A Game Theory Approach To Distributed Multi-Rate Model Predictive Control

Samira Roshany-Yamchi[†], Rudy R. Negenborn[*], Marcin Cychowski[†], Kieran Delaney[†] and Joe Connell[†]

[†]*NIMBUS Centre, Cork Institute of Technology, Cork, Ireland*

[*]*Department of Marine and Transport Technology Delft University of Technology, The Netherlands*

E-mail: [†]samira.roshany@cit.ie [*]r.r.negenborn@tudelft.nl
[†]marcin.cychowski@cit.ie
[†]kieran.delaney@cit.ie
[†]joe.connell@cit.ie

Abstract — **In this paper, we study the effect of tuning factors on our formerly proposed method [1] for control of distributed multi-rate systems with linear input coupled dynamics. These systems are multi-rate in the sense that neither output measurements nor input updates are available at certain sampling times. Such systems can arise when the number of sensors is less than the number of variables to be controlled. Or when measurements of outputs cannot be completed simultaneously because of application limitations. The multi-rate nature gives rise to lack of information which will cause uncertainty in the system's performance. A distributed model predictive control (MPC) approach based on Nash game theory was proposed earlier in [1] to control distributed multi-rate systems in which multiple control agents each determined actions for their own parts of the system. Via communication, the agents can take one another's actions into account. To compensate for the information loss due to the multi-rate nature of the system in this study, a distributed Kalman Filter is proposed to provide optimal estimation of the missing information. The effect of changing control and prediction horizons as control tuning parameters on performance is studied. Using simulation studies on a distillation column the effect of tuning parameters on the proposed controller is illustrated.**

Keywords — **Multi-Agent Systems, Distributed Predictive Control, Distributed Estimation, Nash game.**

I Introduction

Model predictive control (MPC) is a popular technique and has been successfully used in the control of various linear and nonlinear dynamic systems [2, 3]. The main advantage of MPC is that the objectives and constraints associated with the control problem are embedded in the control algorithm. However, an obvious drawback of MPC for large-scale systems is that it is computationally intensive resulting in formidable on-line computational effort. Practically, there exist a great number of complex high dimensional systems in which the number of variables and constraints is large. Thus, it has become very important to develop computationally efficient control architectures and algorithms with less computational burden. Many distributed MPC methods have been developed by researchers to cope with large-scale control problems. Examples include the work in [3, 4, 5]. In multi-rate plants either the measurements are available less frequently or the control moves are made at a lower rate. This kind of system can be seen in many industrial applications [6],[7] and it illustrates practical situations in which measurements of process variables and input updates occur at different rates. For instance, they illustrate that in the process industry even when quality vari-

ables such as product concentration or the average molecular weight distribution of a polymerization process are measured on-line time delays involved in the measurements are significantly large when compared to other process measurements. The aim of our proposed framework is to develop a generalized scheme that covers both of these aspects for large-scale systems. In [6] a state space based multi-rate MPC scheme for the centralized case has been developed. In a distributed MPC architecture the overall system is decomposed into a number of small subsystems. Each subsystem is controlled by a so-called *agent* which solves its own local optimization problem. We proposed a new MPC control strategy for large-scale systems with *multi-rateness* in its subsystems. This means that each of the subsystems is multi-rate in inputs and/or outputs. The multi-rate control method that we have developed allows control moves to be made using state estimates from a distributed Kalman Filter.

II Distributed Multi-Rate MPC

a) State-Space Representation

Consider distributed MPC of plants with linear dynamics whose centralized nominal model is decomposed into m subsystems. Let t be the global discrete-time index for the system under control. Let the following augmented model [1] represents a distributed multi-rate system with input coupling:

$$\mathbf{x}_i(t+1) = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(t) + \mathbf{D}_i\mathbf{v}_i(t) + \sum_{\substack{j=1\\ j\neq i}}^{m}\mathbf{B}_{ij}\Delta\mathbf{u}_j(t) \quad (1)$$

$$\mathbf{y}_i(t) = \mathbf{C}_i\mathbf{x}_i(t) + \mathbf{z}_i(t), \quad (2)$$

where for each subsystem i, $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the state and output variables respectively, $\mathbf{v}_i$, $\mathbf{z}_i$ are the state and measurement noise sequences respectively. $\mathbf{A}_i$, $\mathbf{B}_{ii}$, $\mathbf{B}_{ij}$, $\mathbf{D}_i$ and $\mathbf{C}_i$ are matrices of appropriate dimensions. Variable $\Delta\mathbf{u}_i(t)$ is the multi-rate input signal which is injected into subsystem i at sampling time t. As can be seen in (1), subsystems are coupled through inputs only ($\Delta\mathbf{u}_j(t)$) which restricts the use to linear plants with block diagonal matrix $\mathbf{A}_i$ when the plant under control is in continuous-time form.

