\left(\begin{array}{l}\int_{t_i}^{t_i+\delta_i} g_{ij}\left(\mathbf{x}_{T,i}^*(\tau;t_{s,i}),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right)d\tau \\ +L_{ij}^g\left(B_i^{(1)}+B_i^{(2)}\right)\end{array}\right)\right)$$

where

$$B_i^{(1)} = \frac{b_{s,i}}{L_{x,i}}\left(e^{2L_{x,i}\delta_i}-e^{L_{x,i}\delta_i}\right)+\frac{b_i}{\left(L_{x,i}\right)^2}\left(e^{2L_{x,i}\delta_i}-e^{L_{x,i}\delta_i}\right)-\frac{b_i\delta_i}{L_{x,i}}$$
$$B_i^{(2)} = \frac{\varepsilon_i L_{u,i}}{\left(L_{x,i}\right)^2}\left(e^{L_{x,i}\left|\delta_i-\delta_i^p\right|}-1\right)\left(e^{L_{x,i}\delta_i}-1\right) \quad (29)$$

Remark 4. For proof of this Lemma, see [11].

Proposition 1: Consider n dynamically decoupled subsystems with equations presented in (1) and (2), controlled by RHC using only one processor. In addition, the subsystems may be coupled due to state information exchanging that was presented in (3). Besides, the assumptions of Lemma 5 are valid. Since uncertainties in the subsystems are different and the measurements are performed with bounded sensor noise, show that the following constraint optimization problem can be used to determine the execution horizons of all subsystems, optimally:

$$\min_{\delta_i}\sum_{i=1}^{n}\left(\begin{array}{l}\frac{\alpha_i}{\delta_i}\left(\begin{array}{l}\int_{t_i}^{t_i+\delta_i} q_i\left(\mathbf{x}_{T,i}^*(\tau;t_{s,i}),\mathbf{u}_{T,i}^*(\tau;t_{s,i})\right)d\tau \\ +P_i\left(B_i^{(1)}+B_i^{(2)}\right)\end{array}\right)+ \\ \sum_{j\in A_i}\frac{\beta_{ij}}{\delta_i}\left(\begin{array}{l}\int_{t_i}^{t_i+\delta_i} g_{ij}\left(\mathbf{x}_{T,i}^*(\tau;t_{s,i}),\mathbf{x}_{T,j}^*(\tau;t_{s,j})\right)d\tau \\ +L_{ij}^g\left(B_i^{(1)}+B_i^{(2)}\right)\end{array}\right)\end{array}\right)$$

$$\text{Subject to:}\begin{cases} C1: \delta_{lb,i} < \delta_i \le \delta_{ub,i} \\ C2: \sum \frac{\delta_{c,i}}{\delta_i} \le n(2^{\frac{1}{n}}-1)\end{cases} \quad (30)$$

where $\delta_{ub,i}$ and $\delta_{ub,i}$ are the maximum and minimum acceptable execution horizons for subsystem i, respectively; $B_i^{(1)}$ and $B_i^{(2)}$ are defined in (29), α_i and β_{ij} are weighting parameters.

Proof: Lemma 5 presented an upper bound on the overall closed loop cost of (23). If δ_i determined such that result in the reduction of the foresaid overall cost, the performance of the overall system should be improved. Instead of directly minimization of overall cost, which we do not have access to that, our proposed scheduling algorithm is to minimize its upper bound. In addition, as explained before, equation (7) must hold to guarantee schedulability of the system using RM method. Therefore, it is a straightforward to propose (30) for scheduling algorithm, considering α_i and β_{ij} as the weighting parameters. □

Based on Proposition 1, the dynamic scheduling algorithm can be expressed. Note that this algorithm must be updated every T_{sc}.

1- Calculate the scheduler parameters (i.e., P_i), as explained in [10].

2- Estimate $\delta_{c,i}$ from previous computational time of each RHC calculation (See [9]) or use the worst-case computational time.

3- Solve (30) and calculate δ_i for all subsystems.

4- Update the next execution horizons by the calculated δ_i from step 4.

5- Wait for the next T_{sc} seconds and repeat the procedure from step 1.

V. Application to Hovercraft Problem

The proposed scheduling algorithm is applied to the concurrent control of multiple unmanned radio controlled (RC) hovercrafts on multiple computers. Modeling of the RC hovercraft is presented in [9].

A. Simulation results

The simulation results obtained by applying the proposed dynamic scheduling algorithm to formation control of three RC hovercrafts on a single processor are presented. Subsystem 1 is following the trajectory (leader) while the others maintain the formation (followers). The followers check their position with the leader. In addition, subsystem 2 has more uncertainty than the others do. The proposed dynamic scheduling algorithm is compared to the result of no-scheduling case. This means using similar execution horizons for all subsystems resulted from worst-case analysis.

The simulations are performed using a 3.2 GHz Intel Pentium IV processor, Microsoft Visual C++ 6.0, and the RHC Object Oriented Library (RHCOOL) [9]. In addition, we used Venturcom RTX 6.0.1 to implement the approach in a hard real-time environment.

For the RC hovercraft example with model parameters presented in [9], the scheduling parameters are found offline for different values of states. These parameters are used later in the dynamic scheduling. In addition, the uncertainties in the system are modeled using white noise.

