# LIMITED-COMMUNICATION DISTRIBUTED MODEL PREDICTIVE CONTROL FOR HVAC SYSTEMS

A Dissertation

by

RAWAND EHSAN JALAL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | Bryan Rasmussen |
| Committee Members, | Raktim Bhattacharya |
| | Won-Jong Kim |
| | Yong-Joe Kim |
| Head of Department, | Andreas A. Polycarpou |

December 2016

Major Subject: Mechanical Engineering

## ABSTRACT

This dissertation proposes a Limited-Communication Distributed Model Predictive Control algorithm for networks with constrained discrete-time linear processes as local subsystems. The introduced algorithm has an iterative and cooperative framework with neighbor-to-neighbor communication structure. Convergence to a centralized solution is guaranteed by requiring coupled subsystems with local information to cooperate only. During an iteration, a local controller exchanges its predicted effects with local neighbors (which are treated as measured input disturbances in local dynamics) and receives the neighbor sensitivities for these effects at next iteration. Then the controller minimizes a local cost function that counts for the future effects to neighbors weighted by the received sensitivity information. Distributed observers are employed to estimate local states through local input-output signals. Closed-loop stability is proved for sufficiently long horizons. To reduce the computational loads associated with large horizons, local decisions are parametrized by Laguerre functions. A local agent can also reduce the communication burden by parametrizing the communicated data with Laguerre sequences.

So far, convergence and closed-loop stability of the algorithm are proven under the assumptions of accessing all subsystem dynamics and cost functions information by a centralized monitor and sufficient number of iterations per sampling. However, these are not mild assumptions for many applications. To design a local convergence condition or a global condition that requires less information, tools from dissipativity theory are used.

Although they are conservative conditions, the algorithm convergence can now be ensured either by requiring a distributed subsystem to show dissipativity in the local information dynamic inputs-outputs with gain less than unity or solving a global dissipative inequality with subsystem dissipativity gains and network topology only. Free variables are added to the local problems with the object of having freedom to design such convergence conditions. However, these new variables will result into a suboptimal algorithm that affects the proposed closed-loop stability. To ensure local MPC stability, therefore, a distributed synthesis, which considers the system interactions, of stabilizing terminal costs is introduced. Finally, to illustrate the aspects of the algorithm, coupled tank process and building HVAC system are used as application examples.

## ACKNOWLEDGEMENTS

First of all I would like to thank my advisor, Professor Bryan Rasmussen, for having given me the opportunity not only to spend such a good time in Texas A&M University but also to live the like the native American people. Last four years have been the most excitement and productive period of my entire life.

I also thank my committee members Professors: Raktim Bhattacharya, Won-Jong Kim, and Yong-Joe Kim for serving in my committee. A special thanks goes to Professor Raktim Bhattacharya whose door was always open for me.

Thanks to my colleagues and friends in the Thermo-Fluids Control Library. Thanks Chris Price, Chris Bay, Rohit, Traver, Chaw, Kaimi, and Austin for always having been real and honest friends. You cannot image how hard is for me to leave you with the intention that I might not see you again in the future. I was really lucky to have you all in my life.

Thanks to everyone in the Higher Committee for Education Development (HCED) in Iraq for offering me this great scholarship and facilitating the obstacles. I Also would like to thank the National Institute of Standards and Technology (NIST) and the US Dept. of Commerce for partially funning my project and work.

Thanks Aram Raheem for being such a wonderful friend. Thank you for helping me all the time and giving me hope when I miss it.

I leave a very special and deep thanks for my mom who always believed in me and encouraged me. Mom I would not be in this position without you. Thanks Dad for all what you have done for me. I love you more than I can show.

At the end, there is a person who owns all what I have now and in the future. Hawree my beautiful wife who supported me in every difficulty I have been in with a nice laughter on her face that says: Hey you can do it. Thanks Hawree for everything and thank you for being in my life.

## NOMENCLATURE

| | |
|---|---|
| $\alpha_i$ | Local design variable for the suboptimal LC-DMPC algorithm |
| $\boldsymbol{\alpha}$ | Block diagonal matrix of $\alpha_i$ for all subsystems in the network |
| $\beta$ | Convex combination scalar constant for all subsystems in the network |
| $\gamma_i$ | cost sensitivities w.r.t. $V_i$ for subsystem $\Sigma_i$ |
| $\boldsymbol{\gamma}$ | Stacked vector of $\gamma_i$for all subsystems in the network |
| $\boldsymbol{\Gamma}$ | Interconnecting matrix with values of 0 or 1 |
| $\Gamma_1$ | Interconnecting matrix for $N_p = 1$ |
| $\delta$ | Flow parameters used in the six-tank process |
| $\delta_{fan}$ | Variable for the AHU optimizer economic cost function to produce optimum $P_{EDS}$ set-point |
| $\delta_{T_{AHU}}$ | Variable for the AHU optimizer economic cost function to produce optimum discharge air temperature set-point |
| $\varepsilon_i$ | Measurement noise for subsystem $\Sigma_i$ |
| $\mathcal{E}$ | Solution of the stabilizing Lyapunov function – suboptimal LC-DMPC algorithm |
| $\varrho_i$ | Damper position for zone $i$ (%) |
| $\zeta_i$ | Local mapping matrix for terminal stabilizing cost with values of 0 and 1 |
| $\ell_i$ | Lagrange multiplier of the local defined optimization problem |
| $\boldsymbol{\ell}$ | Lagrange multiplier of the systemwide defined optimization problem |
| $\lambda$ | Flow parameters used in the six-tank process |
| $\mu_i$ | Local design variable for the suboptimal LC-DMPC algorithm |
| $\boldsymbol{\mu}$ | Block diagonal matrix of $\mu_i$ for all subsystems in the network |
| $\aleph, i$ | Includes subsystem $\Sigma_i$ and coupled downstream neighbors, state, weight, stabilizing terminal weight, … |
| $\rho$ | Flow parameters used in the six-tank process |
| $\sigma_i(t)$ | Inlet flow rate for tank $i$ for the six-tank process $(cm^3/sec)$ |

| | |
|---|---|
| $\sigma$ | Flow parameters used in the six-tank process |
| $\Sigma_i$ | A subsystem in the network |
| $\tau$ | Flow parameters used in the six-tank process |
| $\psi_i$ | Downstream neighbor cost sensitivities as seen by subsystem $\Sigma_i$ |
| $\Psi_i$ | Predicted values of $\psi_i$ along $N_p$ |
| $\boldsymbol{\Psi}$ | Stacked vector of $\Psi_i$ for all subsystems in the network |
| $\bar{\boldsymbol{\Psi}}$ | Steady-state point per sampling for $\boldsymbol{\Psi}$ according to the LC-DMPC algorithm |
| $\mho_i$ | Local mapping matrix for terminal stabilizing cost with values of 0 and 1 |
| $\mathfrak{H}_{i_{k\|k}}$ | Covariance for $\tilde{x}_{i_{k\|k}}$ for subsystem $\Sigma_i$ |
| $a_j$ | Pole location of discrete Laguerre functions |
| $A_i$ | State matrix for subsystem $\Sigma_i$ |
| $A$ | Block diagonal matrix of $A_i$ for all subsystems in the network |
| $\mathring{A}_i$ | Cross sectional area of the tanks $(cm^2)$ in the six-tank process |
| $\hat{A}_i$ | State matrix for a local information dynamics |
| $b_i$ | Tank cross section of the outlet hole $(cm^2)$ in the six-tank process |
| $B_{u,i}$ | Input control matrix for subsystem $\Sigma_i$ |
| $B_{v,i}$ | Input measured disturbance matrix for subsystem $\Sigma_i$ |
| $B_{d,i}$ | Input unmeasured disturbance matrix for subsystem $\Sigma_i$ |
| $B_u$ | Block diagonal matrix of $B_{u,i}$ for all subsystems in the network |
| $B_d$ | Block diagonal matrix of $B_{d,i}$ for all subsystems in the network |
| $\hat{B}_i$ | Input control matrix for a local information dynamics |
| $\mathcal{B}$ | Flow parameters used in the six-tank process |
| $c_j$ | Coefficients of discrete Laguerre functions |
| $cfm_i$ | Damper inlet flow rate for zone $i$ $(ft^3/min)$ |
| $cfm_{total}$ | Total flow rate across all zones $(ft^3/min)$ |
| $C_{y,i}$ | Regulated output matrix for subsystem $\Sigma_i$ |
| $C_{z,i}$ | Downstream output matrix for subsystem $\Sigma_i$ |

| | |
|---|---|
| $C$ | Block diagonal matrix of $C_{y,i}$ for all subsystems in the network |
| $\hat{C}_i$ | Output matrix for a local information dynamics |
| $D_{y,i}$ | Carry through matrix of the regulated output for subsystem $\Sigma_i$ |
| $D_{z,i}$ | Carry through matrix of the downstream output for subsystem $\Sigma_i$ |
| $d_i$ | Unmeasured disturbances for subsystem $\Sigma_i$ |
| $\mathrm{D}_i$ | Predicted values of $d_i$ along $N_p$ |
| $\mathbf{D}$ | Stacked vector of $\mathrm{D}_i$ for all subsystems in the network |
| $e_i$ | $r_i - y_i$ for subsystem $\Sigma_i$ at $k$ |
| $\mathrm{e}_i$ | $\mathrm{r}_i - \mathrm{Y}_i$ along $N_p$ for subsystem $\Sigma_i$ |
| $e$ | Stacked vector of $e_i$ for all subsystems in the network |
| $\mathbf{e}$ | Stacked vector of $\mathrm{e}_i$ for all subsystems in the network |
| $F_{y,i}$ | Matrix for predicted regulated outputs with $x_{0,i}$ for subsystem $\Sigma_i$ |
| $F_{z,i}$ | Matrix for predicted downstream outputs with $x_{0,i}$ for subsystem $\Sigma_i$ |
| $\mathbf{F_y}$ | Block diagonal matrix of $F_{y,i}$ for all subsystems in the network |
| $\mathbf{F_z}$ | Block diagonal matrix of $F_{z,i}$ for all subsystems in the network |
| $g$ | Gravity constant $(cm/sec^2)$ in the six-tank process |
| $h_i(t)$ | Water height for tank $i$ for the six-tank process $(cm)$ |
| $H$ | Network topology |
| $I_{cl,i}$ | Clothing insulation for zone $i$ $(m^2\mathrm{K/W})$ |
| $I_n$ | Identity matrix with dimension $n$ |
| $k$ | Current discrete time step |
| $k_c$ | Tank level sensor accuracy $(\mathrm{V}/cm)$ in the six-tank process |
| $k_i$ | Voltage constant for pumps $(cm^3/\mathrm{V}\cdot sec)$ in the six-tank process |
| $M_i$ | Metabolic rate for zone $i$ $(\mathrm{W}/m^2)$ |
| $M_{y,i}$ | Matrix for predicted regulated outputs with $\mathrm{U}_i$ for subsystem $\Sigma_i$ |
| $M_{z,i}$ | Matrix for predicted downstream outputs with $\mathrm{U}_i$ for subsystem $\Sigma_i$ |
| $\mathbf{M_y}$ | Block diagonal matrix of $M_{y,i}$ for all subsystems in the network |
| $\mathbf{M_z}$ | Block diagonal matrix of $M_{z,i}$ for all subsystems in the network |

| | |
|---|---|
| $N_{c,i}$ | Constraint horizon of the optimization problem with Laguerre functions |
| $n_i$ | Number of states for subsystem $\Sigma_i$ |
| $N_i$ | Number of discrete Laguerre functions |
| $N_p$ | Length of the state prediction horizon |
| $N_{y,i}$ | Matrix for predicted regulated outputs with $\mathrm{V}_i$ for subsystem $\Sigma_i$ |
| $N_{z,i}$ | Matrix for predicted downstream outputs with $\mathrm{V}_i$ for subsystem $\Sigma_i$ |
| $\mathbf{N_y}$ | Block diagonal matrix of $N_{y,i}$ for all subsystems in the network |
| $\mathbf{N_z}$ | Block diagonal matrix of $N_{z,i}$ for all subsystems in the network |
| $\mathbb{N}_i$ | Number of iterations |
| $p$ | Total number of subsystems in the network |
| $P_{EDS}$ | End static pressure $(in.water)$ |
| $p_{y,i}$ | Number of regulated outputs for subsystem $\Sigma_i$ |
| $p_{z,i}$ | Number of outputs for downstream neighbors for subsystem $\Sigma_i$ |
| $p_y$ | Total number of regulated outputs for the network |
| $p_z$ | Total number of outputs for downstream neighbors for the network |
| $P_{y,i}$ | Matrix for predicted regulated outputs with $\mathrm{D}_i$ for subsystem $\Sigma_i$ |
| $P_{z,i}$ | Matrix for predicted downstream outputs with $\mathrm{D}_i$ for subsystem $\Sigma_i$ |
| $\mathbf{P_y}$ | Block diagonal matrix of $P_{y,i}$ for all subsystems in the network |
| $\mathbf{P_z}$ | Block diagonal matrix of $P_{z,i}$ for all subsystems in the network |
| $\hat{P}_i$ | Weighting matrix used in the Lyapunov function for a local dissipativity inequality |
| $p_{year}$ | Yearly payment for zone $i$ (\$) |
| $q_i$ | Positive definite weighting matrix for errors for subsystem $\Sigma_i$ |
| $\mathrm{Q}_i$ | Block diagonal matrix of $q_i$ along $N_p$ for subsystem $\Sigma_i$ |
| $Q$ | Block diagonal matrix of $q_i$ for all subsystems in the network |
| $\mathbf{Q}$ | Block diagonal matrix of $\mathrm{Q}_i$ for all subsystems in the network |
| $Q_{d,i}$ | Output weighting matrix for a local supply rate function |
| $Q_{r,i}$ | Weighting matrix for computing reachable references |

| | |
|---|---|
| $Q_{v,i}$ | Covariance for $\varepsilon_i$ |
| $Q_{w,i}$ | Covariance for $w_i$ |
| $r_i$ | Desired tracking reference at $k$ for subsystem $\Sigma_i$ |
| $\mathrm{r}_i$ | Desired tracking reference along $N_p$ for subsystem $\Sigma_i$ |
| $\mathbf{r}$ | Stacked vector of $\mathrm{r}_i$ for all subsystems in the network |
| $r_{d,i}$ | Number of unmeasured input disturbances for subsystem $\Sigma_i$ |
| $R_{d,i}$ | Input weighting matrix for a local supply rate function |
| $r_{u,i}$ | Number of controlled inputs for subsystem $\Sigma_i$ |
| $r_{v,i}$ | Number of measured inputs from upstream neighbors for subsystem $\Sigma_i$ |
| $r_v$ | Total number of measured input disturbances for the network |
| $s_i$ | Positive definite weighting matrix for control actions $u_i$ for subsystem $\Sigma_i$ |
| $\mathrm{s}_i$ | Local supply rate function |
| $\mathrm{S}_i$ | Block diagonal matrix of $s_i$ along $N_p$ for subsystem $\Sigma_i$ |
| $S$ | Block diagonal matrix of $s_i$ for all subsystems in the network |
| $\mathbf{S}$ | Block diagonal matrix of $\mathrm{S}_i$ for all subsystems in the network |
| $S_{d,i}$ | Output-input weighting matrix for a local supply rate function |
| $T_{AHU}$ | AHU discharge air temperature (°F) |
| TCD | AHU total cooling demand |
| $T_{out}$ | Outside temperature (°F) |
| $\bar{T}_r$ | Mean radiant temperature (°C) |
| $u$ | Stacked vector of $u_i$ for all subsystems in the network |
| $u_i$ | Input control vector for subsystem $\Sigma_i$ |
| $\mathrm{U}_i$ | Predicted values of $u_i$ along $N_p$ |
| $\mathbf{U}$ | Stacked vector of $\mathrm{U}_i$ for all subsystems in the network |
| $\bar{\mathbf{U}}$ | Steady-state point per sampling for $\mathbf{U}$ according to the LC-DMPC algorithm |
| ${\mathrm{U}_i}^{QP}$ | Solution of the local QP control optimization problem |
| $\mathbf{U^{QP}}$ | Stacked vector of ${\mathrm{U}_i}^{QP}$ for all subsystems in the network |

| | |
|---|---|
| $u_{min,i}$ | Minimum limit of the local control action $u_i$ |
| $u_{max,i}$ | Maximum limit of the local control action $u_i$ |
| $u_{min}$ | Stacked vector of $u_{min,i}$ for all subsystems in the network |
| $u_{max}$ | Stacked vector of $u_{max,i}$ for all subsystems in the network |
| $u_{r,i}$ | Reachable control vector for subsystem $\Sigma_i$ |
| $U_{r,i}$ | Stacked vector of $u_{r,i}$ along $N_p$ for subsystem $\Sigma_i$ |
| $V_{ar,i}$ | Relative air velocity for zone $i$ $(m^2/sec)$ |
| $v_i$ | Measured input disturbances from upstream neighbors for subsystem $\Sigma_i$ |
| $\mathrm{V}_i$ | Predicted values of $v_i$ along $N_p$ |
| $V_i$ | Lyapunov function for a local dissipativity inequality |
| $\mathbf{V}$ | Stacked vector of $\mathrm{V}_i$ for all subsystems in the network |
| $\overline{\mathbf{V}}$ | Steady-state point per sampling for $\mathbf{V}$ according to the LC-DMPC algorithm |
| $w_i$ | Process noise for subsystem $\Sigma_i$ |
| $W_i$ | Effective mechanical power for zone $i$ $(\mathrm{W}/m^2)$ |
| $x_i$ | State vector for subsystem $\Sigma_i$ |
| $x_{0,i}$ | Initial condition of the state vector for subsystem $\Sigma_i$ |
| $\mathbf{X_0}$ | Stacked vector of $x_{0,i}$ for all subsystems in the network |
| $x_{r,i}$ | Reachable state vector for subsystem $\Sigma_i$ |
| $X_{r,i}$ | Stacked vector of $x_{r,i}$ along $N_p$ for subsystem $\Sigma_i$ |
| $\tilde{x}_{i_{k\|j}}$ | Estimation of $x_i$ at time $k$ based on the information in time $j$, $k \geq j$ |
| $y_i$ | Regulated output vector for subsystem $\Sigma_i$ |
| $\mathrm{Y}_i$ | Predicted values of $y_i$ along $N_p$ |
| $\mathbf{Y}$ | Stacked vector of $\mathrm{Y}_i$ for all subsystems in the network |
| $z_i$ | Output disturbances for downstream neighbors for subsystem $\Sigma_i$ |
| $\mathrm{Z}_i$ | Predicted values of $z_i$ along $N_p$ |
| $\mathbf{Z}$ | Stacked vector of $\mathrm{Z}_i$ for all subsystems in the network |
| $\mathbb{Z}_i$ | Local productivity for the economic zone cost function |

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 1. INTRODUCTION AND LITERATURE REVIEW

## 1.1 Introduction

Building operations account for approximately 23% of the total energy consumption in the European Union [1] and 41% of US energy usage and carbon emissions [2], [3]. Moreover, the competitive economic and energy costs are additional motives to develop practical approaches for improving the energy efficiency in commercial buildings as well as industrial facilities.