Assume that the sampling instants for the system vary as $t = 1, 2, \ldots, T_f$, where T_f is the final sampling time. We follow a strategy similar to [5] to implement multi-rate measurement or input updating mechanisms for subsystem i. In a multi-rate output setting, the output vector $\mathbf{y}_i(t)$ of subsystem i can be measured every T_{y_i} units, where $T_{y_i} > 0$. Define the output switching function for subsystems i, γ_{i_j}, for $j = 1, 2, \ldots, q_i$ with q_i being the number of outputs of subsystem i as follows:

$$\gamma_{i_j}(t) = \begin{cases} 1 & \text{if } t = \tau T_{y_j} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where τ is an integer. The following multi-rate output vector $\boldsymbol{\varphi}_i(t)$ can now be defined:

$$\boldsymbol{\varphi}_i(t) = \boldsymbol{\Upsilon}_i(t)\mathbf{y}_i(t), \quad (4)$$

where

$$\boldsymbol{\Upsilon}_i(t) = \mathrm{diag}[\gamma_{i_1}(t)\ \ \gamma_{i_2}(t)\ \ \ldots\ \ \gamma_{i_q}(t)]. \quad (5)$$

In a multi-rate input setting, the input vector $\boldsymbol{\Delta}\mathbf{u}_i(t)$ of subsystem i is updated every T_{u_i} units, where $T_{u_i} > 0$. Introduce the input switching function μ_{i_j} for $j = 1, 2, \ldots, l_i$ with l_i being the number of inputs of subsystem i. Define the inputs holding mechanism as:

$$\mu_{i_j}(t) = \begin{cases} 1 & \text{if } t = \tau T_{u_j} \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where τ is an integer. The following input matrix $\boldsymbol{\Psi}_i(t)$ for subsystem i can be defined.

$$\boldsymbol{\Psi}_i(t) = \mathrm{diag}\left[\mu_{i_1}(t)\ \ \mu_{i_2}(t)\ \ \ldots\ \ \mu_{i_l}(t)\right]. \quad (7)$$

Now a new control variable $\boldsymbol{\vartheta}_i(t)$ is introduced to implement the input administering mechanism:

$$\Delta\boldsymbol{u}_i(t) = \boldsymbol{\Psi}_i(t)\boldsymbol{\vartheta}_i(t). \quad (8)$$

In fact, the control input computed by the local MPC controller is $\boldsymbol{\vartheta}_i(t)$ and not $\boldsymbol{\Delta}\mathbf{u}_i(t)$. However, in a multi-rate system the manipulated variables are defined as in (8) which includes both the computed inputs and the input updating matrix $\boldsymbol{\Psi}_i(t)$. After substituting (8) into (1) we get:

$$\mathbf{x}_i(t+1) = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_{ii}\boldsymbol{\Psi}_i(t)\boldsymbol{\vartheta}_i(t) + \mathbf{D}_i\mathbf{v}_i(t) + \sum_{\substack{j=1\\ j\neq i}}^{m}\mathbf{B}_{ij}\boldsymbol{\Psi}_j(t)\boldsymbol{\vartheta}_j(t). \quad (9)$$

As multi-rate systems' output measurements are made at specific sampling times, the output sampling mechanism needs to be included in the system's model. To do that, both sides of (2) are multiplied by the output sampling parameter $\boldsymbol{\Upsilon}_i(t)$:

$$\boldsymbol{\varphi}_i(t) = \boldsymbol{\Upsilon}_i(t)\mathbf{C}_i\mathbf{x}_i(t) + \boldsymbol{\Upsilon}_i(t)\mathbf{z}_i(t). \quad (10)$$

Equations (9) and (10) give the linear state-space representation of the distributed multi-rate system for $i = 1, 2, \ldots, m$. Next, the Nash-based solution to distributed MPC problem will be formulated for such a system.

[1] Augmenting the state-space model is an approach to include the *integration* in state-space model when Δu is used as input signal instead of u [2].