The paths followed by systems in dynamic scheduling and no-scheduling cases are presented in Figure 2 and Figure 3, respectively. The computation time of each subsystem needed in (30) was selected based on the worst-case computation, and used for both dynamic and static scheduling cases. In addition, for no-scheduling (static scheduling) case, the execution horizon of all subsystems were equal to 0.8 seconds and calculated from (7).

Remark 5. As explained in [9], one can estimate the computation time and use the processor more effectively. However, in this paper worst-case computation is used for both static and dynamic schedulers to compare the presented dynamic scheduling method to a common static scheduler. If the computation time is estimated properly, the dynamic scheduler will be more effective.

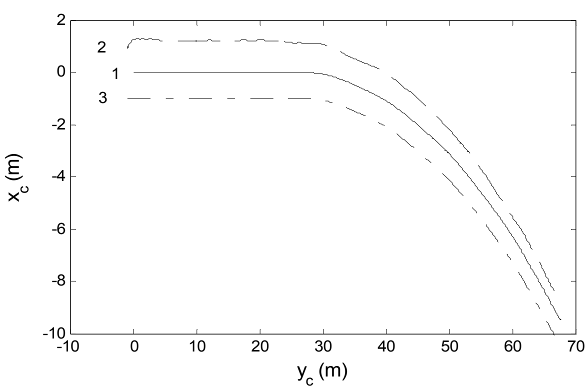

Figure 2- Paths followed by three hovercrafts in dynamic scheduling using the proposed algorithm

As presented in these two figures, in dynamic scheduling case, the second subsystem maintain the formation better than the static scheduling case and there is no difference for the other two hovercrafts.

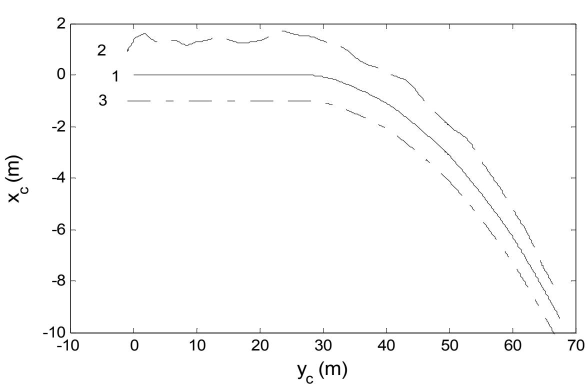

Figure 3- Paths followed by three hovercrafts in no scheduling case

VI. Conclusion and Future Work

In this paper, a new method was developed for dynamic scheduling of multiple decentralized RHC systems on a single processor. The problems studied in this paper, have nonlinear dynamics subject to computation delay and uncertainties in the model and sensor noise. In addition, the subsystems were coupled due to the information exchange between deferent subsystems. Execution horizons of all subsystems were determined dynamically using the proposed method. The proposed algorithm was applied to formation control of three RC hovercrafts simulations to illustrate the new approach. For future work, the scheduling method will be applied to an experimental apparatus for multiple RC hovercrafts with an overhead vision system for feedback. Extensions to computer clusters will also be investigated.

Acknowledgment

This work was partially funded through the Defence Research and Development Canada (DRDC). The authors would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for partial funding of this project.

References

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

[2] H. Kopetz, "Real-time systems: Design principles for distributed embedded applications", Chapters 9 and 11, *Springer*, 1997.

[3] C. L. Liu, J. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment", *Journal of the ACM*, 20(1), pp. 46-61, 1973.

[4] D. Henriksson, J. Akesson, "Flexible implementation of model predictive controller using sub-optimal solutions", *Technical Report*, Department of Automatic Control, Lund Institute of Technology, Sweden, 2004.

[5] M. B. Milam, R. Franz, J. E. Hauser, R. M. Murray, "Receding horizon control of a vectored thrust flight experiment", *IEE Proceedings on Control Theory and Applications*, 152(3), 2005, pp. 340-348.

[6] R. Franz, M. Milam, J. Hauser "Applied receding control of Caltech ducted fan", *Proc. of the American Control Conference*, pp. 3735-3740, 2002.

[7] C. Lu, J. A. Stankovic, S. H. Son, G. Tao, "Feedback control real-time scheduling: framework, modeling, and algorithms", *Real-Time Systems*, Vol. 23, pp. 85-126, 2002.

[8] T. Keviczky, "Decentralized receding horizon control of large scale dynamically decoupled systems", *Ph.D. Thesis, Control Science and Dynamical Systems Center, University of Minnesota,* September 2005.

[9] A. Azimi, B. W. Gordon, C. A. Rabbath, "Dynamic scheduling of receding horizon controllers with application to multiple unmanned hovercraft systems", *Proceedings of the American Control Conference*, USA, July 2007, pp. 3324-3329.

[10] A. Azimi, B. W. Gordon, C. A. Rabbath, "Dynamic scheduling of decentralized receding horizon controllers on concurrent processors for the cooperative control of unmanned systems", in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, December 2007, pp. 518-523.

[11] A. Azimi, B. W. Gordon, C. A. Rabbath, "Dynamic scheduling of multiple coupled receding horizon controllers", To be submitted to *IEEE Transactions on Control Systems Technology*, 2008. http://users.encs.concordia.ca/~cis/IEEETCST08_Ali_01.pdf