Heating, Ventilation and Air-Conditioning (HVAC) systems consume about 50% of the energy in buildings and approximately 20% of the total energy consumption in Europe [4]. On the other hand, according to systematic building management estimations, HVAC systems are consuming more energy than expected and there are potential energy savings between 5% to 30% [5], [6]. This justifies the different efforts and methods to increase the efficiency of the HVAC systems around the world. Many advanced control strategies have been suggested for controlling HVAC systems with the goal of minimizing the energy consumptions and keeping the indoor comfort.

HVAC systems introduce one of the most challenging problems to deal with from a control point of view. Energy demands that fluctuate from season to season or even from day to day, together with a complex combination of human requirements, contribute to a highly non-stable environment for control systems to be designed.

In commercial buildings, the HVAC system is a set of interconnected devices, and actuators that are devoted to the conditioning of the air inside these buildings, i.e. to

control the levels of temperature, humidity, and $CO_2$. Usually these different components are managed through a centralized monitoring supervisor with widely distributed sensors and independent controllers. Figure 1.1 illustrates a typical HVAC system in buildings. Mainly there are two parts in such systems: the water cooling and heating system which involves boilers, chillers, and cooling towers and the air conditioning and ventilation system which includes Air Handling Units (AHU), Variable Air Volume (VAV) boxes, and ducts. In addition, there is a network of pumps to circulate the water. The cooling and heating systems are utilized to cool or heat the water and to reject the waste heat that the water carries from condensers. The cold or hot water is then pumped into the conditioning and ventilation system. Here, the AHU condition the air through heat exchangers and a network of fans and ducts carry the conditioned air to the desired locations. The VAV boxes control the amount of air that enters the rooms based on the desired temperature setpoints. The chillers are themselves interconnected subsystems consisting of multiple compressors, heat exchangers, and expansion valves.

Low level controllers, particularly proportional-integral controllers, are commonly used in buildings in a completely decentralized fashion. The lack of coordination of such controllers and the nature of the complex interconnected components of HVAC systems may result in inefficient operation. On the other hand, applying a centralized control is impractical to be implemented for large buildings due to the large number of variables and widely distributed units. The centralized controller requires one dynamical model that reveals the behavior of the all HVAC systems components with all state and decision variables. The realization of such a model is challenging and solving one large

problem per sampling is time consuming. Furthermore, the controller needs to communicate all

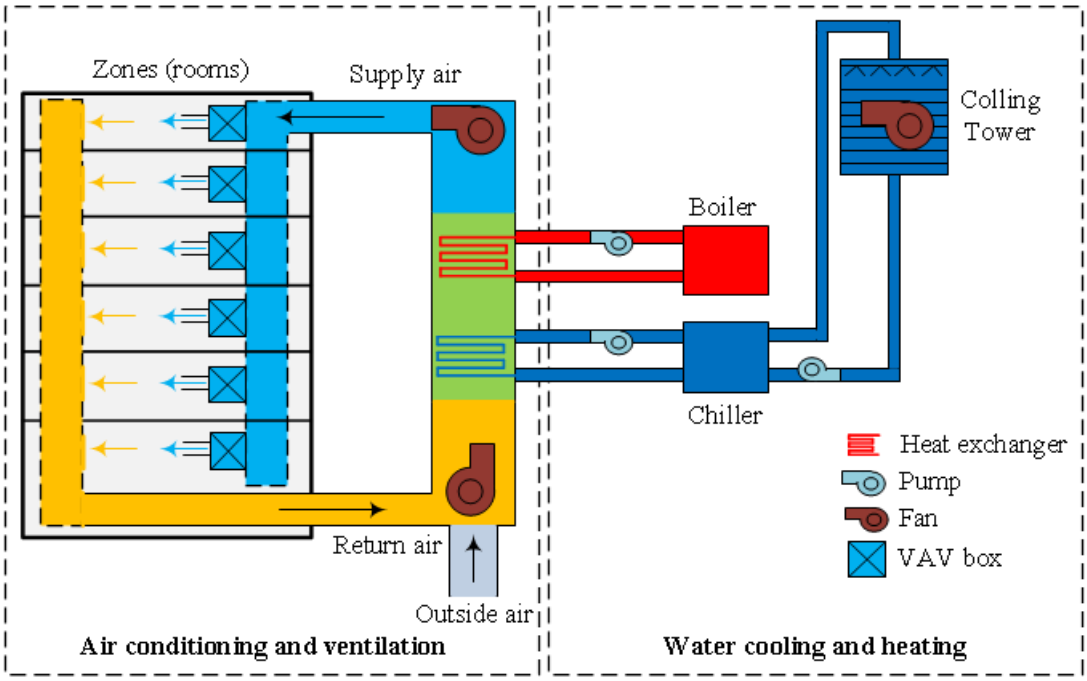



Figure 1.1: A typical HVAC system in commercial buildings

decision variables and collect the feedbacks for the all subsystems which may result in a high demand of communication at each sampling. These challenges could impede the controller to provide real-time decisions within allowed limit or sampling. In addition to the possibility of control failure in case of communication absence, the redesign of the entire controller is required if one of the components is changed or updated. Based on these difficulties, any feasible alternative plantwide control approach is required to have a certain level of modularity and data communication bandwidth. The new control structure needs to have a decentralized design flexibility where only local information is

used by the local agents. At the same time, it has some coordination to approach the centralized control optimum solution and robustness in term of communications.

The problem of the optimal HVAC control can be approached from the perspective of a distributed control. In this structure, the centralized optimization is divided into distributed and cooperative control structures where local decisions are taken based on solving local optimal control problems in coordination toward the centralized optimum point. The network is dealt with as a collection of individual subsystems and each subsystem has a local independent controller that needs some knowledge about the other direct coupling subsystems. This gives an advantage over the decentralized controllers in that it can approach the systemwide optimum through sharing or communicating information.

This dissertation studies the design of a distributed control approach that can reduce the communication load and computational time that a centralized controller may have. The proposed control architecture has a high level of modularity where any change or update in a subsystem does not require changes in its coupled neighbors which meets the control requirement of any HVAC systems. The main approach with convergence and closed-loop stability theories is presented in Chapter II. The design methodology of the local matrices and definition of the local variables for the approach is demonstrated through simulation of coupling tanks process in Chapter III. Chapter IV introduces the suboptimal version of the proposed algorithm where no centralized monitor is needed. With dissipativity theory, the global convergence is ensured locally. New conditions for convergence as well as closed-loop stability are derived. As a real demonstration of the

approach presented in Chapter II, an HVAC system in a real building is simulated and controlled with the application of the algorithm and Chapter V shows the application results.

The remaining sections in this chapter are outlined as follows. First, an overview of Model Predictive Control (MPC) is provided as it is the main control structure used throughout this work. The architecture of MPC is then introduced followed up with various techniques used for guaranteeing the closed-loop stability. In the following sections, several methods are discussed for reparametrizing the optimization problem involved in the MPC along with state observers and dissipativity theory for distributed MPC. The chapter ends with an introduction to the limited-communication distributed model predictive control algorithm. All sections are provided with literature.

## 1.2 Literature Review

### 1.2.1 Model Predictive Control (MPC)

Throughout this work, the main control structure used is Model Predictive Control (MPC). MPC has a well-known history and witnessed many successful applications as one of the fundamental methods of optimal control theory. During the last two decades the application of MPC has received a growing interest for HVAC systems. The popularity of MPC stems in part from its ability to handle physical constraints and multi-variable processes explicitly. At the heart of the MPC is the controlled system model and concept of open-loop optimal feedback. The model is used to predict the future behaviors of the system and compute the optimum control actions along some horizons.

Using current state information, an optimization problem (associated with the future states and manipulated variables combined in a cost function) is solved at each sampling. Because of this optimization problem, a desired cost function with constraints on actuator actions and/or preferred output behaviors can easily be imposed. The first value of the computed optimal control action sequence is injected into the system, and the procedure is repeated for new states in what is called Receding Horizon (RH) control mode. Figure 1.2 shows the MPC controller with the RH structure. MPC has two horizons that are expressed in terms of sampling instants. State or output prediction horizon which defines the plant states or outputs over a span of time and control horizon that gives the number of future control actions that are calculated in the state or output prediction horizon. Usually both are considered to be equal, but for different horizons, the control horizon is always smaller. The size of the prediction horizon determines the size of the online optimization problem which is normally limited by the available computational speed. The control horizon has to be chosen such that the dynamics in the system settle out [7]. Morari and Lee provided a useful survey of the MPC in [8] while Qin and Badgwell gave a technique overview in [7]. There are many works about stability, constraints handling and feasibility, and robustness of MPC as by Kothare [9], Mayne [10], and Kerrigan [11].

A typical MPC problem has a quadratic cost function that is subjected to the constraints of the controlled process dynamics, actuator saturations, and/or state or output limitations along some finite horizons. The following optimization illustrates a tracking MPC problem along the prediction $N_p$ and control $N_c$ horizons:

$$\min_{u_t,u_{t+1},\cdots,u_{t+N_c-1}} \left\{ \sum_{k=1}^{N_p} \|y_{t+k} - r_{t+k}\|_q^2 + \sum_{k=0}^{N_c-1} \|u_{t+k}\|_s^2 + \left\| y_{t+N_p} - r_{t+N_p} \right\|_{q_{N_p}}^2 \right\}$$

subject to:

$$x_{t+k+1} = f(x_{t+k}, u_{t+k})$$

$$y_{t+k} = h(x_{t+k}, u_{t+k})$$

$$u_{min} \leq u_{t+k} \leq u_{max}, \qquad k = 0,1,\cdots N_c - 1$$

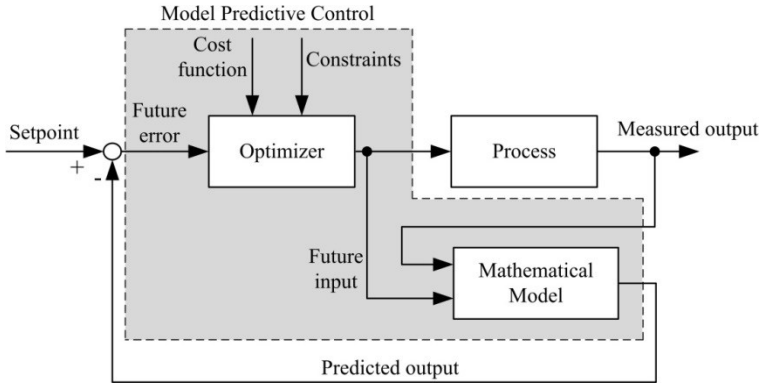
$$y_{min} \leq y_{t+k} \leq y_{max}, \qquad k = 1,2,\cdots N_p$$

$$x_t = x(t)$$

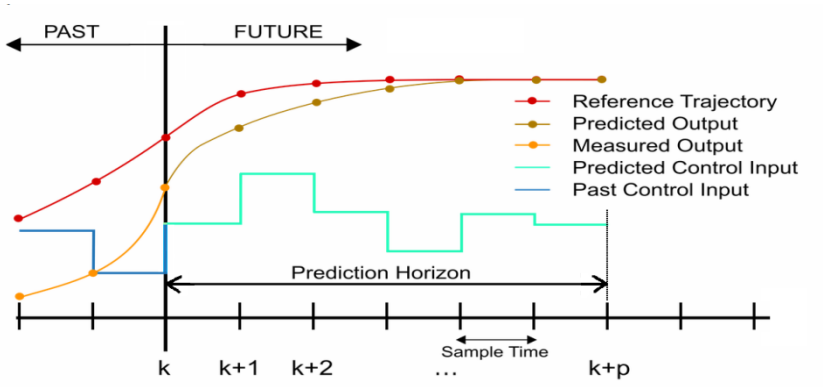
where $r_{t+k}$ is the desired tracking reference. The matrices $q$ and $s$ are the weighting matrices that are used for tuning the optimization problem. They could have constant values or values that change along the horizons. The matrix $q_{N_p}$ is also a weighting matrix that is used for the terminal cost. This is a discrete MPC problem subjected to the process dynamics that can be linear or nonlinear. A similar problem could also be written for continuous problems but with continues costs and constraints. Usually the constraints on the control actions are hard constraints, however, the output constraints are generally soften by slack variables so that they can be violated whenever the feasibility of the problem is not satisfied. MPC feasibility means that the final answer of the MPC problem is satisfying the imposed constraints.

The finite MPC problem minimizes the future cumulative cost over the horizons which can be implemented online. However, this could result in undesirable response or behavior of the process because of the limited horizons. On the other hand, an infinite horizon MPC problem minimizes the average cost over the infinite horizon. This can

give stable solutions for problems that have well defined terminal states but it is hard to be executed online as the computational time is expensive for problems with infinite variables. In order to have a finite online MPC problem that conducts an infinite behavior, the terminal weighting matrix $q_{N_p}$ is designed such as the final cost function acts as an infinite cost. By this way, the MPC problem gives stable solutions and costs that are close to the average cost along an infinite horizon.




a) The Model Predictive Control (MPC)




b) The receding horizon mode

Figure 1.2: The MPC problem with receding horizon mode from [12]

## 1.3 Architectures of MPC

### 1.3.1 Centralized and Decentralized MPC

Figure 1.3 shows a network of coupled subsystems which are controlled by one model-based predictive controller. This centralized MPC has one central model of all subsystems with couplings and one online optimization problem. In many cases the

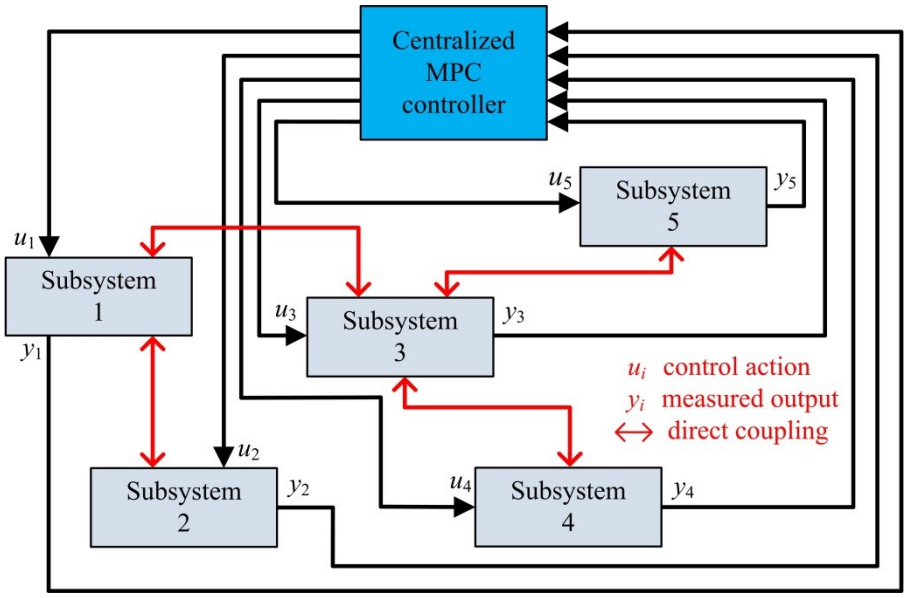



Figure 1.3: Centralized MPC with network of subsystems

centralized MPC control has been implemented successfully. However, in general, industrial plants tend to be highly complex and nonlinear and applying one centralized controller to regulate these plants is often unrealistic due to computational complexities and communication bandwidth [13]. Moreover, unstable closed-loop operation is expected if unfeasible output constraints are included in the online optimization even though the unconstrained algorithm is stable [14]. Usually large-scale plants consist of several subsystems that have different control tasks and they are connected through

material, energy, or information streams which affect the stability performance of the closed-loop system. For instance, typical heating, ventilation, and air conditioning (HVAC) applications in commercial buildings are structures of many interconnected subsystems. There are multiple chillers, heat exchangers, and a network of fans and dampers which are independently controlled and widely distributed. In addition, many of these components have dynamics that are constantly changing, with varying objectives, and are subject to failure, updates, or replacement. These couplings and different control objectives require any plantwide control approach to have a certain level of modularity and communication bandwidth so that the update of the whole network is not required for any local changes and any communication failure does not affect the control algorithm.

As an early solution for the centralized approach, completely decentralized control designs have been suggested since the 1970s. In these control architectures, the local controller attempts to adjust each plant in an interconnected system with no awareness of the dynamic couplings between the plants. The local plants are predefined by decomposing the large-scale system into softly coupled subsystems. As the actions of one agent will affect other coupled agents, design of such architecture is not easy. As an example of the design complexity is the change of plant zeros from minimum to non-minimum phase [15] and non-convex optimization problem [16]. Current research directions in decentralized control can be found in [17]. For networks with weak interacting subsystems, decentralized controllers may able to overcome the coupling if

tuned deliberately. However, if the interactions are strong, these controllers can lead to loss in performance and stability [18].

In decentralized MPC, shown in figure 1.4, a local model collectively with a local cost function is selected that does not consider the interactions or couplings between the subsystems. Although disturbances can be accounted for in any MPC controller, but if the coupling between the subsystems are significant, decentralized MPC can suffer same interaction problems. Besides interactions, stability of the closed-loop is another non-easy point to be guaranteed. Magni and Scattolini [19] added a stability constraint into the optimization problem that can ensure the stability of the local MPCs for nonlinear systems. This can solve the stability problem, but it can easily lead to feasibility issues.

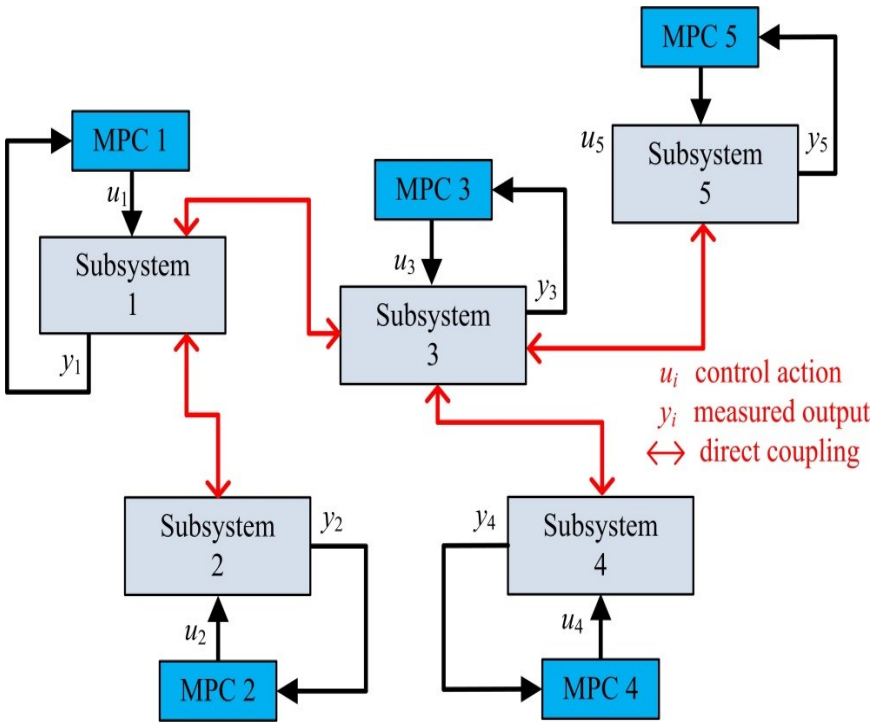



Figure 1.4: A network of subsystems with decentralized MPC controllers

### 1.3.2 Distributed MPC (DMPC)

A feasible alternative control approach is to divide the centralized optimization problem into distributed and cooperative control structures where local decisions are taken based on solving local optimal control problems in a coordinated fashion toward the centralized optimum point. In the distributed optimization approach, or more specifically the Distributed Model Predictive Control (DMPC) approach, the large-scale plant is decomposed into a number of subsystems where each has a local controller and the interaction between the subsystems are considered in the local dynamics and/or costs. Seeking the centralized performance, these local controllers exchange or share some information. Figure 1.5 shows a fully connected DMPC structure. Various DMPC algorithms have been proposed in the literature and widely referenced overviews of these algorithms can be found in [20] and [21].