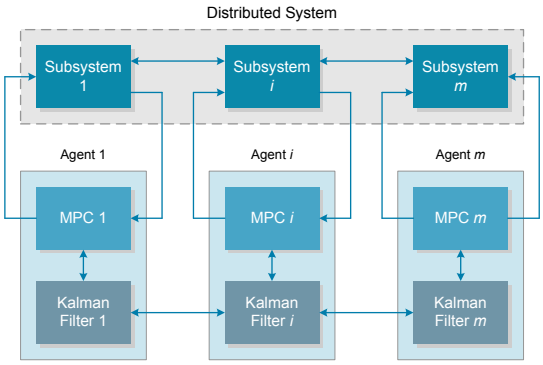

Fig. 1: Distributed control and estimation architecture.

b) Nash Game Approach

In the distributed control structure, input coupling among subsystems is considered as given by (9). These subsystems communicate with one another to accomplish a global objective (see Fig. 1). One type of Distributed MPC based on Nash optimality has been investigated in [4, 5]. In this approach, the agents communicate but they do not take a cooperative decision. The agents iterate to resolve their local optimization problem simultaneously and obtain their optimal solution [5]. An initial guess for each agent is first given based on the solution found at the last sampling time. Then, each agent checks if its terminal condition satisfies a user-defined threshold. This implies that the agents do not share information about the utility of each decision; agreement (Nash equilibrium) among the agents is reached when neither of their solutions can be improved. The main advantage of this scheme is that the on-line optimization of a large-scale problem can be converted into several small-scale subproblems. Thus, reducing the computational complexity significantly while keeping satisfactory performance.

Consider a linear multi-rate system consisting of m subsystems and m control agents (9),(10). In Nash-based distributed MPC each control agent calculates the manipulated variable $\boldsymbol{\vartheta}_i(t)$ by minimizing its local cost function as follows:

$$\min_{\boldsymbol{\theta}_i(t))} \; J_i(t) = \left\|\mathbf{Y}_i(t) - \mathbf{Y}_i^0(t)\right\|^2_{\mathbf{Q}_i} + \|\boldsymbol{\theta}_i(t)\|^2_{\mathbf{R}_i}, \tag{11}$$

subject to

$$\boldsymbol{\theta}_{i,\min} \le \boldsymbol{\theta}_i(t) \le \boldsymbol{\theta}_{i,\max}, \tag{12}$$

$$\mathbf{Y}_i(t) = \mathbf{F}_i\mathbf{x}_i(t) + \boldsymbol{\phi}_{ii}(t)\boldsymbol{\theta}_i(t) + \boldsymbol{\Gamma}_i\boldsymbol{\zeta}_i(t) + \sum_{\substack{j=1\\ j\neq i}}^{m} \boldsymbol{\phi}_{ij}(t)\boldsymbol{\theta}_j(t), \tag{13}$$

where $\boldsymbol{\theta}_{i,\min}$ and $\boldsymbol{\theta}_{i,\max}$ are the lower and upper bounds for the inputs, respectively. And,

$$\begin{aligned}
\mathbf{Y}_i(t) &= [\mathbf{y}_i^{\mathrm{T}}(t+1)\;\; \mathbf{y}_i^{\mathrm{T}}(t+2), \ldots, \mathbf{y}_i^{\mathrm{T}}(t+N_p)]^T,\\
\mathbf{Y}_i^0(t) &= [\mathbf{y}_i^{0\mathrm{T}}(t+1)\; \mathbf{y}_i^{0\mathrm{T}}(t+2), \ldots, \mathbf{y}_i^{0\mathrm{T}}(t+N_p)]^T,\\
\boldsymbol{\theta}_i(t) &= [\boldsymbol{\vartheta}_i^{\mathrm{T}}(t)\;\; \boldsymbol{\vartheta}_i^{\mathrm{T}}(t+1), \ldots, \boldsymbol{\vartheta}_i^{\mathrm{T}}(t+N_c-1)]^T,\\
\boldsymbol{\zeta}_i(t) &= [\mathbf{v}_i^{\mathrm{T}}(t)\;\; \mathbf{v}_i^{\mathrm{T}}(t+1), \ldots, \mathbf{v}_i^{\mathrm{T}}(t+N_c-1)]^T,\\
\mathbf{F}_i &= [(\mathbf{C}_i\mathbf{A}_i)^{\mathrm{T}}\;\; (\mathbf{C}_i\mathbf{A}_i^2)^{\mathrm{T}}, \ldots, (\mathbf{C}_i\mathbf{A}_i^{N_p})^{\mathrm{T}}]^T,
\end{aligned} \tag{14}$$

$$\boldsymbol{\phi}_{ij}(t) = \begin{bmatrix}
\mathbf{C}_i\mathbf{B}_{ij}\boldsymbol{\Psi}_j(t) & 0 & \ldots & 0\\
\mathbf{C}_i\mathbf{A}_i\mathbf{B}_{ij}\boldsymbol{\Psi}_j(t) & \vdots & \ddots & \vdots\\
\vdots & \vdots & \vdots & 0\\
\mathbf{C}_i\mathbf{A}_i^{N_p-1}\mathbf{B}_{ij}\boldsymbol{\Psi}_j(t) & \ldots & \ldots & \mathbf{C}_i\mathbf{A}_i^{N_p-N_c}\mathbf{B}_{ij}\boldsymbol{\Psi}_j(t+N_c-1)
\end{bmatrix} \tag{15}$$

$$\boldsymbol{\Gamma}_i = \begin{bmatrix}
\mathbf{C}_i\mathbf{D}_i & 0 & \ldots & 0\\
\mathbf{C}_i\mathbf{A}_i\mathbf{D}_i & \mathbf{C}_i\mathbf{D}_i & \ddots & \vdots\\
\vdots & \vdots & \vdots & 0\\
\mathbf{C}_i\mathbf{A}_i^{N_p-1}\mathbf{D}_i & \mathbf{C}_i\mathbf{A}_i^{N_p-2}\mathbf{D}_i & \ldots & \mathbf{C}_i\mathbf{A}_i^{N_p-N_c}\mathbf{D}_i
\end{bmatrix}. \tag{16}$$

Note that in (11), $\boldsymbol{\theta}_i(t)$ is used as the control input and the input sampling matrix $\boldsymbol{\Psi}_j(t)$ is embedded in (15). In practice, the current state $\mathbf{x}_i(t)$ is usually not available from measurements and a state observer needs to be used to reconstruct the full state vector. In this case, we replace $\mathbf{x}_i(t)$ by its estimate $\widehat{\mathbf{x}}_i(t)$, in (13) and also, replace $\mathbf{Y}_i(t)$ by $\widehat{\mathbf{Y}}_i(t)$ in (11).

Problem (11)–(13) is a quadratic programming problem which can be solved efficiently and reliably using standard off-the-shelf solvers. The Nash-based MPC algorithm for solving the control problem proceeds by allowing each subsystem/agent to optimise its objective function using its own control decision $\boldsymbol{\vartheta}_i(t)$ assuming that other subsystems' solutions $\boldsymbol{\vartheta}_j(t)$ are known. Let $\boldsymbol{\vartheta}_i^n(t)$ define the computed control input for subsystem i at iteration n, $(n \ge 0)$. At each sampling time each agent makes an initial guess of its decision variables over the control horizon and broadcasts that to other agents:

$$\boldsymbol{\theta}_i^n(t) = [\boldsymbol{\vartheta}_i^n(t)\;\; \boldsymbol{\vartheta}_i^n(t+1), \ldots, \boldsymbol{\vartheta}_i^n(t+N_c-1)]^T, \tag{17}$$

Then, each agent solves its optimization problem (11)–(13) and gets its optimal solution $\boldsymbol{\theta}_i^{n+1}(t)$. Next, all the agents compare the new solution $\boldsymbol{\theta}_i^{n+1}(t)$ with the solution obtained at the previous iteration $\boldsymbol{\theta}_i^n(t)$ and check the convergence condition:

$$||\boldsymbol{\theta}_i^{n+1}(t) - \boldsymbol{\theta}_i^n(t)||_\infty \le \boldsymbol{\epsilon}, \tag{18}$$

in which $\boldsymbol{\epsilon}$ is the error accuracy. If the Nash optimal solution has been achieved, each subsystem does not change its decision $\boldsymbol{\theta}_i^n(t)$ because it has achieved an equilibrium point of the coupling decision process [5], [8]; otherwise the local cost function $J_i(t)$ will degrade. In the following section, a novel distributed Kalman Filter algorithm is proposed to provide optimal estimation $\widehat{\mathbf{x}}_i(t)$ of the state vector $\mathbf{x}_i(t)$ while compensating for the inter-sampling information loss due to the multi-rate nature of the systems under study.