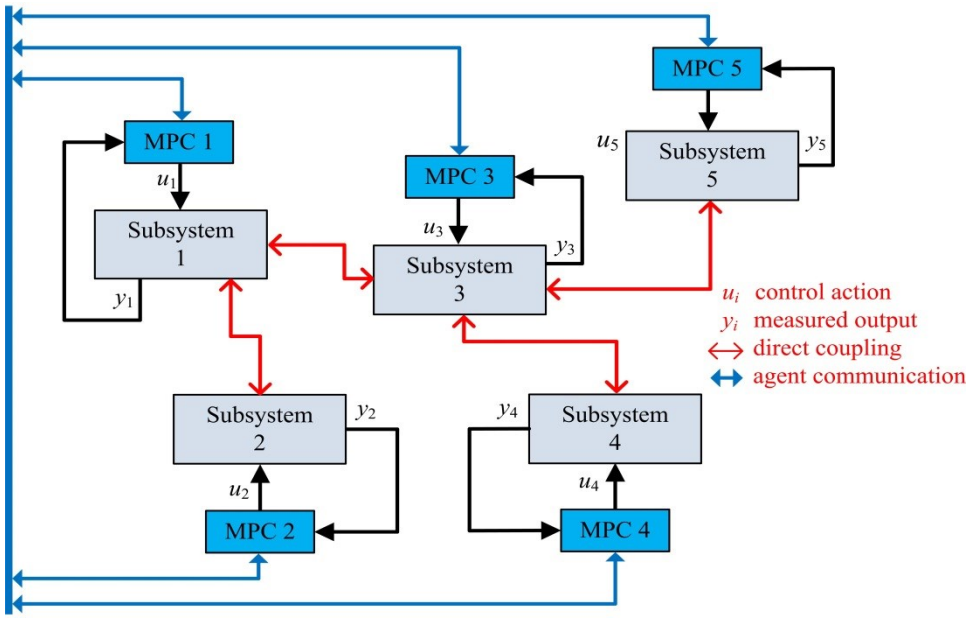



Figure 1.5: A network of subsystems with fully connected distributed MPC controllers

In literature, generally two different schemes are observed in dealing with DMPC approaches. The first suggested approach is referred to as the coordination-based approach [22] where local controllers try to improve the solution from a centralized perspective. Each local controller has an explicit centralized cost and aims to improve the systemwide solution by informing other controllers about its actions on the desired centralized performance. Decomposition optimization offers a powerful tool for solving the global optimization problems in a distributed method by informing local controllers of the impact of their actions on the global objective. Giselsson and Rantzer [23] introduced this idea in their proposed DMPC theory through dual decomposition method. Additional works with decomposition methods include Wakasa. et al. [24] and Conte et al. [25] with dual decomposition and Keviczky & Johansson [26] and Johansson et al. [27] with primal decomposition.

The second proposed methodology is based on game theory where cooperation between agents is allowed. This approach is identified as communication based or independent DMPC structures [28]. In this concept, the local agents are permitted to exchange information without having any knowledge of others cost or dynamics and the interactions between the subsystems are modeled. Scattolini [20] classified the communication based DMPC techniques according to the information exchange procedure required (iterative and non-iterative algorithms), the topology of the information sharing in the network (fully and partially connected set-up), and the modification of the local cost function (cooperative and non-cooperative local costs). Figure 1.6 shows a partially connected DMPC architecture.

Jia and Krogh [29] presented a communication-based DMPC algorithm in which the local controllers view the coupling influences as measured disturbances into the local dynamics and exchange bounds of their state trajectories. A min-max local optimization problem is solved by each controller per iteration considering the worst disturbance case. An iterative framework DMPC structure was introduced by Venkat et al. [30] to automate generation control. In this context the local MPC agents work toward satisfying the global control objective in a fully connected and cooperative fashion. Farina and Scattolini [31] proposed a non-iterative, non-cooperative partially connected DMPC algorithm where a neighbor-to-neighbor communication is required for linear discrete

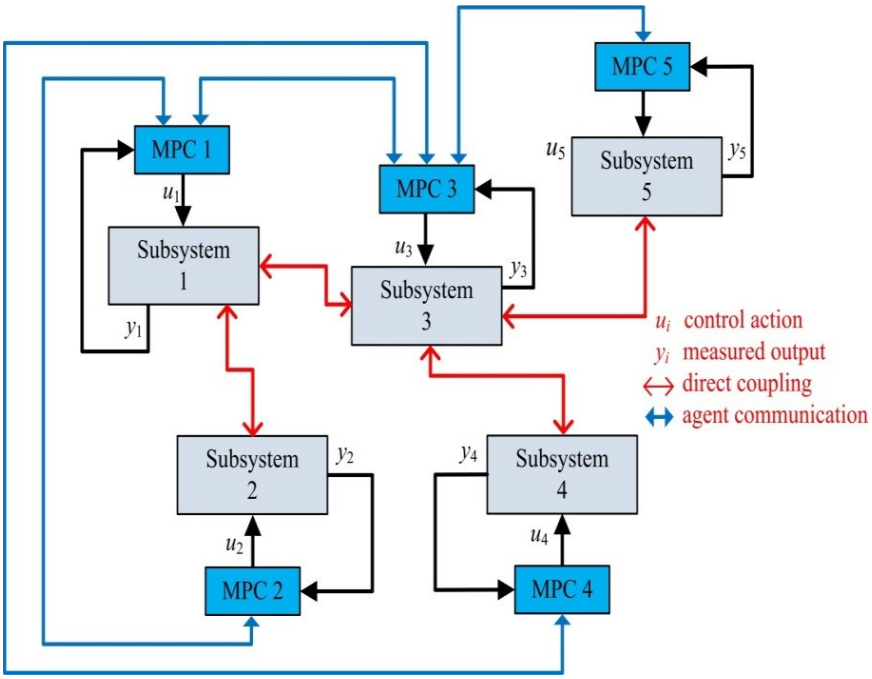



Figure 1.6: A network of subsystems with partially connected distributed MPC controllers

time systems and the local agents have no information about the local coupling dynamics. In this scheme the closed-loop convergence is proved under a number of assumptions that include solving a centralized stabilizing matrix and distributing the result through the local agents, in addition to some restrictions on the local costs.

Convergence to Nash equilibrium (the global Pareto optimum is not achieved) and poor control performance or even instability have been noted in communication-based DPMC approaches [32], [33], and [23]. This restricted the application of the communication-based DMPC for networks with weak coupled subsystems only. And as a solution, local cost functions are replaced with global objective measures or modified local cost functions [13]. The feasible cooperation-based MPC suggested by Venkat et al. [34] is a fully connected set-up at which the controllers are solving the same cost function (a strong convex combination of the local objective indices), iterating, and converging to the centralized optimum.

Sharing data with all subsystems in a network (fully connected networks) or solving a centralized problem (for example stabilizing terminal costs) as a condition for convergence to the global solution would make the DPMC algorithm impractical for many industrial applications. Communication load issues and changes in subsystem dynamics or set-points are common problems in real time applications and can easily complicate the applications of such algorithms.

Different numbers of subsystems with different scales or sizes have been used to demonstrate the aspects of many proposed DMPC theories. Alvarado et al. [35] used an experimental 4-tank process to validate the design and implementation of several

distributed control algorithms. The global system is divided into two subsystems with two states per each and they are coupled through the control actions and objective functions. For the purpose of illustration, Putta et al. [36] used a simplified state-space model of thermal zone and the accompanying AHU. A two identical-zone building is suggested in which the zones are coupled in their output temperature in addition to coupling in controls. Each zone has a state space dimension of 10 and 16 exogenous inputs. Morosan et al. [28] suggested using a virtual building composed of three coupled rooms for testing the introduced DMPC algorithm. The centralized model has 50 states which is subdivided into three coupled subsystems. For testing an accelerated gradient based distributed MPC framework, Giselsson et al. [37] generated 100 random optimization problems as a network of 100 different subsystems. The local problems are stable and controllable and have different structures and initial conditions.

## 1.4 Stability of MPC

Solving a finite optimization problem in MPC does not guarantee closed-loop stability as finite horizons do not deliver an asymptotic property [38]. Several methods can be found in literature for closed-loop stability of MPC. Terminal state inequality [39] or equality [40] constraints with a terminal cost is typically added to the MPC optimization problem to ensure the closed-loop stability. Different techniques for terminal costs and constraints can be found in [38]. Recently, global Control Lyapunov Function (CLF) was suggested by Primbs et al. [41] to develop a stabilizing RH structure with better cost objective. Because a global CLF is often not possible, this motivated Jadbabaie and

Hauser [42] to introduce the exponentially stability for input constrained RH concept using a general terminal cost that is nonnegative for a sufficiently long horizon.

Stability of MPC can also be achieved by imposing constraints on the first predicted value where it is required the two-norm of the first state to contract. Xie and Fierro [43] suggested the first-sate contractive MPC algorithm while Cheng and Jia in [44] and [45] presented the quadratic robust first-sate contractive MPC. The constrained first-state technique earns guaranteed stability but with considerable performance loss making it suitable for nonlinear MPC where stability is more important than performance [46].

Conte et al. [47] proposed a distributed synthesis for the local terminal costs and sets for a network of coupled discrete-time linear systems that can satisfy the invariance conditions in a distributed way. A structural constraint is imposed where the summation of the local stage and terminal costs must equal the global stage and terminal costs. In addition, a local agent needs to know the dynamics of the couplings with its neighbors. As an alternative method, Venkat [48] recommended solving a centralized linear quadratic regulating problem and distributing the resulted costs and sets through the subsystems. This problem is recomputed whenever a setpoint changes or a subsystem model or constraint is updated. A decentralized solution is proposed by Hermans et al [49] for stabilizing a network of nonlinear coupled discrete-time systems. For local stability, a set of structured CLF is used to decentralize the closed-loop stability conditions for which the maximum over all the functions in the set is a CLF of the global network of systems. Jia and Krogh [50] suggested adding a stability-constraint on the next predicted state in the local problems to have a stable closed-loop DMPC approach.

This approach works for full state feedback network that is controllable and satisfies a structural matching condition.

The closed-loop stability of an MPC can also be achieved by assuming sufficiently long horizons with some characteristics of the controlled system. The problems of stability and feasibility of finite receding horizon control formulation was studied by Primbs and Nevistic in [51] and [52]. For linear systems with constraints on state and control, it has been proved that for any feasible and compact initial condition set there exist a finite horizon length above which an MPC controller will deliver both stability and feasibility properties without adding any terminal costs or constraints. A similar result was also found by Boccia et al. [53] but for constrained nonlinear RH scheme with the assumptions of local controllability around the stabilized equilibrium point and sufficiently large horizon. The asymptotically stabilization and an estimated horizon length have been shown. Worthmann [54] also studied the estimation of the horizon length without imposing terminal cost or constraint at which stability or a desired degree of suboptimality is ensured. With Dini's theorem, Jadbabaie and Hauser [42] showed that a finite horizon always exists at which the RH structure is stabilizing without adding terminal costs or constraints.

## 1.5 The Online MPC Problem Size Reduction

As the length of the prediction horizon mostly determines the numerical effort needed to solve the given optimal problem in MPC, this result inevitably leads to the question of

how to reduce the computational time or the size of the online optimization problem. A number of techniques have been suggested in literature.

An explicit solution for the MPC problem can be derived as a function of the initial state that replaces the repeated online solutions. The inputs are treated as multiparametric variables and implemented online from a lookup table [55]. Hovland et al. [56] proposed using an explicit MPC structure conjugated with model reduction for real-time control application for systems with fast dynamics. An explicit MPC algorithm was also suggested by Tondel et al. [57]. The main drawback of the explicit MPC solution is that the size of the lookup table grows exponentially with the horizon and number of states and controls which makes it reliable for small problems [58].

Singular Value Decomposition (SVD) was suggested to reduce the size of variables involved in the problem as well. In SVD, the new manipulated variables are selected based on the maximum energy they can contribute to the problem in addition to constraints satisfaction. For bounded input disturbance constrained linear systems, Ong and Wang [59] proposed the SVD approach to reduce the control variables in the MPC problem. They also ensure the recursive feasibility by an auxiliary state. For DMPC, Cai et al. [60] developed an SVD framework at which the structure of the SVD is decomposed into a number of independent subproblems that can reduce the number of communications between the subsystems and ensure the convergence to the global solution for a fully connected structure.

Laguerre functions can be used to diminish the number of the online variables as well. These functions are used for parameterization because of their orthogonality and

simplicity. Wang [61] designed an MPC with Laguerre functions which was equivalent to the typical MPC but with less manipulated variables by relaxing the constraint on the exponential rate of change. Laguerre functions conjugated with multi-parametric Quadratic Programming (mp-QP) were used by Palomo and Rossiter [62] to improve the efficiency of the (mp-QP) and to largely decrease the online computations of the MPC and increase the feasibility region.

For DMPC, using sufficiently long horizons for closed-loop stability increases the size of the local problems (which can be reduced by Laguerre functions) but gives a higher modularity and more design flexibility for the local cost function structures.

### 1.6 Dissipativity Theory for Network Systems

The control of large scale networked systems can also be approached through tools from the theory of dissipative systems. Dissipativity (or passivity as a special case) is a property of an input-output dynamical system. In dissipativity, the idea is that some energy dissipates as it is transformed from point to point. The theory of dissipativity is an effective tool for analyzing network performance where the stability or stabilizability of a large-scale problem basically can be dealt with as the dissipativity analysis of the connected individual processes and coupling structure [63]. For nonlinear systems, Isidori et al. [64] developed some approaches to define storage and supply functions using the dynamic model of the system. However, these functions can also be driven based on the physics of the system such as energy [65]. Rojas et al [66] addressed the stability of a network of nonlinear processes based on the dissipativity properties of

individual subsystems and topology of the network. Storage functions that derived from the subsystem's energy (the $Q, R, S$ storage functions) are used as the base for designing the local control structures. Through the process chemistry and conservation laws, Hioe et al [67] extended the idea in [66] within $Q, R, S$ dissipation structure. The dissipativity of the local processes was extended to dissipativity-based conditions that are applicable for network analysis. A distributed dissipative-based MPC approach was proposed by Zheng et al [68] for networks with low communication rates. The controllers in this concept are exchanging information at a rate slower than the sampling rate of individual subsystems to reduce the communication burden. A dissipative analysis was designed to study the effects of such low communication rate. The analysis was formulated in terms of the plantwide dissipativity properties where supply rates in the form of Quadratic difference Forms (QdFs) were used. These supply rates are dynamic functions that change with respect to a polynomial matrix and can overcome the conservation bounds that $Q, R, S$ functions have. This leads at the end to have a dissipative trajectory conditions for the local controllers. For cascades of physically interconnected systems, a decentralized dissipativity-based nonlinear MPC approach was proposed by Varutti et al [69]. The approach can prove the systemwide stability through the independent stability of the local subsystems. An additional stability condition, which is based on dissipativity, had to be stratified by the local nonlinear MPC with the intention of staffing the global dissipativity and consequently the stability. For MPC stability, many works have been conducted based on the dissipativity inequality. For more information see [70], [71] and [72].

## 1.7 DMPC Structures with State Observer/Estimator

MPC is a state feedback controller and if an online state observer/estimator is combined with, this improves the prediction accuracy of the MPC and extends it for output feedback systems with process input and measurement noises. The state estimator unit in MPC provides the optimal estimation of the process state and disturbance based on the measurements of the outputs and control inputs in addition to an accurate knowledge of the actual noise statistics. The structure of a state estimator or observer includes the dynamical model of the plant that is used to simulate the real system with same initial conditions and inputs as the actual plant. The difference between the real and simulated measurements is used as a correction or feedback part into the simulated plant. This simulation error will be important if the process or plant is exposed to disturbances such as noise or has some uncertainties in its models. Therefore, the state observer has two phases. The first phase is the time update or prediction where the model is used to predict the current state using past information. The second phase of the observer is the measurement update where the current measurement is refined. The results from both phases are added in order to have more accurate state estimation.

For a single MPC, numerous methods have been suggested to combine the function of the MPC with state observers or Kalman filters (KF). Wang et al. [73] used KF with MPC for a class of plants with random disturbances. Luenberger state estimator was used by Mayne et al. [74] with tube-based MPC to develop a robust output feedback controller for constrained linear systems with input and output disturbances. Yan and Bitmead [75] studied the interaction between the state estimation error and constraints

for a stochastic MPC. With a Gaussian assumption, the original stochastic problem was approximated by a standard deterministically constrained MPC for the conditional mean process of the state. The idea was verified with an application of a network congestion control. Further works in this area include Geetha et al. [76], Hong and Cheng [77], and Ruscio [78].

Recently, considerable attentions have been given to distributed estimation problems in large-scale systems. Distributed state estimators can be employed to approximate states of local subsystems that are usually difficult to measure, by accessing to local and partial measurements.

Several decentralized and distributed estimation approaches have been introduced for networked large-scale systems. A decentralized and scalable form of Kalman filter for multirate sampled-data subsystems was introduced by Vadigepalli and Doyle [79]. The multi-rate design was formulated by differing the sampling for distributed agent computations and the availability of the measurements. The decomposed multivariable dynamics result in communication overhead for the estimation where heuristic guidelines were introduced to balance the computation and communication loads. Farina and Scattolini [80] used decoupled Luenberger estimators to extend the DMPC algorithm presented in [31] for output feedback case. The state estimator error was considered as an additional disturbance for local controllers. Extended Kalman filters were used by Zhenget et al. [81] to estimate the temperature for local MPCs in a DMPC schema that was used to control a hot-rolled strip laminar cooling process. For multi-rate local systems, Yamchi et al. [82] proposed a cooperative distributed Kalman filter

algorithm along with a DMPC scheme. The local subsystems were defined to be multi-rate systems as the input updates or output measurement is not available at a particular sampling. Each distributed agent (KF+MPC) considers other agent actions into local dynamics through communication. A networked control structure based on distributed Kalman filters for unconstrained systems was also presented by Menighed et al. [83]. The proposed architecture is suitable for fault tolerant control of distributed subsystems under actuator faults. The local state estimators are Kalman filters which are the key components for the realistic DMPC algorithm.

## 1.8 Introduction to the LC-DMPC Algorithm

In this work, a novel Limited-Communication Distributed Model Predictive Control (LC-DMPC) scheme for coupled and constrained linear discrete-time systems is introduced. The LC-DMPC approach is an iterative, cooperative, partially connected algorithm. This algorithm has been analyzed under two different cases.

In the first case, the local dynamics and cost function information for all subsystems are known at a central monitor for checking the convergence of the algorithm. Here, the systemwide optimum performance can be achieved where convergence requires only coupled subsystems with local structural information to cooperate for sufficient number of iterations. In order to have an element of modularity, the closed-loop stability of local agents is ensured by assuming appropriately long horizons. To overcome the computational efforts associated with large numbers of manipulated variables, local control actions are parametrized by Laguerre functions which leads to smaller local

optimization problems. Laguerre functions are also used to parametrize the exchanged signals in a different event.