III Distributed multi-Rate Kalman Filter

Consider the linear model in (9)–(10). We want to use the available measurements $\boldsymbol{\varphi}_i$ to estimate the state of the system $\mathbf{x}_i$. We propose a linear optimal filter which is based on Kalman Filter for distributed systems. To understand the distributed Kalman Filter equations, let us consider the process noise $\mathbf{v}_i(t)$ to be discrete-time white noise for each subsystem i. The following covariance matrix for each agent can hence be defined:

$$\mathbb{E}\{v_i(t)v_i^T(t)\} = S_{p_i}(t) \tag{19}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator and $S_{p_i}(t)$ represents the covariance matrix of the process noise. Consider measurement noise $z_i(t)$ in (10) to be discrete-time white noise, the following covariance matrix for the measurement noise $S_{m_i}(t)$ can be defined similarly:

$$\mathbb{E}\{z_i(t)z_i^T(t)\} = S_{m_i}(t). \tag{20}$$

Let the states estimated by the distributed Kalman Filter for a multi-rate system be given by:

$$\begin{aligned}\widehat{\mathbf{x}}_i(t+1|t) &= \mathbf{A}_i\widehat{\mathbf{x}}_i(t|t-1) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(t)\\ &+ \mathbf{L}_i(t)[\boldsymbol{\varphi}_i(t) - \boldsymbol{\Upsilon}_i(t)\mathbf{C}_i\widehat{\mathbf{x}}_i(t|t-1)]\\ &+ \sum_{\substack{j=1\\ j\neq i}}^{m}\big[\mathbf{B}_{ij}\Delta\mathbf{u}_j(t)\\ &+ \mathbf{L}_j(t)[\boldsymbol{\varphi}_j(t) - \boldsymbol{\Upsilon}_j(t)\mathbf{C}_j\widehat{\mathbf{x}}_j(t|t-1)]\big],\end{aligned} \tag{21}$$

where the terms $\mathbf{L}_i(t)$ and $\mathbf{L}_j(t)$ are referred to as the *Kalman Gains*. Variable $\mathbf{L}_j(t)$ is the Kalman gain which is made by neighbouring agents and can be different from $\mathbf{L}_i(t)$. From (21) it is clear that local estimators share their gains and also estimated states to accomplish their estimation task. Substituting (10) into (21) and subtracting that from (9) we proceed to the next step to obtain the estimation error $\mathbf{e}_i(t+1|t) = \mathbf{x}_i(t+1|t) - \widehat{\mathbf{x}}_i(t+1|t)$ at sampling time t. The index $(t|t-1)$ refers to the information at sampling time t given knowledge of the process prior to sampling time t. Therefore,

$$\begin{aligned}\mathbf{e}_i(t+1|t) &= [\mathbf{A}_i - \mathbf{L}_i(t)\boldsymbol{\Upsilon}_i(t)\mathbf{C}_i]\mathbf{e}_i(t|t-1)\\ &+ \mathbf{D}_i\mathbf{v}_i(t) - \mathbf{L}_i(t)\boldsymbol{\Upsilon}_i(t)\mathbf{z}_i(t)\\ &- \sum_{\substack{j=1\\ j\neq i}}^{m}\big(\mathbf{L}_j(t)\boldsymbol{\Upsilon}_j(t)\mathbf{C}_j\mathbf{e}_j(t|t-1)\\ &+ \mathbf{L}_j(t)\boldsymbol{\Upsilon}_j(t)\mathbf{z}_j(t)\big).\end{aligned} \tag{22}$$

Now to initialize the estimator algorithm, consider $\mathbb{E}[\mathbf{x}_i(0|-1)] = \widehat{\mathbf{x}}_i(0|-1)$ then $\mathbb{E}[\mathbf{e}_i(t|t-1)] = \mathbf{0}$, $\forall t$. This means we assume that the mean of the estimates should be equal to the mean of the expected value in the Kalman Filter design. In order to develop the Kalman Filter for the multi-rate and distributed case we define a covariance matrix $\mathbf{S}_i(t+1) = \mathbb{E}\{\mathbf{e}_i(t+1|t)\mathbf{e}_i^T(t+1|t)\}$. By the properties of the vector covariance and expansion of the terms [6] we obtain the final form of the multi-rate distributed Kalman Filter as:

$$\begin{aligned}\mathbf{S}_i(t+1) &= \mathbf{A}_i\mathbf{S}_i(t)\mathbf{A}_i^T + \mathbf{D}_i\,\mathbf{S}_{p_i}(t)\,\mathbf{D}_i^T\\ &- \mathbf{A}_i\mathbf{S}_i(t)\mathbf{C}_i^T\boldsymbol{\Upsilon}_i(t)\boldsymbol{\Omega}_i^{-1}(t)\boldsymbol{\Upsilon}_i(t)\mathbf{C}_i\mathbf{S}_i(t)\mathbf{A}_i^T\\ &+ \sum_{\substack{j=1\\ j\neq i}}^{m}\mathbf{A}_j\mathbf{S}_i(t)\mathbf{C}_j^T\boldsymbol{\Upsilon}_j(t)\boldsymbol{\Omega}_j^{-1}(t)\,\boldsymbol{\Upsilon}_j(t)\mathbf{C}_j\mathbf{S}_j(t)\mathbf{A}_j^T,\end{aligned} \tag{23}$$

with $\boldsymbol{\Omega}_i(t)$ and $\boldsymbol{\Omega}_j(t)$ positive definite and defined as:

$$\begin{aligned}\boldsymbol{\Omega}_i &= \boldsymbol{\Upsilon}_i(t)\mathbf{C}_i\mathbf{S}_i(t)\mathbf{C}_i^T\boldsymbol{\Upsilon}_i(t) + \boldsymbol{\Upsilon}_i(t)\mathbf{S}_{m_i}(t)\boldsymbol{\Upsilon}_i(t)\\ &+ [\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_i(t)]\end{aligned} \tag{24}$$

$$\begin{aligned}\boldsymbol{\Omega}_j &= \boldsymbol{\Upsilon}_j(t)\mathbf{C}_j\mathbf{S}_j(t)\mathbf{C}_j^T\boldsymbol{\Upsilon}_j(t) + \boldsymbol{\Upsilon}_j(t)\mathbf{S}_{m_j}(t)\boldsymbol{\Upsilon}_j(t)\\ &+ [\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_j(t)].\end{aligned} \tag{25}$$

It should be noted that (23) is an algebraic Ricatti equation. The solution of the Ricatti equation is found iteratively backwards in time by using (24) and (25). Then, the Kalman gains are computed as:

$$\mathbf{L}_i(t) = \mathbf{A}_i\mathbf{S}_i(t)\mathbf{C}_i^T\boldsymbol{\Upsilon}_i(t)\boldsymbol{\Omega}_i^{-1}(t), \tag{26}$$

$$\mathbf{L}_j(t) = \mathbf{A}_j\mathbf{S}_i(t)\mathbf{C}_j^T\boldsymbol{\Upsilon}_j(t)\boldsymbol{\Omega}_j^{-1}(t). \tag{27}$$

The reason for adding $[\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_i(t)]$ and $[\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_j(t)]$ to (24) and (25), respectively, is that in the process of inverting $\boldsymbol{\Omega}_i^{-1}(t)$ and $\boldsymbol{\Omega}_j^{-1}(t)$ for the Kalman gain equations (26) and (27), singularity may occur at those sampling times for which we have no output measurements ($\boldsymbol{\Upsilon}_i(t) = 0$). Thus, to guarantee the non-singularity of $\boldsymbol{\Omega}_i^{-1}(t)$ and $\boldsymbol{\Omega}_j^{-1}(t)$ the extra terms $[\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_i(t)]$ and $[\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}_j(t)]$ have been added to (24) and (25), respectively, in which $\mathbf{I}_{q\times q}$ is $q$by$q$ identity matrix, [6]. The block diagonal matrix $[\mathbf{I}_{q\times q} - \boldsymbol{\Upsilon}(t)]$ only adds non-zero terms to

the scalar diagonal elements of $\mathbf{\Omega}_i(t)$ and $\mathbf{\Omega}_j(t)$ during the output sampling mechanism and in no way affects the predictor equation.

IV Simulation Results

In this section the proposed method has been analyzed for a multi-rate large-scale system through simulation. The system under control is a high purity distillation column studied as a benchmark for large-scale systems [4]. The outputs of the system, y_1 and y_2, are top and bottom product compositions, respectively, and the inputs u_1 and u_2 are reflux flow-rate and boil-up, respectively. As the composition dynamics in this system are usually much slower than the flow dynamics, the system can be considered as a multi-rate system. The inputs are constrained to $-2 \leq u_1 \leq 2$ and $-1 \leq u_2 \leq 2$; there are no constraints on the outputs. The weighting matrices for the outputs and inputs are $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$, respectively. The set-point value for the first subsystem switches between one and zero every 200 minutes and for the second subsystem the set-point is zero. The nominal model is decomposed into two subsystems as follows:

Subsystem 1:

$$\dot{x}_1 = -0.0133x_1 + 0.0117u_1 + 0.0115u_2 \quad (28)$$

$$y_1 = x_1 \quad (29)$$

Subsystem 2:

$$\dot{x}_2 = -0.0133x_2 + 0.0146u_1 + 0.0144u_2 \quad (30)$$

$$y_2 = x_2. \quad (31)$$

The continuous-time model is discretized with a sampling time of 1 min. The process and measurement noises for both subsystems are zero mean white noise sequences with covariances $\boldsymbol{S}_{p_1}(t) = \boldsymbol{S}_{p_2}(t) = \boldsymbol{S}_{m_1}(t) = \boldsymbol{S}_{m_2}(t) = 10^{-5}$. In Fig. 2, the effect of changing prediction and control horizons on the tracking of the proposed method is illustrated. In Fig. 2a, the proposed method has been simulated over various control horizons and the prediction horizon is kept as $N_p = 20$. As it can be seen from Fig. 2a, the control horiozn does not affect the set-point tracking significantly. In Fig. 2b, the proposed method has been simulated over different prediction horizons and the control horizon is kept as $N_c = 5$. From Fig. 2b, it is clear that the longer the prediction horizon, the better set-point tracking.

In Fig. 3 the number of iterations in the proposed method has been ploted againts simulation time. In Fig. 3a, the prediction horizon is $N_p = 20$ and the proposed method simulated over different control horizons. From Fig. 3a, it can be observed that the number of iterations needed for convergence is more or less the same in different control horizons, whereas from Fig. 3b it can be observed that the larger predictioction horizon needs a larger number of iterations to converge. Since from Fig. 2a and Fig. 3a, changing the control horizon does not affect the system's performance significantly, we computed the elapsed CPU time for computation of optimal inputs at different control horizons while the prediction horizon was fixed. As is clear from Table1, a shorter control horizon requires a shorter simulation time.

Table 1: The elapsed CPU computation time for optimal input at different control horizons in the proposed method.

Control Horizon (N_c)	CPU Elapsed Time (sec)
20	0.0045
15	0.0030
10	0.0020
5	0.0012

V Conclusions And Future Research

From the simulation results, it can be concluded that for the proposed distributed multi-rate controller, selection of large prediction horizon and short control horizon drives the system toward a desired tracking performance. The presented method uses a communication-based optimization based on Nash Equilibrium (NE), which is non-cooperative. However, the best achievable performance is characterized by a Pareto set which represents the set of optimal trade-offs among the competing controller objectives [3]. Further research is required to extend the proposed method for cooperative situations. Also, further research needs to be done toward analysing different aspects of the proposed method such as stability and convergence.

VI Acknowledgment

This research is supported by the Irish Programme for Research in Third Level Institutions (Cycle 4) and VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" of the Dutch Technology Foundation STW, a subdivision of the Netherlands Organisation for Scientific Research (NWO).

References

[1] S. Roshany-Yamchi, R. R. Negenborn, M. Cychowski, B. De Schutter, J, Connell and K. Delaney. "Distributed Model Predictive Control and Estimation of Large-Scale Multi-Rate Systems". *18th IFAC World Congress*, Milan, Italy, 28 Aug – 2 Sep, 2011 (Accepted).

[2] J. M. Maciejowski. " Predictive Control with Constraints". *Prentice Hall*, England, 2002.