In the second case, the suboptimal performance of the LC-DMPC algorithm is analyzed through the dissipativity theory. There is no need for a centralized supervisor and the network convergence is guaranteed through the dissipativity of the local communication signals in the iteration domain. As convergence to the global performance is lost, a new method is introduced to prove the local closed-loop stability. A distributed method for designing stabilizable terminal costs is developed which requires the local agents to share the interaction dynamics only.

In both cases, the LC-DMPC involves neighbor-to-neighbor communication structure. During a sampling time, each distributed agent shares its predicted future effects with its neighbors and receives the sensitivity that the neighbors' cost functions have for these effects at the next iteration. Then the controller solves an updated problem by minimizing the summations of a deviation from a local reachable reference, local control efforts, and the future effects for neighbors weighted by the received sensitivity information. Constraints on state and control actions can be handled by local controllers.

The proposed algorithm in the first case has the following highlighted characteristics: (i) any physical changes or model updates in a subsystem do not require any updates for its neighbors. Only the central monitor needs to be updated; (ii) solution of the distributed controllers converges to the global optimum point through: parietal commination (neighbor-to-neighbor), solving a modified cost function, and local knowledge of dynamics and costs; (iii) the central supervisor only checks the

convergence condition which does not involve solutions of any centralized problem and distribute the results..

On the other hand, the suboptimal LC-DMPC framework characterizes most of the above highlights except changes in the coupling dynamics need to be exchanged and the stabilizing terminal costs have to be recomputed. However, this is accomplished in a distributed fashion. In addition, the algorithm will be suboptimal with respect to the centralized performance.

The high level of modularity of the LC-DMPC algorithm makes it an appropriate plantwide control approach for many industrial plants such as HVAC systems in commercial buildings where components' setpoints or models change frequently.

# 2. THEORY OF LIMITED-COMMUNICATION DISTRIBUTED MODEL PREDICTIVE CONTROL (LC-DMPC)*

In this dissertation, the Limited-Communication DMPC (LC-DMPC) algorithm is introduced for coordinating constrained and coupled linear distributed subsystems that can be found in any large-scale plants such as HVAC systems. The centralized problem is subdivided into cooperative local subproblems using the structure of upstream and downstream subsystems. The distributed agents in the LC-DMPC algorithm function as supervisor controllers by computing the optimum set-points for lower level controllers (usually PI or PID) based on solving the local optimization problems. The presented algorithm in this chapter can converge to the centralized solution by only requiring an iterative neighbor-to-neighbor communication architecture. Using local dynamics and costs information, each local agent solves the distributed cost function with local constraints and shares some predicted data with its immediate local neighbors (not with all agents). Assuming equal prediction and control horizons and all subsystems have same prediction length; stability of the closed-loop subsystems under the action of the LC-DMPC algorithm is proved with constraints only on control actions. The stability is ensured by assuming sufficiently long horizons. To overcome the problems associated with long horizons, Laguerre functions are used to parameterize the local control actions and reduce the number of the optimized variables. In a different case, Laguerre functions

* Parts of this section are reprinted with permission from R. Jalal and B. Rasmussen, "Limited-communication distributed model predictive control for coupled and constrained linear systems," IEEE Transaction on Control System Technology, accepted, 2016.

are used by the local agent to parametrize the exchanged signals as well to reduce the communication burden.

This chapter is organized as follows: First, the mathematical preliminaries which include the structure of upstream and downstream, local and global dynamics and cost functions, and local Kalman filters are introduced. The main algorithm, convergence, and closed-loop stability are then detailed. The last section presents the Laguerre functions for parametrization of the local control actions and exchanged vectors.

## 2.1 Preliminaries

### 2.1.1 Topology

The definitions, structures and equations used to describe the LC-DPMC approach are presented in this section. For $p$ dynamically coupled subsystems in a network, the dynamics of subsystem $\Sigma_i$ can be realized as a set of discrete-time difference equations as follows:

$$\begin{aligned} x_i(k+1) &= f_i\{x_i(k), u_i(k), v_i(k), d_i(k)\} \\ y_i(k) &= h_i\big(x_i(k), u_i(k)\big) \\ z_i(k) &= q_i\big(x_i(k), u_i(k)\big) \end{aligned} \tag{2.1}$$

where $x_i \in \Re^{n_i}$ is the state vector, $u_i \in \Re^{r_{u,i}}$ is the input control vector that affects subsystem $\Sigma_i$ locally, $y_i \in \Re^{p_{y,i}}$ is the regulated output vector, and let $x_{0,i}(k)$ be the initial condition of the state vector where $k$ is the current discrete time step. The vector $v_i \in \Re^{r_{v,i}}$ represents the measured input disturbances from upstream neighboring dynamic subsystems. On the other hand, if subsystem $\Sigma_i$ has outputs that act as disturbances for some downstream subsystems in the network, then these outputs are

referred to as the vector $z_i \in \Re^{p_{z,i}}$. Unmeasured disturbances enters as $d_i \in \Re^{r_{d,i}}$. This partition of a subsystem is referred to as upstream and downstream architecture in this work. Figure 2.1 illustrates such structure.

To predict the future values of vectors $y_i$ and $z_i$, let $N_{p,i}$ be the prediction horizon (assume equal state and control prediction horizons). Stacking all predicted values of both vectors along $N_{p,i}$, one can write:

$$\mathrm{Y}_i = \left[y_i^T(k+1) \;\; y_i^T(k+2) \;\; \cdots \;\; y_i^T\left(k+N_{p,i}\right)\right]^T \tag{2.2}$$

$$\mathrm{Z}_i = \left[z_i^T(k) \;\; z_i^T(k+1) \;\; \cdots \;\; z_i^T\left(k+N_{p,i}-1\right)\right]^T \tag{2.3}$$

Similarly define:

$$\mathrm{V}_i = \left[v_i^T(k) \;\; v_i^T(k+1) \cdots \;\; v_i^T\left(k+N_{p,i}-1\right)\right]^T \tag{2.4}$$

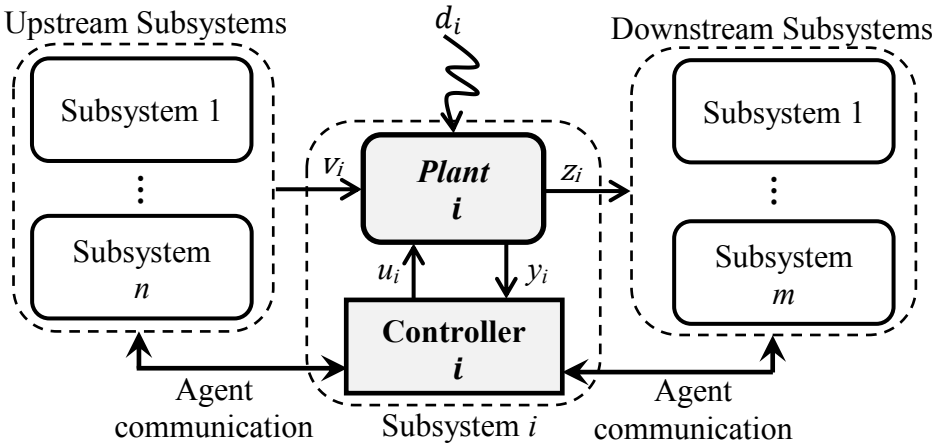



Figure 2.1: A schematic diagram of an LC-DMPC subsystem

For simplicity let all of the subsystems in the network have same prediction horizons equal to $N_p$, then the network (systemwide) disturbance inputs and outputs can be formulated as:

$$\mathbf{V}_{r_v \times 1} = \left[\mathrm{V}_1^T \;\; \mathrm{V}_2^T \cdots \mathrm{V}_p^T\right]^T \tag{2.5}$$

$$\mathbf{Z}_{p_z \times 1} = \left[\mathrm{Z}_1^T \;\; \mathrm{Z}_2^T \cdots \mathrm{Z}_p^T\right]^T \tag{2.6}$$

with $r_v = N_p \cdot \sum_{i=1}^{p} r_{v,i}$ and $p_z = N_p \cdot \sum_{i=1}^{p} p_{z,i}$.

For theoretical analysis only, an interconnecting matrix $\mathbf{\Gamma} \in \{0,1\}$ with dimension of $r_v \times p_z$ is defined to manage the communication of the predicted data (2.5) and (2.6) between the local agents. This matrix relates the network inputs and outputs as:

$$\mathbf{V} = \mathbf{\Gamma Z} \tag{2.7}$$

The structure of the interconnecting matrix must reveal the actual coupling between the subsystems. That is for each subsystem disturbance output $\mathrm{Z}_i$, there is a corresponding subsystem input vector(s) $\mathrm{V}_i$ (i.e. $r_v = p_z$). As an example, figure 2.2 shows five coupled plants in a network with different couplings. Plant 2 affects plants 3, 4, and 5, however it has only one disturbance input coming from plant 1. Plants 3 and 4 affect each other and both disturb plant 5. Finally, plant 1 has two disturbance outputs for plants 2 and 5 and it has only one disturbance from plant 5.

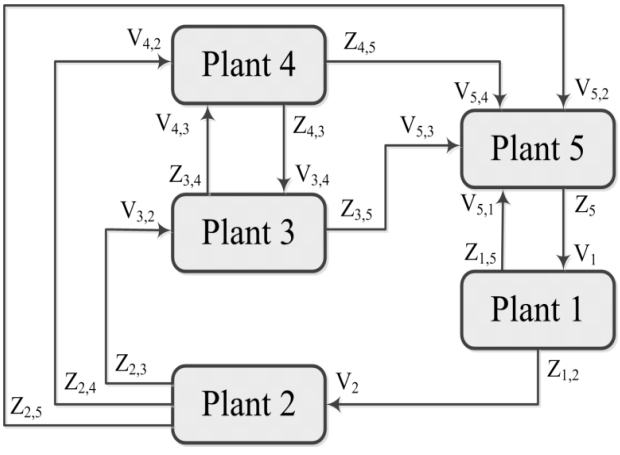



Figure 2.2: An example of five coupled subsystems in a

Assuming $N_p = 1$, then the network outputs of these plants could be expressed as:

$$\mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} Z_{1,2} \\ Z_{1,5} \end{bmatrix} \\ \begin{bmatrix} Z_{2,3} \\ Z_{2,4} \\ Z_{2,5} \end{bmatrix} \\ \begin{bmatrix} Z_{3,4} \\ Z_{3,5} \end{bmatrix} \\ \begin{bmatrix} Z_{4,3} \\ Z_{4,5} \end{bmatrix} \\ Z_5 \end{bmatrix} \tag{2.8}$$

The network disturbance inputs are related to **Z** through the interconnecting matrix **Γ** as:

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ \begin{bmatrix} V_{3,2} \\ V_{3,4} \end{bmatrix} \\ \begin{bmatrix} V_{4,2} \\ V_{4,3} \end{bmatrix} \\ \begin{bmatrix} V_{5,1} \\ V_{5,2} \\ V_{5,3} \\ V_{5,4} \end{bmatrix} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} \begin{bmatrix} Z_{1,2} \\ Z_{1,5} \end{bmatrix} \\ \begin{bmatrix} Z_{2,3} \\ Z_{2,4} \\ Z_{2,5} \end{bmatrix} \\ \begin{bmatrix} Z_{3,4} \\ Z_{3,5} \end{bmatrix} \\ \begin{bmatrix} Z_{4,3} \\ Z_{4,5} \end{bmatrix} \\ Z_5 \end{bmatrix} \tag{2.9}$$

This concept can be extended for any value of $N_p$ (see appendix I).

Through the structure of **Γ** one can get the desired type of coupling for simulation. This example also shows that two coupled plants can simultaneously be upstream and downstream neighbors where the neighbors are defined based on the flow of the coupling signals.

### 2.1.2 Local Dynamic Models

For linear discrete-time systems, the equations given in (2.1) can be expressed as:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_{d,i} d_i(k) \\ y_i(k) &= C_{y,i} x_i(k) + D_{y,i} u_i(k) \\ z_i(k) &= C_{z,i} x_i(k) + D_{z,i} u_i(k) \end{aligned} \tag{2.10}$$

where the exogenous disturbance $d_i$ is assumed to measurable. The predicted future values of the vectors $y_i$ and $z_i$ at the sampling time $m$ can be computed by the repeated application of (2.10) as:

$$y_i(k+m) = C_{y,i} {A_i}^m x_{0,i}(k) + D_{yu,i} U_i + D_{yv,i} V_i + D_{yd,i} D_i \tag{2.11}$$

$$z_i(k+m) = C_{z,i} {A_i}^m x_{0,i}(k) + D_{zu,i} U_i + D_{zv,i} V_i + D_{zd,i} D_i \tag{2.12}$$

where:

$$D_{yu,i} = [C_{y,i} {A_i}^{m-1} B_{u,i} \;\; C_{y,i} {A_i}^{m-2} B_{u,i} \;\cdots\; C_{y,i} B_{u,i} \;\; D_{y,i}]$$

$$D_{yv,i} = [C_{y,i} {A_i}^{m-1} B_{v,i} \;\; C_{y,i} {A_i}^{m-2} B_{v,i} \;\cdots\; C_{y,i} B_{v,i}]$$

$$D_{yd,i} = [C_{y,i} {A_i}^{m-1} B_{d,i} \;\; C_{y,i} {A_i}^{m-2} B_{d,i} \;\cdots\; C_{y,i} B_{d,i}]$$

$$D_{zu,i} = [C_{z,i} {A_i}^{m-1} B_{u,i} \;\; C_{z,i} {A_i}^{m-2} B_{u,i} \;\cdots\; C_{z,i} B_{u,i} \;\; D_{z,i}]$$

$$D_{zv,i} = [C_{z,i} {A_i}^{m-1} B_{v,i} \;\; C_{z,i} {A_i}^{m-2} B_{v,i} \;\cdots\; C_{z,i} B_{v,i}]$$

$$D_{zv,i} = [C_{z,i} {A_i}^{m-1} B_{v,i} \;\; C_{z,i} {A_i}^{m-2} B_{v,i} \;\cdots\; C_{z,i} B_{v,i}]$$

$$U_i = [u_i^T(k) \;\; u_i^T(k+1) \;\cdots\; u_i^T(k+m)]^T$$

$$V_i = [v_i^T(k) \;\; v_i^T(k+1) \;\cdots\; v_i^T(k+m-1)]^T$$

$$D_i = [d_i^T(k) \;\; d_i^T(k+1) \;\cdots\; d_i^T(k+m-1)]^T$$

Referring to equations (2.2) and (2.3), all predicted values of $\mathrm{Y}_i$ and $\mathrm{Z}_i$ along $N_p$ can be written in matrix and vector formats as:

$$\mathrm{Y}_i = F_{y,i}x_{0,i}(k) + M_{y,i}\mathrm{U}_i + N_{y,i}\mathrm{V}_i + P_{y,i}\mathrm{D}_i \tag{2.13}$$

$$\mathrm{Z}_i = F_{z,i}x_{0,i}(k) + M_{z,i}\mathrm{U}_i + N_{z,i}\mathrm{V}_i + P_{z,i}\mathrm{D}_i \tag{2.14}$$

where

$$F_{y,i} = \begin{bmatrix} C_{y,i}A_i \\ C_{y,i}{A_i}^2 \\ C_{y,i}{A_i}^3 \\ \vdots \\ C_{y,i}{A_i}^{N_p} \end{bmatrix}, \quad F_{z,i} = \begin{bmatrix} C_{z,i} \\ C_{z,i}A_i \\ C_{z,i}{A_i}^2 \\ \vdots \\ C_{z,i}{A_i}^{N_p-1} \end{bmatrix}$$

and $M_{y,i}, N_{y,i}, M_{z,i}, N_{z,i}, P_{y,i}, \& P_{z,i}$ are the lifted representations of the Markov parameters computed from $D_{yu,i}, D_{yv,i}, D_{zu,i}, D_{zv,i}, D_{yd,i}, \& D_{zd,i}$ along $N_p$, given as:

$$M_{y,i} = \begin{bmatrix} C_{y,i}B_{u,i} & D_{y,i} & 0 & \cdots & 0 & 0 \\ C_{y,i}A_iB_{u,i} & C_{y,i}B_{u,i} & D_{y,i} & \cdots & 0 & 0 \\ C_{y,i}A_i^2B_{u,i} & C_{y,i}A_iB_{u,i} & C_{y,i}B_{u,i} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{y,i}{A_i}^{N_p-1}B_{u,i} & C_{y,i}{A_i}^{N_p-2}B_{u,i} & C_{y,i}{A_i}^{N_p-3}B_{u,i} & \cdots & C_{y,i}B_{u,i} & D_{y,i} \end{bmatrix}$$

$$N_{y,i} = \begin{bmatrix} C_{y,i}B_{v,i} & 0 & 0 & \cdots & 0 & 0 \\ C_{y,i}A_iB_{v,i} & C_{y,i}B_{v,i} & 0 & \cdots & 0 & 0 \\ C_{y,i}A_i^2B_{v,i} & C_{y,i}A_iB_{v,i} & C_{y,i}B_{v,i} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{y,i}{A_i}^{N_p-1}B_{v,i} & C_{y,i}{A_i}^{N_p-2}B_{v,i} & C_{y,i}{A_i}^{N_p-3}B_{v,i} & \cdots & C_{y,i}A_iB_{v,i} & C_{y,i}B_{v,i} \end{bmatrix}$$

$$P_{y,i} = \begin{bmatrix} C_{y,i}B_{d,i} & 0 & 0 & \cdots & 0 & 0 \\ C_{y,i}A_iB_{d,i} & C_{y,i}B_{d,i} & 0 & \cdots & 0 & 0 \\ C_{y,i}A_i^2B_{d,i} & C_{y,i}A_iB_{d,i} & C_{y,i}B_{d,i} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{y,i}{A_i}^{N_p-1}B_{d,i} & C_{y,i}{A_i}^{N_p-2}B_{d,i} & C_{y,i}{A_i}^{N_p-3}B_{d,i} & \cdots & C_{y,i}A_iB_{d,i} & C_{y,i}B_{d,i} \end{bmatrix}$$

$$M_{z,i} = \begin{bmatrix} D_{z,i} & 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}B_{u,i} & D_{z,i} & 0 & \cdots & 0 & 0 \\ C_{z,i}A_iB_{u,i} & C_{z,i}B_{u,i} & D_{y,i} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{z,i}{A_i}^{N_p-2}B_{u,i} & C_{z,i}{A_i}^{N_p-3}B_{u,i} & C_{z,i}{A_i}^{N_p-4}B_{u,i} & \cdots & C_{z,i}B_{u,i} & D_{z,i} \end{bmatrix}$$

$$N_{z,i} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}B_{v,i} & 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}A_iB_{v,i} & C_{z,i}B_{v,i} & 0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{z,i}{A_i}^{N_p-2}B_{v,i} & C_{z,i}{A_i}^{N_p-3}B_{v,i} & C_{z,i}{A_i}^{N_p-4}B_{v,i} & \cdots & C_{z,i}B_{v,i} & 0 \end{bmatrix}$$

$$P_{z,i} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}B_{d,i} & 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}A_iB_{d,i} & C_{z,i}B_{v,i} & 0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ C_{z,i}{A_i}^{N_p-2}B_{d,i} & C_{z,i}{A_i}^{N_p-3}B_{d,i} & C_{z,i}{A_i}^{N_p-4}B_{d,i} & \cdots & C_{z,i}B_{d,i} & 0 \end{bmatrix}$$

and also

$$\mathrm{U}_i = \left[u_i^T(k) \;\; u_i^T(k+1) \;\cdots\; u_i^T\left(k+N_p\right)\right]^T$$

$$\mathrm{V}_i = \left[v_i^T(k) \;\; v_i^T(k+1) \;\cdots\; v_i^T\left(k+N_p-1\right)\right]^T$$

$$\mathrm{D}_i = \left[d_i^T(k) \;\; d_i^T(k+1) \;\cdots\; d_i^T\left(k+N_p-1\right)\right]^T$$

### 2.1.3 The Global and Local Cost Functions

The main objective of the LC-DMPC algorithm presented in this chapter is to solve a distributed optimum control problem and ensure that the solution converges to the solution of the systemwide optimum per sampling. In this work, the centralized controller is solving a typical linear quadratic MPC problem that is a linearly constrained quadratic cost function which penalizes the sum of the square of the tracking error (deviation form a desired reference) and the square of the manipulated control inputs along $N_p$. On the other hand, each distributed local agent is required to solve a modified cost function with three terms. The first two terms comprise the centralized cost terms but defined locally, while the last term is the sensitivity of the local downstream

neighbor cost functions to the change in the local network inputs (i.e. $V_i$). That is, agent $i$ is solving a cost function with penalties on its effects for the downstream subsystems. These effects are updated frequently by downstream agents.

For both systemwide and local cost functions, let the constant matrices $q_i \in \Re^{p_{y,i} \times p_{y,i}} > 0$ and $s_i \in \Re^{r_{u,i} \times r_{u,i}} > 0$ be two matrices that serve as weights on the predicted errors and future control actions, respectively.

#### 2.1.3.1 The Centralized Optimization Problem

Before introducing more variables it is important to note that any defined variable with an italicized lower case letter signifies a local scalar variable at the indicated time. On the other hand, a variable that is defined with a non-italicized and lower case letter refers to a vector variable along the horizon. Moreover, variables defined with a non-italicized and bold lower case letters represent variables at the network level and along the horizon.

Define the local error signal along $N_p$ as:

$$\mathrm{e}_i = (\mathrm{r}_i(k) - \mathrm{Y}_i) \tag{2.15}$$

where

$$\mathrm{r}_i(k) = \left[r_i^T(k+1) \;\; r_i^T(k+2) \;\cdots\; r_i^T(k+N_p)\right]^T$$

and $r_i$ is the desired local reference that must be reachable by the local controller with the local constraints.

To state the centralized objective function and prove the convergence of the LC-DMPC algorithm, the following vectors and block diagonal matrices need to be defined:

**Stacked vectors**:

Network reference signals:

$$\mathbf{r}_{p_y \times 1} = \left[\mathrm{r}_1^T \ \mathrm{r}_2^T \cdots \mathrm{r}_p^T\right]^T$$

Regulated outputs:

$$\mathbf{Y}_{p_y \times 1} = \left[\mathrm{Y}_1^T \ \mathrm{Y}_2^T \cdots \mathrm{Y}_p^T\right]^T$$

Network error signals:

$$\mathbf{e}_{p_y \times 1} = \left[\mathrm{e}_1^T \ \mathrm{e}_2^T \cdots \mathrm{e}_p^T\right]^T$$

Control inputs:

$$\mathbf{U}_{r_u \times 1} = \left[\mathrm{U}_1^T \ \mathrm{U}_2^T \cdots \mathrm{U}_p^T\right]^T$$

Network initial conditions:

$$\mathbf{X_0}(k)_{n \times 1} = \left[x_{0,1}(k)^T \ \ x_{0,2}(k)^T \cdots x_{0,p}(k)^T\right]^T$$

Measured exogenous disturbance:

$$\mathbf{D}_{r_d \times 1} = \left[\mathrm{D}_1^T \ \mathrm{D}_2^T \cdots \mathrm{D}_p^T\right]^T$$

**Block diagonal matrices**:

$$\mathbf{Q} = diag\left(Q_1, Q_2, \cdots, Q_p\right), \mathbf{S} = diag\left(S_1, S_2, \cdots, S_p\right)$$

where $\mathrm{Q}_i$ and $\mathrm{S}_i$ are given by:

$$\mathrm{Q}_i = diag\left(q_i(k+1),\ q_i(k+2), \cdots,\ q_i(k+N_p)\right)$$

$$\mathrm{S}_i = diag\left(s_i(k+1),\ s_i(k+2), \cdots,\ s_i(k+N_p)\right)$$

$$\mathbf{F_y} = diag\left(F_{y,1}, F_{y,2}, \cdots, F_{y,p}\right), \quad \mathbf{F_z} = diag\left(F_{z,1}, F_{z,2}, \cdots, F_{z,p}\right)$$

$$\mathbf{M_y} = diag\left(M_{y,1}, M_{y,2}, \cdots, M_{y,p}\right), \quad \mathbf{M_z} = diag\left(M_{z,1}, M_{z,2}, \cdots, M_{z,p}\right)$$

$$\mathbf{N_y} = diag\left(N_{y,1}, N_{y,2}, \cdots, N_{y,p}\right), \quad \mathbf{N_z} = diag\left(N_{z,1}, N_{z,2}, \cdots, N_{z,p}\right)$$

$$\mathbf{P_y} = diag\left(N_{y,1}, N_{y,2}, \cdots, N_{y,p}\right), \quad \mathbf{P_z} = diag\left(N_{z,1}, N_{z,2}, \cdots, N_{z,p}\right)$$

Therefore, the network regulated output $\mathbf{Y}$ and disturbance output $\mathbf{Z}$ for the entire system are given by:

$$\mathbf{Y} = \mathbf{F_y X_0}(k) + \mathbf{M_y U} + \mathbf{N_y V} + \mathbf{P_y D} \tag{2.16}$$

$$\mathbf{Z} = \mathbf{F_z X_0}(k) + \mathbf{M_z U} + \mathbf{N_z V} + \mathbf{P_z D} \tag{2.17}$$

The network disturbance input $\mathbf{V}$ can be written as a function of the initial conditions and manipulated inputs using equation (2.17) as shown below:

$$\mathbf{V} = \mathbf{\Gamma Z} = (I - \mathbf{\Gamma N_z})^{-1}\mathbf{\Gamma}(\mathbf{F_z X_0}(k) + \mathbf{M_z U} + \mathbf{P_z D}) = \mathbf{W}(\mathbf{F_z X_0}(k) + \mathbf{M_z U} + \mathbf{P_z D}) \tag{2.18}$$

where $\mathbf{W} = (I - \mathbf{\Gamma N_z})^{-1}\mathbf{\Gamma}$.

The global cost function is a typical squared $\ell_2$ norm cost with constraints imposed only on the control actions:

$$\min_{\substack{u(k+m) \\ m=0,1,\cdots,N_p-1}} J = \sum_{m=1}^{N_p} \|e(k+m)\|_Q^2 + \sum_{m=0}^{N_p-1} \|u(k+m)\|_S^2$$

subject to:

$$x(k+1) = Ax(k) + B_u u(k) + B_d d(k)$$
$$y(k) = Cx(k) + Du(k)$$

$$u_{min} \le u(k+m) \le u_{max} \qquad m = 0,1,\cdots,N_p - 1$$

where

$$e = \left[e_1^T \;\; e_2^T \;\; \cdots \;\; e_p^T\right], e_i = r_i - y_i, i = 1,2,\cdots,p$$

$$u = \left[u_1^T \;\; u_2^T \;\; \cdots \;\; u_p^T\right], d = \left[d_1^T \;\; d_2^T \;\; \cdots \;\; d_p^T\right]$$

$$Q = diag\left(q_1, q_2, \cdots, q_p\right), \; S = diag\left(s_1, s_2, \cdots, s_p\right)$$

$$A = diag(A_1, A_2, \cdots, A_2) + diag\big(B_{v,1}, B_{v,2}, \cdots, B_{v,p}\big)\Gamma_1 diag\big(C_{z,1}, C_{z,2}, \cdots, C_{z,p}\big)$$

$$B_u = diag\big(B_{u,1}, B_{u,2}, \cdots, B_{u,p}\big) + diag\big(B_{v,1}, B_{v,2}, \cdots, B_{v,p}\big)\Gamma_1 diag\big(D_{z,1}, D_{z,2}, \cdots, D_{z,p}\big)$$

$$B_d = diag\big(B_{d,1}, B_{d,2}, \cdots, B_{d,p}\big)$$

$$C = diag\big(C_{y,1}, C_{y,2}, \cdots, C_{y,p}\big),\ D = diag\big(D_{y,1}, D_{y,2}, \cdots, D_{y,p}\big)$$

$$u_{min} = \left[u_{min,1}^T\ u_{min,2}^T\ \cdots\ u_{min,p}^T\right]^T,\ u_{max} = \left[u_{max,1}^T\ u_{max,2}^T\ \cdots\ u_{max,p}^T\right]^T$$

$\Gamma_1 \equiv$ the interconnecting matrix for horizon equal to one ($N_p = 1$)

and the norm $\|x\|_P$ is given as $\sqrt[2]{x^T P x}$.

In terms of the defined column vectors and block diagonal matrices, however, the above centralized problem can be expressed as following:

$$\min_{\mathbf{U}} J = \mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{U}^T\mathbf{S}\mathbf{U}$$

subject to:

$$\mathbf{Y} = \mathbf{F_y X_0}(k) + \mathbf{M_y U} + \mathbf{N_y W}(\mathbf{F_z X_0}(k) + \mathbf{M_z U} + \mathbf{P_z D}) + \mathbf{P_y D}$$

$$\mathbf{U}_{min} \leq \mathbf{U} \leq \mathbf{U}_{max}$$

which can also be reformulated as:

$$\left.\begin{array}{c} \mathbf{min_U}\, J = \mathbf{U^T H U} + 2\mathbf{U^T F} \\ \mathbf{A_{iq} U} \leq \mathbf{B_{iq}} \end{array}\right\} \quad (2.19)$$

subject to:

where:

$$\mathbf{H} = \mathbf{S} + (\mathbf{M_y} + \mathbf{N_y W M_z})^T \mathbf{Q} (\mathbf{M_y} + \mathbf{N_y W M_z})$$

$$\mathbf{F} = \big(\mathbf{M_y} + \mathbf{N_y W M_z}\big)^T \mathbf{Q}\Big(\big(\mathbf{F_y} + \mathbf{N_y W F_z}\big)\mathbf{X_0}(k) + \big(\mathbf{P_y} + \mathbf{N_y W P_z}\big)\mathbf{D}\Big) -$$

$$(\mathbf{M_y} + \mathbf{N_y W M_z})^T \mathbf{Q r}(k)$$

$$\mathbf{A_{iq}} = diag\begin{pmatrix} I_{r_{u,i}*N_p} \\ -I_{r_{u,i}*N_p} \end{pmatrix}, \cdots, \begin{matrix} I_{r_{u,p}*N_p} \\ -I_{r_{u,p*N_p}} \end{matrix}\Bigg)$$

$$\mathbf{B_{iq}} = \left[\begin{pmatrix} \mathrm{U}_i^T{}_{max} & \mathrm{U}_i^T{}_{min} \end{pmatrix} \quad \cdots \quad \begin{pmatrix} \mathrm{U}_p^T{}_{max} & \mathrm{U}_p^T{}_{min} \end{pmatrix}\right]^T$$

$$\mathrm{U}_{i_{min}} = \left[u_{min,i}^T(k) \ \cdots \ u_{min,i}^T(k+N_p-1)\right]^T$$

$$\mathrm{U}_{i_{max}} = \left[u_{max,i}^T(k) \ \cdots \ u_{max,i}^T(k+N_p-1)\right]^T$$

and $I_n$ is the square identity matrix with dimension $n$.

### 2.1.3.2 The LC-DMPC Local Optimization Problem

For the distributed LC-DMPC optimization problems, each local controller is assigned to solve the following local problem:

$$\min_{\substack{u_i(k+m) \\ m=0,1,\cdots,N_p-1}} J_i = \sum_{m=1}^{N_p} \|e_i(k+m)\|_{q_i}^2 + \sum_{m=0}^{N_p-1} \|u_i(k+m)\|_{s_i}^2 +$$

$$\sum_{m=0}^{N_{p-1}} \psi_i(k+m)^T z_i(k+m)$$

subject to:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_{d,i} d_i(k) \\ y_i(k) &= C_{y,i} x_i(k) + D_{y,i} u_i(k) \\ z_i(k) &= C_{z,i} x_i(k) + D_{z,i} u_i(k) \end{aligned}$$

$$u_{min,i} \le u_i(k+m) \le u_{max,i} \qquad m = 0,1,\cdots,N_p-1$$

With the defined local vectors and matrices along $N_p$, the cost function in the above optimization problem can be written as:

$$J_i = \ \mathrm{e}_i^T \mathrm{Q}_i \mathrm{e}_i + \mathrm{U}_i^T \mathrm{S}_i \mathrm{U}_i + \Psi_i^T \mathrm{Z}_i \tag{2.20}$$

where

$$\Psi_i = \left[\psi_i^T(k) \;\; \psi_i^T(k+1) \;\; \cdots \;\; \psi_i^T\left(k+N_p-1\right)\right]^T$$

The vector $\Psi_i$ represents the sensitivity of downstream neighbors' cost functions to the disturbance coming from the $i^{\text{th}}$ plant. This optimization problem can be reformulated as:

$$\left.\begin{array}{c} \min_{U_i} J_i = \; \mathrm{U}_i^T H_i \mathrm{U}_i + 2\mathrm{U}_i^T F_i + \mathrm{V}_i^T E_i \mathrm{V}_i + 2\mathrm{V}_i^T T_i \\ \text{subject to} \\ A_{i_{iq}} \mathrm{U}_i \le B_{i_{iq}} \end{array}\right\} \tag{2.21}$$

where:

$$H_i = M_{y,i}^T \mathrm{Q}_i M_{y,i} + \mathrm{S}_i$$

$$F_i = M_{y,i}^T \mathrm{Q}_i \left[F_{y,i} x_{0,i}(k) + N_{y,i}\mathrm{V}_i + P_{y,i}\mathrm{D}_i - \mathrm{r}_i(k)\right] + 0.5 M_{z,i}^T \Psi_i$$

$$E_i = N_{y,i}^T \mathrm{Q}_i N_{y,i}$$

$$T_i = N_{y,i}^T \mathrm{Q}_i \left[F_{y,i} x_{0,i}(k) + P_{y,i}\mathrm{D}_i - \mathrm{r}_i(k)\right] + 0.5 N_{z,i}^T \Psi_i$$

$$A_{i_{iq}} = diag\begin{pmatrix} I_{r_{u,i}*N_p} \\ -I_{r_{u,i}*N_p} \end{pmatrix}$$

$$B_{i_{iq}} = \left[\mathrm{U}_{i\,max}^T \quad \mathrm{U}_{i\,min}^T\right]^T$$

The value of $\Psi_i$ is computed and communicated by downstream agents. More specifically, denote the current subsystem as $\Sigma_i$ and downstream subsystem as $\Sigma_{i+1}$. If $\Sigma_{i+1}$ receives the disturbance $\mathrm{V}_{i+1}$ from $\Sigma_i$ ($\mathrm{V}_{i+1} = \mathrm{Z}_i$), it computes the sensitivity of its cost w.r.t. to $\mathrm{V}_{i+1}$ as:

$$\gamma_{i+1} = \frac{\partial J_{i+1}}{\partial \mathrm{V}_{i+1}} = 2\left[E_i \mathrm{V}_{i+1} + T_{i+1} + N_{y,i+1}^T \mathrm{Q}_{i+1} M_{y,i+1} \mathrm{U}_{i+1}\right]$$

$$\Rightarrow \gamma_{i+1} = 2N_{y,i+1}^{T}\mathrm{Q}_{i+1}N_{y,i+1}\mathrm{V}_{i+1} - 2N_{y,i+1}^{T}\mathrm{Q}_{i+1}\mathrm{r}_{i+1} + 2N_{y,i+1}^{T}\mathrm{Q}_{i+1}F_{y,i+1}x_{0,i+1}(k) +$$

$$N_{z,i+1}^{T}\Psi_{i+1} + 2N_{y,i+1}^{T}\mathrm{Q}_{i+1}M_{y,i+1}\mathrm{U}_{i+1} + 2N_{y,i+1}^{T}\mathrm{Q}_{i+1}P_{y,i+1}\mathrm{D}_{i+1} \qquad (2.22)$$

and shares the values of $\gamma_{i+1}$ with $\Sigma_i$, i.e. $\Psi_i = \gamma_{i+1}$.

At the level of the network, this is accomplished using the defined interconnecting matrix $\boldsymbol{\Gamma}$ as:

$$\boldsymbol{\Psi} = \left[{\Psi_1}^T, {\Psi_2}^T, \cdots, {\Psi_p}^T\right]^T = \boldsymbol{\Gamma}^T\left[{\gamma_1}^T, {\gamma_2}^T, \cdots, {\gamma_p}^T\right]^T = \boldsymbol{\Gamma}^T\boldsymbol{\gamma}$$

where

$$\boldsymbol{\gamma} = \left[{\gamma_1}^T, {\gamma_2}^T, \cdots, {\gamma_p}^T\right]^T$$

The next section will show that both optimal control problems given by (2.19) and (2.21) share the same optimum point by application of the LC-DMPC algorithm. The convergence and closed-loop stability will be stated as well.

### 2.1.4 Distributed Kalman Filter

To extend the application of the NC-DMPC algorithm for output-feedback subsystems, Kalman filter is used to estimate the local states in a noisy environment for each distributed subsystem. Process (exogenous input $w_i$) and measurement noises (sensor $\varepsilon_i$) are added to local plants. Both $w_i$ and $\varepsilon_i$ are assumed to be white noise and normally distributed with zero mean and covariance $Q_{w,i}$ and $Q_{v,i}$ respectively.

$$w_i \sim N\left(0, Q_{w,i}\right), \qquad \varepsilon_i \sim N\left(0, Q_{v,i}\right)$$

Then an estimate of $x_i$ is given by:

$$\begin{aligned}\tilde{x}_{i_{k+1|k}} &= E\{x_{i_{k+1}}|y_{i_k}, y_{i_{k-1}}, \cdots, y_0\} \\ &= E\{x_{i_{k+1}}|\mathcal{Y}_i\} \\ &= E\left\{A_i x_{i_k} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k} + w_{i_k}\middle|\, \mathcal{Y}_i\right\} \\ &= A_i E\{x_{i_k}|\mathcal{Y}\} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k} + E\{w_{i_k}|\mathcal{Y}_i\} \\ &= A_i\tilde{x}_{i_{k|k}} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k} + \widetilde{w}_{i_{k|k}}\end{aligned}$$

where the notation $\tilde{x}_{i_{k|j}}$ means the estimation of the state $x_i$ at time $k$ based on the information in time $j$, $k \geq j$.