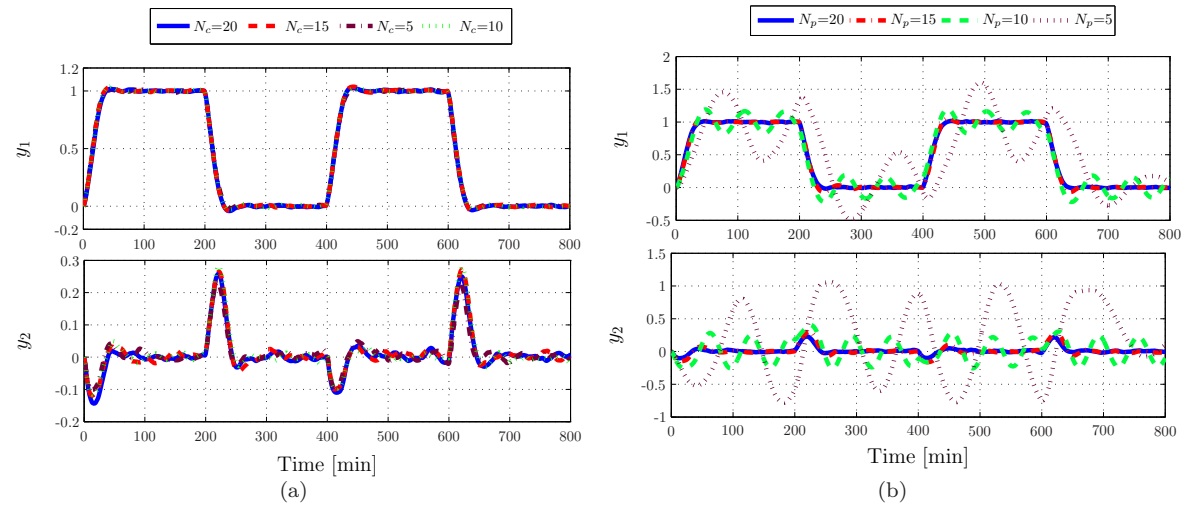

Fig. 2: Closed-loop response of asynchronous agents ($T_{u_1} = 1, T_{u_2} = 2, T_{y_1} = 3, T_{y_2} = 4$), at different control horizons (a) and different prediction horizons (b).

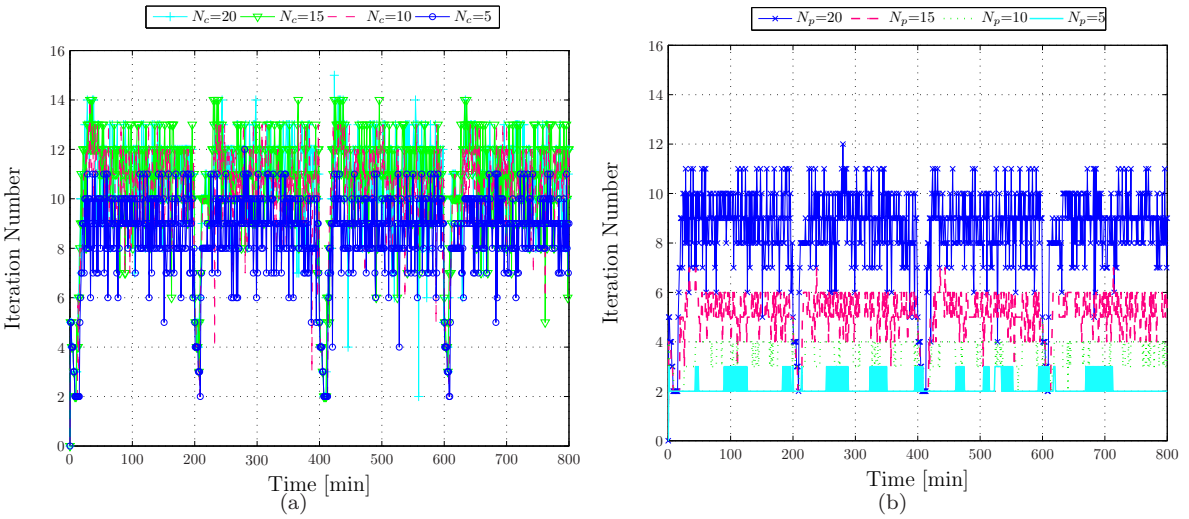

Fig. 3: Comparison of required iteration numbers of the proposed method at different control horizons (a) and different prediction horizons (b).

[3] A. N. Venkat, J. B. Rawlings and S. J. Wright. "Implementable distributed model predictive control with guaranteed performance properties". *In Proc. of American Control Conference*, Minneapolis, MN, 2006, pp. 813–618.

[4] W. Al-Gherwi and H. Budman and A. Elkamel. "Selection of control structures for distributed model predictive control in the presence of model errors". *Process Control*, 20:270–284, 2010.

[5] S. Li and Y. Zhang and Q. Zhu. "Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem". *Information Sciences*, 170:329–349, 2005.

[6] R. Scattolini and N. Schiavoni. "A Multi-Rate Model Based Predictive Control". *IEEE Transactions on Automatic Control*, 40:1093–1097, 1995.

[7] J. H. Lee and M. S. Gelormino and M. Morari. "Model predictive control of multi-rate sampled-data systems: a state-space approach". *International Journal of Control*, 55:153–191, 1992.

[8] L. Giovanini and J. Balderud. "Game approach to distributed model predictive control". *In International Control Conference*, Glasgow, UK, December 2006.