Since the mean value of $w_i$ is zero, then the noisy state estimate becomes:

$$\tilde{x}_{i_{k+1|k}} = A_i\tilde{x}_{i_k} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k}$$

This means that $x_{i_{k|k}}$ is normal distributed with mean $\tilde{x}_{i_{k|k}}$ and error covariance $\mathfrak{H}_{i_{k|k}}$:

$$x_{i_{k|k}} \sim N(\tilde{x}_{i_{k|k}}, \mathfrak{H}_{i_{k|k}})$$

Then the local Kalman filter estimates $\tilde{x}_{i_{k+1|k}}$ through:

$$\tilde{x}_{i_{k|k}} = A_i\tilde{x}_{k-1|k-1} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k} +$$

$$A_i\mathfrak{H}_{i_{k|k-1}}{C_{y,i}}^T(C_{y,i}\mathfrak{H}_{i_{k|k-1}}{C_{y,i}}^T + Q_{v,i})^{-1}(y_{i_k} - C_{y,i}\tilde{x}_{i_{k|k-1}})$$

and

$$\mathfrak{H}_{i_{k|k}} = A_i\mathfrak{H}_{i_{k|k-1}}{A_i}^T + G_iQ_{w,i}{G_i}^T -$$

$$A_i\mathfrak{H}_{i_{k|k-1}}{C_{y,i}}^T(C_{y,i}\mathfrak{H}_{i_{k|k-1}}{C_{y,i}}^T + Q_{v,i})^{-1}C_{y,i}\mathfrak{H}_{i_{k|k-1}}A_i$$

Figure (2.3) illustrates the local dynamics and MPC with Kalman filter according to the LC-DMPC algorithm.

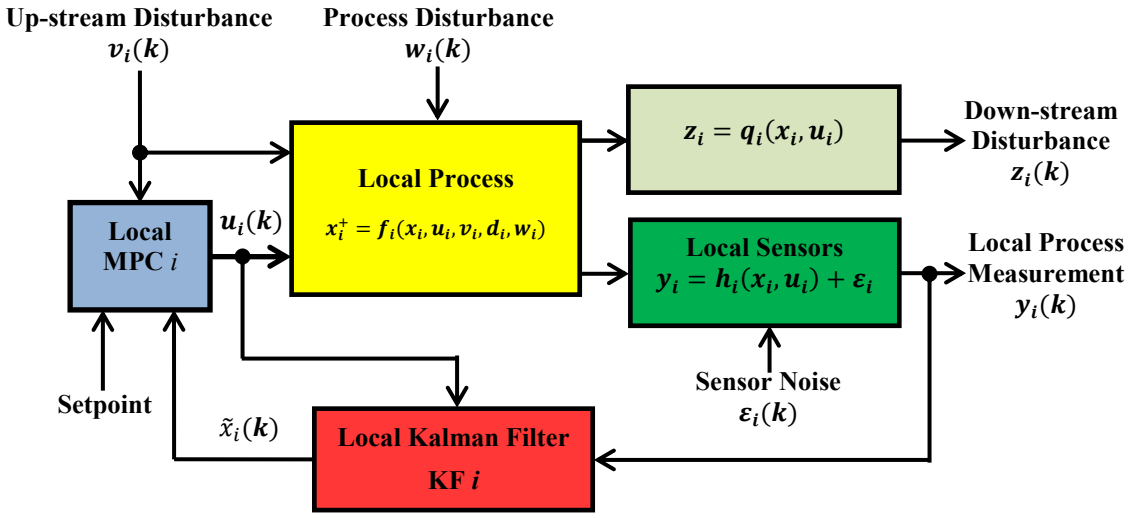



Figure 2.3: A distributed subsystem $\Sigma_i$ with a local Kalman filter

## 2.2 The LC-DMPC Main Approach

In the LC-DMPC architecture, each local controller shares two vectors per iteration with its neighbors: the vector $Z_i$ which becomes $V_{i+1}$ for downstream subsystems and vector $\gamma_i$ that appears as $\Psi_{i-1}$ in upstream costs. At same time, the controller will receive vectors $V_i$ and $\Psi_i$, which were $Z_{i-1}$ and $\gamma_{i+1}$ from upstream and downstream neighbors respectively. Figure 2.4 shows the communication structure. The index $i$ is used to show the directions of the information flow. For a sufficient number of iterations, the local controllers will end up solving the centralized problem as they will consider the influences of the local controlled plants on the network as a whole.

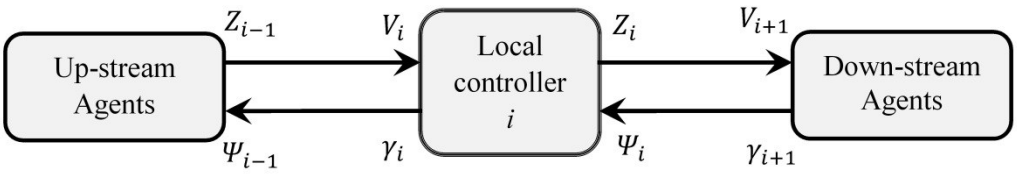



Figure 2.4: Data communication structure of an LC-DMPC local controller

### 2.2.1 The LC-DMPC Algorithm

The LC-DMPC approach divides the global optimization given in (2.19) into a number of local subproblems where each controller solves the local optimum problem given in (2.21). As a coordination to the centralized optimum solution, the local controllers need to solve an updated optimization problem frequently, and therefore an iterative approach is required. Algorithm 2.1 shows the actual implementation of the LC-DMPC approach. The algorithm starts with the initializations of some of the variables and the actual iteration begins with exchanging information between the local controllers. With updated information, the local controllers begin to solve the distributed optimization problems separately. Then the upstream and downstream neighbor vectors are updated. The algorithm continues to iterate for the specified number of iterations, and then control actions are performed on local plants. The index *j* is used for counting the iterations.

**Algorithm 2.1: The LC-DMPC Algorithm**

**Input:** Number of iterations $\mathbb{N}_i$ and initial values for states $x_{0,i}$ and error covariance $\mathfrak{H}_i$.

**Set:** initial values for $V_i = 0,\ U_i = 0,\ \Psi_i = 0$ or take previous values from last time step.

At each time step $k$ do:

**Start of Iteration:** For $j$ = 1 to $\mathbb{N}_i$ Do:

**Step 1:** Exchange current information with local agents:

$$\boldsymbol{\Psi}(j+1) = \left[\Psi_1(j+1), \cdots, \Psi_p(j+1)\right]^T = \boldsymbol{\Gamma}^T\left[\gamma_1(j), \cdots, \gamma_p(j)\right]^T = \boldsymbol{\Gamma}^T\boldsymbol{\gamma}(j)$$

$$\mathbf{V}(j+1) = \left[\mathrm{V}_1(j+1), \cdots, \mathrm{V}_p(j+1)\right]^T = \boldsymbol{\Gamma}\left[\mathrm{Z}_1(j), \cdots, \mathrm{Z}_p(j)\right]^{\boldsymbol{T}} = \boldsymbol{\Gamma}\mathbf{Z}(j)$$

**Step 2**: Solve the optimization problem given in (2.21):

$$\mathrm{U}_i{}^{QP}(j) \leftarrow \text{argmin Problem}(2.21)$$

**Step 3**: For $\beta \in [0,1)$, set:

$$\mathrm{U}_i(j+1) \leftarrow \beta U_i(j) + (1-\beta)\mathrm{U}_i{}^{QP}(j)$$

**Step 4**: Update local output disturbance:

$$\mathrm{Z}_i(j+1) \leftarrow F_{z,i}x_{0,i}(k) + M_{z,i}\mathrm{U}_i(j+1) + N_{z,i}\mathrm{V}_i(j) + P_{z,i}\mathrm{D}_i(k)$$

**Step 5**: Update sensitivity to input disturbance. Set:

$$\gamma_i(j+1) \leftarrow -2N_{y,i}^T Q_i \mathrm{r}_i(k) + 2N_{y,i}^T Q_i M_{y,i}\mathrm{U}_i(j+1) + 2N_{y,i}^T Q_i N_{y,i}\mathrm{V}_i(j)$$

$$+N_{z,i}^T \Psi_i(j) + 2N_{y,i}^T Q_i F_{y,i}x_{0,i}(k) + 2N_{y,i}^T Q_i P_{y,i}\mathrm{D}_i(k)$$

Next *j*

**End of Iteration**

**<u>Output</u>:** First value of the computed control action $\mathrm{U}_i$. (Inject this value into the local subsystem).

**Get:** new measurements for $y_i$, $u_i$, and $v_i$ then set:

$$x_{0,i} \leftarrow A_i\tilde{x}_{k-1|k-1} + B_{u,i}u_{i_k} + B_{v,i}v_{i_k} + B_{d,i}d_{i_k} +$$

$$A_i\mathfrak{H}_{i_{k|k-1}}C_{y,i}{}^T(C_{y,i}\mathfrak{H}_{i_{k|k-1}}C_{y,i}{}^T + Q_{v,i})^{-1}(y_{i_k} - C_{y,i}\tilde{x}_{i_{k|k-1}})$$

and

$$\mathfrak{H}_i \leftarrow A_i\mathfrak{H}_{i_{k|k-1}}A_i{}^T + G_iQ_{w,i}G_i{}^T -$$

$$A_i\mathfrak{H}_{i_{k|k-1}}C_{y,i}{}^T(C_{y,i}\mathfrak{H}_{i_{k|k-1}}C_{y,i}{}^T + Q_{v,i})^{-1}C_{y,i}\mathfrak{H}_{i_{k|k-1}}A_i$$

**Go to: Start of Iteration**.

**End of Algorithm 2.1**

### 2.2.2 Convergence of the LC-DMPC Algorithm

In order to show the convergence conditions for the LC-DMPC algorithm, the set of the vectors U, V, and Ψ in the iteration domain in Algorithm 2.1 is treated as a set of states of a dynamical model with index $j$. As the local controllers exchange data, the set changes and propagates in the same way as a set of states in any dynamical system. Based on this, the evolution of U, V, and Ψ under the LC-DMPC algorithm is characterized as a linear discrete-time dynamic system in the iteration domain.

**Theorem 1:** For unconstrained local MPCs, Algorithm 2.1 converges, if the all eigenvalues of the state matrix in (2.27a) are inside the unit circle.

**Proof:** The proof begins by writing the vectors U, V, and Ψ for the entire network based on how they evolve in Algorithm 2.1. From step 3 in Algorithm 2.1, the network control actions are:

$$\mathbf{U}(j+1) = \beta\mathbf{U}(j) + (1-\beta)\mathbf{U}^{\mathbf{QP}}(j) \tag{2.23}$$

where

$$\mathbf{U}^{\mathbf{QP}} = \left[\mathrm{U}_1{}^{QP^T}\ \mathrm{U}_2{}^{QP^T} \cdots\ \mathrm{U}_p{}^{QP^T}\right]^T$$

The disturbance network input can be formulated from step 1 and (2.17) as:

$$\mathbf{V}(j+1) = \mathbf{\Gamma Z}(j) = \mathbf{\Gamma}[\mathbf{F_z X_0}(k) + \mathbf{M_z U}(j+1) + \mathbf{N_z V}(j) + \mathbf{P_z D}(k)] \tag{2.24}$$

Finally, from step 5 and (2.22), the sensitivity of all local cost functions are:

$$\mathbf{\Psi}(j+1) = \mathbf{\Gamma}^T\boldsymbol{\gamma}(j)$$

$$\mathbf{\Psi}(j+1) = \mathbf{\Gamma}^T\big[-2\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{r}(k) + 2\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{M_y U}(j+1) + 2\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{N_y V}(j) + \mathbf{N}_\mathbf{z}^T\mathbf{\Psi}(j) +$$

$$2\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{F_y X}_0(k) + 2\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{P_y D}(k)\big] \tag{2.25}$$

Now, by substituting (2.23) into (2.24) and (2.25), the resulted equations with (2.23) can be written in the following linear discrete dynamic state-space format:

$$\begin{bmatrix}\mathbf{U}(j+1)\\ \mathbf{V}(j+1)\\ \mathbf{\Psi}(j+1)\end{bmatrix}=\begin{bmatrix}\beta I & 0 & 0\\ \beta\mathbf{\Gamma}\mathbf{M_z} & \mathbf{\Gamma}\mathbf{N_z} & 0\\ 2\beta\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{M_y} & 2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{N_y} & \mathbf{\Gamma}^T\mathbf{N_z^T}\end{bmatrix}\begin{bmatrix}\mathbf{U}(j)\\ \mathbf{V}(j)\\ \mathbf{\Psi}(j)\end{bmatrix}+$$

$$\begin{bmatrix}(1-\beta)I\\ (1-\beta)\mathbf{\Gamma}\mathbf{M_z}\\ 2(1-\beta)\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{M_y}\end{bmatrix}\mathbf{U^{QP}}(j)+\begin{bmatrix}0 & 0 & 0\\ \mathbf{\Gamma}\mathbf{F_z} & \mathbf{\Gamma}\mathbf{P_z} & 0\\ 2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{F_y} & 2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{P_y} & -2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\end{bmatrix}\begin{bmatrix}\mathbf{X}_0(k)\\ \mathbf{D}(k)\\ \mathbf{r}(k)\end{bmatrix} \tag{2.26}$$

For the unconstrained case, the local optimal control action at iteration $j$ is given by:

$$\mathrm{U}_i{}^{QP}(j)=\left[\mathrm{S}_i+M_{y,i}^T\mathrm{Q}_iM_{y,i}\right]^{-1}\left[M_{y,i}^T\mathrm{Q}_i\mathrm{r}_i(k)-0.5M_{z,i}^T\Psi_i(j)-M_{y,i}^T\mathrm{Q}_iN_{y,i}\mathrm{V}_i(j)-\right.$$

$$\left.M_{y,i}^T\mathrm{Q}_iF_{y,i}x_{0,i}(k)-M_{y,i}^T\mathrm{Q}_iP_{y,i}D_i(k)\right]$$

By stacking all local solutions, the network optimum solution is:

$$\mathbf{U^{QP}}(j)=\left[\mathbf{S}+\mathbf{M_y^T}\mathbf{Q}\mathbf{M_y}\right]^{-1}\left[\mathbf{M_y^T}\mathbf{Q}\mathbf{r}(k)-0.5\mathbf{M_z^T}\mathbf{\Psi}(j)-\mathbf{M_y^T}\mathbf{Q}\mathbf{N_y}\mathbf{V}(j)-\mathbf{M_y^T}\mathbf{Q}\mathbf{F_y}\mathbf{X}_0(k)-\right.$$

$$\left.\mathbf{M_y^T}\mathbf{Q}\mathbf{P_y}\mathbf{D}(k)\right]$$

and by substituting for $\mathbf{U^{QP}}$(j) in (2.26) with above equation, the new state space representation would be:

$$\begin{bmatrix}\mathbf{U}(j+1)\\ \mathbf{V}(j+1)\\ \mathbf{\Psi}(j+1)\end{bmatrix}=\begin{bmatrix}\beta I & -(1-\beta)\mathbf{\Lambda}\mathbf{M_y^T}\mathbf{Q}\mathbf{N_y} & -0.5(1-\beta)\mathbf{\Lambda}\mathbf{M_z^T}\\ \mathbf{\Gamma}\mathbf{M_z}\beta & \mathbf{\Gamma}\left(\mathbf{N_z}-(1-\beta)\mathbf{M_z}\mathbf{\Lambda}\mathbf{M_y^T}\mathbf{Q}\mathbf{N_y}\right) & -0.5(1-\beta)\mathbf{\Gamma}\mathbf{M_z}\mathbf{\Lambda}\mathbf{M_z^T}\\ 2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\mathbf{M_y}\beta & 2\mathbf{\Gamma}^T\mathbf{N_y^T}\mathbf{Q}\left(I-(1-\beta)\mathbf{M_y}\mathbf{\Lambda}\mathbf{M_y^T}\mathbf{Q}\right)\mathbf{N_y} & \mathbf{\Gamma}^T\left(\mathbf{N_z^T}-(1-\beta)\mathbf{N_y^T}\mathbf{Q}\mathbf{M_y}\mathbf{\Lambda}\mathbf{M_z^T}\right)\end{bmatrix}\begin{bmatrix}\mathbf{U}(j)\\ \mathbf{V}(j)\\ \mathbf{\Psi}(j)\end{bmatrix}+$$

$$\begin{bmatrix} -(1-\beta)\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q}\mathbf{F}_\mathbf{y} & -(1-\beta)\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q}\mathbf{P}_\mathbf{y} \\ \boldsymbol{\Gamma}\mathbf{F}_\mathbf{z} - (1-\beta)\boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q}\mathbf{F}_\mathbf{y} & \boldsymbol{\Gamma}\mathbf{P}_\mathbf{z} - (1-\beta)\boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q}\mathbf{P}_\mathbf{y} \\ 2\boldsymbol{\Gamma}^T\mathbf{N}_y^T\big(I-(1-\beta)\mathbf{Q}\mathbf{M}_\mathbf{y}\boldsymbol{\Lambda}\mathbf{M}_y^T\big)\mathbf{Q}\mathbf{F}_\mathbf{y} & 2\boldsymbol{\Gamma}^T\mathbf{N}_\mathbf{y}^T\big(I-(1-\beta)\mathbf{Q}\mathbf{M}_\mathbf{y}\boldsymbol{\Lambda}\mathbf{M}_y^T\big)\mathbf{Q}\mathbf{P}_\mathbf{y} \end{bmatrix}$$

$$\begin{bmatrix} (1-\beta)\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q} \\ (1-\beta)\boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\boldsymbol{\Lambda}\mathbf{M}_y^T\mathbf{Q} \\ 2\boldsymbol{\Gamma}^T\mathbf{N}_y^T\left((1-\beta)\mathbf{Q}\mathbf{M}_\mathbf{y}\boldsymbol{\Lambda}\mathbf{M}_y^T - I\right)\mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{X}_0(k) \\ \mathbf{D}(k) \\ \mathbf{r}(k) \end{bmatrix} \tag{2.27a}$$

with $\boldsymbol{\Lambda} = \left[\mathbf{S} + \mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{M}_\mathbf{y}\right]^{-1}$.

Eigenvalues of (2.27a) determines the convergence of Algorithm 2.1. To converge, the eigenvalues of (2.27a) have to be inside the unit circle. The vectors $\mathbf{X}_0, \mathbf{D}$ and $\mathbf{r}$ appear as disturbance signals in both (2.26) and the new state space form. For constrained local MPCs, the convergence of Algorithm 2.1 is also ensured through stability of matrix (2.27a) where the free (design) variables are $\beta, \mathbf{Q}$ and $\mathbf{S}$. It is sufficient only to check the eigenvalues of (2.27) without solving any additional systemwide problems for convergence of Algorithm 2.1.

**Remark:** With the scalar convex combination $\beta$ (same value used for all subsystems), (2.27a) can be expressed as:

$$\begin{bmatrix} 0 & -\mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y} & -0.5\mathbf{X}\mathbf{M}_\mathbf{z}^T \\ 0 & \boldsymbol{\Gamma}\big(\mathbf{N}_\mathbf{z} - \mathbf{M}_\mathbf{z}\mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y}\big) & -0.5\boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\mathbf{X}\mathbf{M}_\mathbf{z}^T \\ 0 & 2\boldsymbol{\Gamma}^T{\mathbf{N}_\mathbf{y}}^T\mathbf{Q}\mathbf{N}_\mathbf{y} - 2\boldsymbol{\Gamma}^T{\mathbf{N}_\mathbf{y}}^T\mathbf{Q}\mathbf{M}_\mathbf{y}\mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y} & \boldsymbol{\Gamma}^T{\mathbf{N}_\mathbf{z}}^T - \boldsymbol{\Gamma}^T{\mathbf{N}_\mathbf{y}}^T\mathbf{Q}\mathbf{M}_\mathbf{y}\mathbf{X}\mathbf{M}_\mathbf{z}^T \end{bmatrix} +$$

$$\beta \begin{bmatrix} I & \mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y} & 0.5\mathbf{X}\mathbf{M}_\mathbf{z}^T \\ \boldsymbol{\Gamma}\mathbf{M}_\mathbf{z} & \boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y} & 0.5\boldsymbol{\Gamma}\mathbf{M}_\mathbf{z}\mathbf{X}\mathbf{M}_\mathbf{z}^T \\ 2\boldsymbol{\Gamma}^T\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{M}_\mathbf{y} & 2\boldsymbol{\Gamma}^T\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{M}_\mathbf{y}\mathbf{X}\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N}_\mathbf{y} & \boldsymbol{\Gamma}^T\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{M}_\mathbf{y}\mathbf{X}\mathbf{M}_\mathbf{z}^T \end{bmatrix} \tag{2.27b}$$

where $\beta$ can be used to adjust the eigenvalues of the state matrix (pole-placement). However, as $0 \leq \beta < 1$ one should not expect to place the eigenvalues anywhere. Therefore both equations (2.27a) and (2.27b) can be used to tune the values of the weighted matrices $\mathbf{Q}$ and $\mathbf{S}$ and the convex combination variable $\beta$.

Checking the convergence for Algorithm 2.1 with (2.27a) requires a centralized monitoring and all subsystems' dynamics and cost function information which is impractical for real large-scale applications. In chapter IV, we present three different convergence conditions that require few or no global information. These conditions depend on the dissipativity of the local information sharing dynamics.

### 2.2.3 Closed-Loop Stability of the LC-DMPC Algorithm

The closed-loop stability of the local subsystems with Algorithm 2.1 is presented in Theorem 2. But before stating the theorem, some assumptions are introduced as follows:

*Assumption I*: Algorithm 2.1 converges, i.e. $|\lambda(\text{state matrix in 2.27a})| < 1$.

*Assumption II*: Algorithm 2.1 has sufficient iterations at each sampling index $k$ which means that (2.27a) is allowed to converge to the optimum point at that sampling index.

*Assumption III*: Each pair $[A_i, B_{u,i}]$ is controllable $\forall\ \Sigma_i \in p$.

**Theorem 2:** Under assumptions I through III, and for a feasible initial condition $x_{0,i}(k)$, the local closed-loop subsystem $\Sigma_i$ with Algorithm 2.1 is stable with a sufficiently long horizon.

**Proof:** The proof is divided into two parts. The first part shows that the LC-DMPC algorithm convergences to the same optimal point of the centralized solution. While the

second part displays the stabilization property of this optimal point for sufficiently long horizons.

**Part one**: Approaching the centralized optimum point:

The Karush-kuhn-Tucker (KKT) conditions are the first-order necessary optimality conditions; thus, if $\boldsymbol{\ell}$ is the Lagrange multiplier for (2.19), then the Lagrangian is given as:

$$\boldsymbol{\mathcal{L}}(\mathbf{U}, \boldsymbol{\ell}) = \mathbf{U}^T\mathbf{H}\mathbf{U} + 2\mathbf{U}^T\mathbf{F} - \boldsymbol{\ell}^T(\mathbf{A_{iq}}\mathbf{U} - \mathbf{B_{iq}})$$

then first order KKT conditions are:

$$2\mathbf{H}\mathbf{U}^{\mathbf{QP}} + 2\mathbf{F} - \boldsymbol{\ell}^{*T}\mathbf{A_{iq}} = 0 \tag{2.28a}$$

$$\mathbf{A_{iq}}\mathbf{U}^{\mathbf{QP}} - \mathbf{B_{iq}} \leq 0 \tag{2.28b}$$

$$\boldsymbol{\ell}^* \geq 0 \tag{2.28c}$$

$$\boldsymbol{\ell}_j\left(\mathbf{A}^{\boldsymbol{T}}_{\mathbf{iq},j}\mathbf{U}^{\mathbf{QP}} - \mathbf{B_{iq},}_j\right) = 0 \tag{2.28d}$$

where the subscript $j$ denotes the $j^{\text{th}}$ row (active constraints).

Condition (2.28a) with the systemwide dynamics can be written as:

$$2\left[\mathbf{S} + (\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z})^T\mathbf{Q}(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z})\right]\mathbf{U}^{\mathbf{QP}} +$$

$$2\left[\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\left(\left(\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z}\right)\mathbf{X_0}(k) + \left(\mathbf{P_y} + \mathbf{N_y}\mathbf{W}\mathbf{P_z}\right)\mathbf{D}\right) -\right.$$

$$\left.\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\mathbf{r}(k)\right] - \boldsymbol{\ell}^{*T}\mathbf{A_{iq}} = 0 \tag{2.28e}$$

Also let $\ell_i$ be the Lagrange multiplier for (2.21) then:

$$\mathcal{L}_i(U_i, \ell_i) = \mathrm{U}_i^T H_i U_i + 2\mathrm{U}_i^T F_i + \mathrm{V}_i^T E_i V_i + 2\mathrm{V}_i^T T_i - \ell_i^T(A_{i_{iq}}U_i - B_{i_{iq}})$$

Then the first order KKT conditions are:

$$2H_i U_i^{QP} + 2F_i - \ell_i^{*T} A_{i_{iq}} = 0 \tag{2.29a}$$

$$A_{i_{iq}} U_i^{QP} - B_{i_{iq}} \leq 0 \tag{2.29b}$$

$$\ell_i^* \geq 0 \tag{2.29c}$$

$$\ell_j \left( A_{i_{iq,j}}^T \mathrm{U}_i^{QP} - B_{i_{iq,j}} \right) = 0 \tag{2.29d}$$

Once again condition (2.29a) with local $\Sigma_i$ dynamics can be written as:

$$2\left[S_i + M_{y,i}^T \mathrm{Q}_i M_{y,i}\right] \mathrm{U}_i^{QP} + 2\left[M_{y,i}^T Q_i F_{y,i} x_{0,i}(k) + 0.5 M_{z,i} \Psi_i + M_{y,i}^T Q_i N_{y,i} V_i + \right.$$

$$\left. M_{y,i}^T Q_i P_{y,i} \mathrm{D}_i(k) - M_{y,i}^T Q_i r_i \right] - \ell_i^{*T} A_{i_{iq}} = 0 \tag{2.29e}$$

Stacking condition (2.29e) for all subsystems:

$$2\left[\mathbf{S} + \mathbf{M_y^T Q M_y}\right] \mathbf{U^{QP}} + 2\left[\mathbf{M_y^T Q F_y X_0}(k) + 0.5 \mathbf{M_z \Psi} + \mathbf{M_y^T Q N_y V} + \mathbf{M_y^T Q P_y D} - \right.$$

$$\left. \mathbf{M_y^T Q r}(k) \right] - \boldsymbol{\ell}^{*T} \mathbf{A_{iq}} = 0 \tag{2.29f}$$

And similarly for conditions (2.29b) through (2.29d):

$$\mathbf{A_{iq} U^{QP} - B_{iq}} \leq 0 \tag{2.30}$$

$$\boldsymbol{\ell}^* \geq 0 \tag{2.31}$$

$$\boldsymbol{\ell_j} \left( \mathbf{A_{iq,j}}^{\boldsymbol{T}} \mathbf{U^{QP}} - \mathbf{B_{iq,j}} \right) = 0 \tag{2.32}$$

Equations (2.30), (2.31) and (2.32) are the same as conditions (2.28b), (2.28c), and (2.28d). The left task now is to show that condition (2.29f) approaches (2.28e) by Algorithm 2.1.

Under assumptions I and II, the dynamic states of (2.26) converge to the following steady-state point:

$$\bar{\mathbf{U}} \rightarrow \mathbf{U}^{\mathbf{QP}}$$

$$\bar{\mathbf{V}} \rightarrow \mathbf{W}(\mathbf{F_z X}(k) + \mathbf{M_z U}^{\mathbf{QP}}) \tag{2.33}$$

$$\bar{\mathbf{\Psi}} \rightarrow 2[I - \mathbf{\Gamma}^T \mathbf{N}_\mathbf{z}^T]^{-1} \mathbf{\Gamma}^T \mathbf{N}_\mathbf{y}^T \mathbf{Q}\big[\mathbf{M_y U}^{\mathbf{QP}} + \mathbf{N_y W}(\mathbf{F_z X_0}(k) + \mathbf{M_z U}^{\mathbf{QP}}) + \mathbf{F_y X_0}(k) +$$

$$\mathbf{P_y D}(k) - \mathbf{r}(k)\big]$$

By substituting (2.33) into (2.29f), the result is written be:

$$2\big[\mathbf{S} + \mathbf{M}_\mathbf{y}^T \mathbf{Q M_y}\big]\mathbf{U}^{\mathbf{QP}} + 2\big[\mathbf{M}_\mathbf{y}^T \mathbf{Q F_y X_0}(k) - \mathbf{M}_\mathbf{y}^T \mathbf{Q r}(k) + \mathbf{M}_\mathbf{y}^T \mathbf{Q N_y W}(\mathbf{F_z X_0}(k) + \mathbf{M_z U}^{\mathbf{QP}})$$

$$+\mathbf{M}_\mathbf{y}^T \mathbf{Q P_y D}$$

$$+\mathbf{M}_\mathbf{z}^T [I - \mathbf{\Gamma}^T \mathbf{N}_\mathbf{z}^T]^{-1} \mathbf{\Gamma}^T \mathbf{N}_\mathbf{y}^T \mathbf{Q}\big[\mathbf{M_y U}^{\mathbf{QP}} + \mathbf{N_y W}(\mathbf{F_z X_0}(k) + \mathbf{M_z U}^{\mathbf{QP}}) + \mathbf{F_y X_0}(k)$$

$$+\mathbf{P_y D}(k) - \mathbf{r}(k)\big] - \boldsymbol{\ell}^{*T} \mathbf{A_{iq}} = 0 \tag{2.34}$$

Note: With the following matrix identity:

$$(I + AB)^{-1} A = A(I + BA)^{-1}$$

The matrix $\mathbf{W}$ can be written as following:

$$\mathbf{W} = (I - \mathbf{\Gamma N_z})^{-1} \mathbf{\Gamma} = \mathbf{\Gamma}(I - \mathbf{N_z \Gamma})^{-1}$$

Then the transform of the matrix $\mathbf{W}$ can be expressed as:

$$\mathbf{W}^T = [\mathbf{\Gamma}(I - \mathbf{N_z \Gamma})^{-1}]^T = (I - \mathbf{\Gamma}^T \mathbf{N}_\mathbf{z}^T)^{-1} \mathbf{\Gamma}^T$$

Using the above matrix identity and rearranging some terms in (2.34), the final result would be:

$$2\big[\mathbf{S} + (\mathbf{M_y} + \mathbf{N_y W M_z})^T \mathbf{Q}(\mathbf{M_y} + \mathbf{N_y W M_z})\big]\mathbf{U}^{\mathbf{QP}} +$$

$$2\Big[\big(\mathbf{M_y} + \mathbf{N_y W M_z}\big)^T \mathbf{Q}\Big(\big(\mathbf{F_y} + \mathbf{N_y W F_z}\big)\mathbf{X_0}(k) + \big(\mathbf{P_y} + \mathbf{N_y W P_z}\big)\mathbf{D}\Big) -$$

$$\big(\mathbf{M_y} + \mathbf{N_y W M_z}\big)^T \mathbf{Q r}(k)\Big] - \boldsymbol{\ell}^{*T} \mathbf{A_{iq}} = 0 \tag{2.35}$$

Equation (2.35) is same as (2.28e), i.e. the solution of Algorithm 2.1 converges to the centralized optimum solution.

**Part two:** Stabilizing property of the optimum point:

In the second part of the proof, two theorems stated in [56] will be restated briefly. Theorems 5.2 and 5.3 in [56] imply that for sufficiently large horizons, the solution applied by the centralized solution in a receding horizon fashion is stabilizing. The main idea of this receding horizon stabilization property is to serve the cost function in (2.19) as a Lyapunov function. To accomplish this property, assumption III must be satisfied in addition to observability of $\left[{q_i}^{1/2}, A_i\right] \forall\, \Sigma_i \in p$ which is already satisfied through the assumptions of $q_i > 0$ and $s_i > 0$.

As the LC-DMPC Algorithm under assumptions I and II converges to the same centralized MPC optimum, then it also stabilizes the network for sufficiently large horizons.

**Remark**: The matrix $[I - \mathbf{\Gamma}^T \mathbf{N}_\mathbf{z}^T]$ is invertible and the proof is shown below:

If the rank of an $n \times n$ square matrix is less than $n$, the matrix does not have an inverse, i.e. for a matrix to have an inverse, it has to be a full-rank matrix.

Since the locally defined $N_{z,i}$ matrix given by:

$$N_{z,i} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ C_{z,i}B_{v,i} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ C_{z,i}{A_i}^{N_p-2}B_{v,i} & C_{z,i}{A_i}^{N_p-3}B_{v,i} & \cdots & C_{z,i}B_{v,i} & 0 \end{bmatrix}$$

is not a full rank matrix (has a zero-row), then global defined matrix: $\mathbf{N_z} = diag(N_{z,1}, \cdots N_{z,p})$ is not a full rank matrix either. According to the definition of the

interconnecting matrix $\boldsymbol{\Gamma}$, it is a full rank matrix (each row or column has an element with a value of 1 and the rest elements are zeros). Thus the combined matrix $\boldsymbol{\Gamma}^T \mathbf{N}_\mathbf{z}^T$ is not a full rank matrix as well. Therefore the matrix $[I - \boldsymbol{\Gamma}^T \mathbf{N}_\mathbf{z}^T]$ is full rank (the identity matrix $I$ realizes the full rank property) which has a determinant, therefore it is invertible.

### 2.2.4 Difference between the Centralized and Distributed Costs

In order to converge to the global minimum, the LC-DMPC approach assigns each local controller to solve a local optimization problem with a cost function that is different than the centralized defined cost. Therefore, the summation of the local costs is not equal to the systemwide objective cost. In the proof of theorem 2, it is shown that the centralized and all local distributed optimization problems are sharing the same optimum point under some assumptions. Thus, both solutions are sharing same optimum but with different optimum cost values and this was the motivation to write this section. For simplicity, in this subsection we assumed that all subsystem dynamics are not subjected to the disturbance $d_i(k)$.

Recall that the centralized cost function is given as:

$$J_{centralized} = \mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{U}^T \mathbf{S} \mathbf{U}$$

which can be written as:

$$J_{centralized} = \mathbf{r}^T \mathbf{Q} \mathbf{r} + \mathbf{Y}^T \mathbf{Q} \mathbf{Y} - \mathbf{2} \mathbf{Y}^T \mathbf{Q} \mathbf{r} + \mathbf{U}^T \mathbf{Q} \mathbf{U} \tag{2.36}$$

and by substituting the network regulated output (2.16):

$$J_{centralized} = \mathbf{U}^T\left[\mathbf{S} + (\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z})^T\mathbf{Q}(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z})\right]\mathbf{U} +$$

$$2\mathbf{U}^T\left[\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\left((\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z})\mathbf{X_0}(k)\right) - (\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z})^T\mathbf{Q}\mathbf{r}(k)\right] +$$

$$\mathbf{X}_0^T(k)\left[(\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z})^T\mathbf{Q}(\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z})\right]\mathbf{X_0}(k) -$$

$$2\mathbf{r}^T(k)\mathbf{Q}\left[\mathbf{F_y}\mathbf{X_0}(k) + \mathbf{N_y}\mathbf{W}\mathbf{F_z}\mathbf{X_0}(k) - 0.5\mathbf{r}(k)\right] \tag{2.37}$$

Also recall that the local cost for subsystem $\Sigma_i$ is given by:

$$J_i = \mathrm{r}_i^T\mathrm{Q}_i\mathrm{r}_i + \mathrm{Y}_i^T\mathrm{Q}_i\mathrm{Y}_i - 2\mathrm{Y}_i^T\mathrm{Q}_i\mathrm{r}_i + \mathrm{U}_i^T\mathrm{Q}_i\mathrm{U}_i + \Psi_i^T\mathrm{Z}_i \tag{2.38}$$

Stacking (2.38) for all local costs:

$$J_{distributed} = \sum_{i=1}^{p} J_i = \mathbf{r}^T\mathbf{Q}\mathbf{r} + \mathbf{Y}^T\mathbf{Q}\mathbf{Y} - 2\mathbf{Y}^T\mathbf{Q}\mathbf{r} + \mathbf{U}^T\mathbf{Q}\mathbf{U} + \mathbf{\Psi}^T\mathbf{Z} \tag{2.39}$$

By introducing the network dynamics (2.16) and (2.17) into (2.39), the result is written as:

$$\mathbf{J}_{distributed} = \mathbf{r}^T(k)\mathbf{Q}\mathbf{r}(k) + \mathbf{U}^T\left[\mathbf{S} + \mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{M_y}\right]\mathbf{U}$$

$$+ 2\mathbf{U}^T\left[\mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{F_y}\mathbf{X_0}(k) + \mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{N_y}\mathbf{V} + \mathbf{0.5}\mathbf{M}_\mathbf{z}^T\mathbf{\Psi} - \mathbf{M}_\mathbf{y}^T\mathbf{Q}\mathbf{r}(k)\right] +$$

$$\mathbf{V}^T\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{N_y}\mathbf{V} + 2\mathbf{V}^T\left[\mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{F_y}\mathbf{X_0}(k) + \mathbf{0.5}\mathbf{N}_\mathbf{z}^T\mathbf{\Psi} - \mathbf{N}_\mathbf{y}^T\mathbf{Q}\mathbf{r}(k)\right] +$$

$$\mathbf{X}_0^T(k)\mathbf{F}_\mathbf{y}^T\mathbf{Q}\mathbf{F_y}\mathbf{X_0}(k) + 2\mathbf{X}_0^T(k)\left[\mathbf{0.5}\mathbf{F}_\mathbf{z}^T\mathbf{\Psi} - \mathbf{F}_\mathbf{y}^T\mathbf{Q}\mathbf{r}(k)\right] \tag{2.40}$$

Equation (2.40) is a function of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{\Psi}$, and $\mathbf{X_0}(k)$. The network initial condition $\mathbf{X_0}(k)$ will be constant during the iterations while (2.33) gives the steady-state values of the other variables. Thus, when Algorithm 2.1 converges equation (2.40) becomes:

$$J_{distributed} = \mathbf{U}^T\left[\mathbf{S} + \left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right) + 2\mathbf{M}_\mathbf{z}^T\mathbf{L}\mathbf{M_y} +\right.$$

$$\left.2\mathbf{M}_\mathbf{z}^T\mathbf{L}\mathbf{N_y}\mathbf{W}\mathbf{M_y}\right]\mathbf{U} +$$

$$2\mathbf{U}^T\left[\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\left(\left(\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z}\right)\mathbf{X_0}(k)\right) - \left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)^T\mathbf{Q}\mathbf{r}(k) -\right.$$

$$\mathbf{M_z^T L r}(k) + \left(\mathbf{M_z^T L N_y W F_z} + \mathbf{M_z^T L F_y} + \mathbf{M_y^T L^T F_z} + \mathbf{M_z^T W^T N_y^T L^T F_y}\right)\mathbf{X_0}(k)\Big] +$$

$$2\mathbf{X}_0^T(k)\left[\left(\mathbf{F_y} + \mathbf{N_y W F_z}\right)^T \mathbf{Q}\left(\mathbf{F_y} + \mathbf{N_y W F_z}\right) + \mathbf{F_z^T L N_y W F_z} + \mathbf{F_z^T L F_y}\right]\mathbf{X_0}(k) -$$

$$2\mathbf{X}_0^T(k)\mathbf{F_z^T L r}(k) - 2\mathbf{r}^T(k)\mathbf{Q}\left[\mathbf{F_y X_0}(k) + \mathbf{N_y W F_z X_0}(k) - 0.5\mathbf{r}(k)\right] \quad (2.41)$$

where $\mathbf{L} = \mathbf{W}^T[I - \mathbf{N_z^T \Gamma}^T]^{-1}\mathbf{N_y^T Q}$

Then with (2.37) and (2.41) the distributed cost can be written as:

$$J_{distributed} = J_{centralized} + \Delta J \quad (2.42)$$

where

$$\Delta J = 2\mathbf{U}^T\left[\mathbf{M_z^T L M_y} + \mathbf{M_z^T L N_y W M_y}\right]\mathbf{U} + 2\mathbf{U}^T\Big[\left(\mathbf{M_z^T L N_y W F_z} + \mathbf{M_z^T L F_y} +$$

$$\mathbf{M_y^T L^T F_z} + \mathbf{M_z^T W^T N_y^T L^T F_y}\right)\mathbf{X_0}(k) - \mathbf{M_z^T L r}(k)\Big] +$$

$$2\mathbf{X}_0^T(k)\left[\left(\mathbf{F_y} + \mathbf{N_y W F_z}\right)^T \mathbf{Q}\left(\mathbf{F_y} + \mathbf{N_y W F_z}\right) + \mathbf{F_z^T L N_y W F_z} + \mathbf{F_z^T L F_y}\right]\mathbf{X_0}(k) -$$

$$2\mathbf{X}_0^T(k)\mathbf{F_z^T L r}(k)$$

### 2.3 Laguerre Functions for Local Problems and Vectors

In Theorem 2, the closed-loop stability is proven assuming appropriately long horizons for the centralized problem. This assumption is distributed for the local problems as well. With long horizons, the local online optimization problem given in (2.21) will have a large number of variables to be optimized which may need additional computational time. In addition, the agents will exchange vectors with large lengths resulting in larger communication requirements. In order to achieve real-time control, and to apply the algorithm for faster systems, the local controllers should provide control inputs faster than the sampling rate of the local plants and share data in a more efficient way. In this

work, Laguerre functions are used to approximate and reduce the size of the distributed optimization problems and to diminish the length of the exchanged vectors.

### 2.3.1 Laguerre Functions for Local Control Actions

In this subsection, the Laguerre functions are used to approximate and reduce the size of the LC-DMPC distributed problems by parameterizing the local control actions via orthogonal Laguerre networks with the variable *a* as a pole for the discrete Laguerre sequences. Many of the following equations are taken from [64]. The control actions are expressed using $N_i$ discrete Laguerre functions as:

$$u_i(k+m) = \sum_{j=1}^{N_i} c_j(k) l_j(m) \tag{2.43}$$

where $m = 0,1,\cdots,N_p$ and $c_j$ and $j = 1,2,\cdots,N_i$ are coefficients to be computed and the set of Laguerre functions in discrete format, respectively. The z-transform of these functions is given as:

$$L_{N_i}(z,a_i) = \frac{\sqrt{(1-a_i^2)}}{(1-a_i z^{-1})}\left(\frac{z^{-1}-a_i}{1-a_i z^{-1}}\right)^{N_i-1} \qquad 0 \le a_i < 1 \tag{2.44}$$

The state-space model of the Laguerre sequences are computed from (2.44) as:

$$L_i(k+1) = A_{L_i} L_i \tag{2.45}$$

where:

$$A_{L_i} = \begin{bmatrix} a_i & 0 & 0 & \cdots & 0 \\ \varrho_i & a_i & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ (-1)^{N_i-2} a_i^{N_i-2} \varrho_i & (-1)^{N_i-3} a_i^{N_i-3} \varrho_i & \cdots & \varrho_i & a_i \end{bmatrix}$$

$$L_i(k) = [l_1(k)\ \ l_2(k)\ \cdots l_{N_i}(k)]^T$$

$$\varrho_i = 1 - a_i^2$$

and initial condition $L_i(0)^T = \sqrt{\varrho_i}[1, -a_i, a_i^2, -a_i^3, \cdots (-1)^{N_i-1} a_i^{N_i-1}]$.

Using this state-space representation, equation (2.43) can be written as:

$$u_i(k+j) = L_i(j)^T \eta_i \tag{2.46}$$

where ${\eta_i}^T = [c_1\ c_2\ \cdots\ c_{N_i}]$.

For multi-input system, each control action can be represented by a different set of Laguerre functions with different numbers $N_i$ and poles $a_i$, i.e.:

$$u_i(k) = \begin{bmatrix} L_{1_i}(k)^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_{r_{u,i}}(k)^T \end{bmatrix} \begin{bmatrix} \eta_{1_i} \\ \vdots \\ \eta_{\ r_{u,i}} \end{bmatrix} = L_{p,i}(k)^T \eta_{p,i}$$

The orthogonally property of the Laguerre functions in discrete time domain can be written as:

$$\begin{aligned} \textstyle\sum_{k=0}^{\infty} l_i(k) l_j(k) &= 0 for\ i \neq j \\ \textstyle\sum_{k=0}^{\infty} l_i(k) l_i(k) &= 1 for\ i = j \end{aligned} \tag{2.47}$$

With Laguerre parametrizations for local dynamics, equations in (2.11) and (2.12) become:

$$\begin{aligned} y_i(k+m) &= C_{y,i} A_i^m x_{0,i}(k) + \Phi_{y,i} \eta_{m,i} + \mathrm{n}_{y,i} \mathcal{V}_i(m) \\ z_i(k+m) &= C_{z,i} {A_i}^{m-1} x_{0,i}(k) + \Phi_{z,i} \eta_i + \mathrm{n}_{z,i} \mathcal{V}_i(m) + t_i \end{aligned} \tag{2.48}$$

where

$$\Phi_{y,i} = \textstyle\sum_{i=0}^{m-1} C_{y,i} {A_i}^{m-i-1} B_{u,i} L_{p,i}(i)^T$$

$$\mathrm{n}_{y,i} = \textstyle\sum_{i=0}^{m-1} C_{y,i} {A_i}^{m-i-1} B_{v,i}$$

$$\mathcal{V}_i(m) = [v(0)\ \ v(1)\ \cdots\ v(m-1)]^T$$

$$t_i = D_{z,i} L_i (m-1)^T \eta_i$$

and $\Phi_{z,i}$ and $\mathrm{n}_{z,i}$ are defined in a similar way with $C_{z,i}$ replacing $C_{y,i}$ with one step late. By introducing (2.48) into the local optimization problem defined in (2.21) and taking the advantage of the orthogonality properties (2.47), the new local optimization problem with reduced manipulated variables and $N_p$ as an upper summation limit is given as:

$$J_i = \eta_{p,i}^T \left\{ \sum_{m=1}^{N_p} \left( \Phi_{y,i}^T q_i \Phi_{y,i} \right) + s_i \right\} \eta_{p,i} + 2 {\eta_{p,i}}^T \left\{ \sum_{m=1}^{N_p} \left( \Phi_{y,i}^T q_i C_{y,i} A_i^m x_{0,i}(k) + \Phi_{y,i}^T q_i \mathrm{n}_{y,i} \mathcal{V}_i(m) + 0.5 \Phi_{z,i}^T \psi(m) - \Phi_{y,i}^T q_i \mathrm{r}_i(k) \right) \right\} + c_{0,i} \tag{2.49}$$

Subjected to:

$$\begin{bmatrix} L_{p,i}(m)^T \\ -L_{p,i}(m)^T \end{bmatrix} \eta_{p,i} \leq \begin{bmatrix} u_{i_{max}} \\ -u_{i_{min}} \end{bmatrix}, \; m = 0,1,\cdots,N_{c,i}$$

where $N_{c,i}$ is the horizon length at which the control actions are saturated, $\psi(m)$ is the shared sensitivity value at time $m$, and $c_{0,i}$ is a constant.

Because of the exponential decay properties of Laguerre functions, the control actions will be faster in the beginning of the horizon and avoid any peaks at the end of the horizon [64]. This further reduces the number of constraints for the online optimization problem of the local agents. The new optimization problem given in (2.49) will replace the one given in step 2 in Algorithm 2.1 with $\eta_{p,i}$ as new variables. This new optimization problem has smaller decision variables where simply the number of terms used to parameterize the control actions is going to be the new optimization variable.

### 2.3.2 Laguerre Functions for Local Shared Vectors

In the previous subsection, Laguerre polynomials is proposed to reparametrize the local control actions which resulted in a smaller local optimization size that can help increasing the number of iterations per sampling in Algorithm 2.1. In this section, Laguerre functions are used to parametrize the signals that are being exchanged between the local agents. This can reduce the communication loads in the Algorithm 2.1 as well.

If a signal $f(k)$ is parameterized by Laguerre sequences, independently of the value of $0 \leq a < 1$, the approximation of the signal improves as the number of Laguerre terms *N* increasers [64]. To accomplish the parameterization objective for a signal $f(k)$, Algorithm 2.2 is proposed using the given definitions and equations in the previous section.

The orthonormal property of Laguerre functions given in (2.47) enables the coefficients of Laguerre polynomials to be defined as:

$$c_i = \sum_{k=0}^{\infty} f(k) l_i(k), \quad i = 1, 2, \cdots, N \tag{2.50}$$

In Algorithm 2.2 the following $L_2$ error norm is used to compute the Laguerre coefficients as well as the pole $0 \leq a < 1$:

$$\left\| f_{param}(k) - f(k) \right\|_2 / \left\| f_{param}(k) \right\|_2 \leq \epsilon$$

where $f_{param}(k)$ is the resulted parametrized signal and $\epsilon$ is a desired accuracy for the parameterized signal.

**Algorithm 2.2: Laguerre Approximations**

**Input:** the signal $f(k)$, the desired accuracy $\epsilon$.

Set: initial values for $a_i = 0.001$, $N_i = 1$.

**Initial Computations:** Find:

Initial Laguerre functions from $a_i$ & $N_i$ (Eq. 2.45):

Set: $L_i \leftarrow$ Laguerre functions

Initial Laguerre coefficients from $f(k)$ & $L_i$ (Eq. 2.50):

Set: $c_i \leftarrow$ Laguerre coefficients

Initial parametrized signal form $c_i$ & $L_i$:

Set: $f_{param_i} \leftarrow c_i * L_i$

Initial error norm:

Set: $er\_no \leftarrow \left\| f_{param_i} - f \right\|_2 / \left\| f_{param_i} \right\|_2$

**While ($er\_no \geq \epsilon$ ) do:**

Set: $N \leftarrow N + 1;\ L \leftarrow 0;\ a \leftarrow 0.$

**While ($er\_no \geq \epsilon$) do:**

Set: $a \leftarrow a + 0.01.$

**Find:**

Laguerre functions from $a$ & $N$ (Eq. 2.45).

Set: $L \leftarrow$ Laguerre functions.

Laguerre coefficients from $f(k)$ & $L$ (Eq. 2.50).

Set: $c \leftarrow$ Laguerre coefficients.

parametrized signal form $c$ & $L$.

Set: $f_{param} \leftarrow c * L.$

Compute error norm:

Set: $er\_no \leftarrow \|f_{param} - f\|_2 / \|f_{param}\|_2$ .

**If** ($a > 1$) **Break**.

**End**

**End**

**Output:** Time constant $a$, Laguerre coefficients $c$.

**End of Algorithm 2.2**

Figure 2.5 gives the flowchart of Algorithm 2.2. When $a \approx 0$, the Laguerre sequences become a set of pulses, i.e. the signal is not parametrized. In Algorithm 2.2, for each new added $N$, the search begins with a very small value of $a$ and goes over all values of $0 \leq a < 1$. This ensures that the algorithm is not over parametrizing, i.e. length of parametrized signal is less than or equal to original signal length.

Each controller parametrizes the shared vectors ($\mathrm{Z}_i$ & $\gamma_i$) and receives the parameterized vectors ($\mathrm{V}_i$ & $\Psi_i$). Each communicated data contains only the time constant $a$ and Laguerre coefficients $c$. Once the Laguerre information of a signal is received, a local controller recovers the original, the controller recovers the original signal along the horizon using (2.45) and $f(k) = \sum_{j=1}^{N} c_j(k) l_j(m)$, where $c_j, a,$ and $N$ are contained in the received signal.

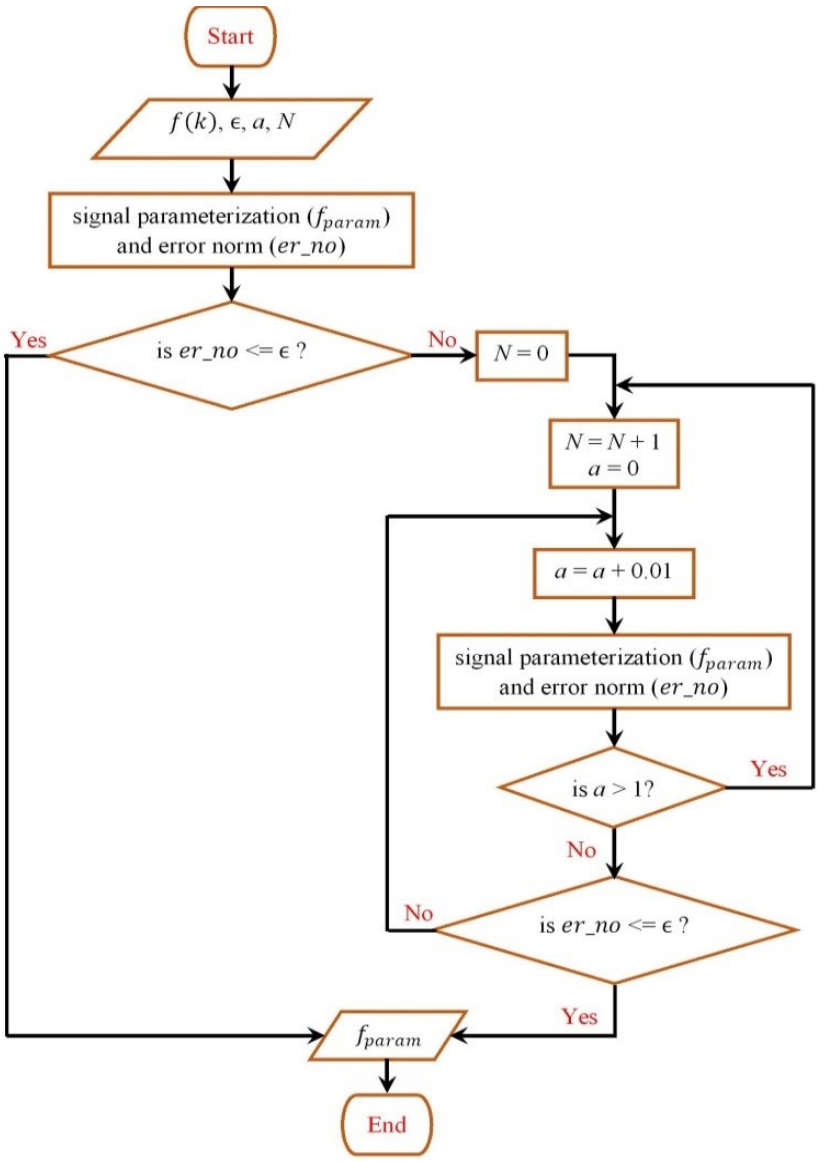



Figure 2.5: Algorithm 2.2 flowchart diagram

# 3. LC-DMPC SIMULATION USING COUPLED SIX-TANK PROCESS*

In chapter II the theory of the LC-DMPC approach is presented where two algorithms are proposed to control a network of coupled subsystems and reduce the computation and communication loads. In this chapter these algorithms are applied to a six-tank process, and the steps of designing the local interaction matrices and demonstration of the LC-DMPC approach application are discussed.

The chapter is structured as follows. First, the nonlinear differential equations that approximate the six-tank process are derived. Then the designing procedure of the local discrete subsystem models and controllers from the linearized central dynamic are detailed. Finally, using both algorithms, the six-tank process is controlled and simulated in two different cases: Full state feedback subsystems with noise-free inputs as the first case and subsystems with output feedback that work in a noisy environment as the second case.

## 3.1 Application and Implementation of the LC-DMPC Algorithm

In this section, a demonstration of the LC-DMPC approach is given by applying the approach to a coupled tank process. The section begins with description, model derivation of the six-tank plant, and application of the LC-DMPC algorithm. Section 3.2 section presents the simulation results. This part of the work mainly focuses on how to apply the proposed algorithm.

* Parts of this section are reprinted with permission from R. Jalal and B. Rasmussen, "Limited-communication distributed model predictive control for coupled and constrained linear systems," IEEE Transaction on Control System Technology, accepted, 2016.

### 3.1.1 The Six-Tank Process and Controller Design

The six-tank process is a square Multi-Input, Multi-Output (MIMO) process that consists of six interconnected water tanks arranged in three parallel interacting systems; each system has two coupled tanks in series. A schematic diagram of the process is shown in Figure 3.1. The target is to control the water levels in the lower tanks with three pumps. Pump 1 extracts water from the main basin below and pours it to tanks 1 and 5, while pump 2 pours water to tanks 2, 4, and 6. Finally tanks 3 and 5 get water from pump 3. In addition to the coupling in control actions, the outputs from upper tanks likewise disturb the states of lower ones. The output of tank 4 affects both tanks 1 and 2. Tanks 1, 2, and 3 are also being affected by the discharge of tank 5. Output of tank 6 impacts the states of tank 2 and 3, respectively. The amount of water flows from the pumps to the tanks and from upper to lower tanks can be controlled throughout the valves $V_i$, $i = 1, 2, \ldots 7$. This MIMO system has three manipulated inputs, the input voltages to pumps 1, 2 and 3, and three outputs, the water levels in tanks 1, 2, and 3. Due to the strong coupling between the tanks, states as well as control actions, the task of controlling water levels in the lower tanks is rather difficult to fulfill. Upper tanks are subjected to unmeasured disturbances and there are measurement noises as well. The pump flow rates along with the levels of the tanks are constrained physically. However, in this work only hard constraints on control actions are considered.

In literature, the used tank process usually has four interacting tanks that are coupled in the control actions only. As shown in the Figure 3.2, the upper tanks disturb the direct

lower tanks thorough the outputs, however, they do not affect the cross tanks, i.e. there is no coupling in states. The pumps, on the other hand, control the water level in the corresponding lower tanks and disturb the upper crossing tanks at the same time. To demonstrate the aspects of the proposed LC-DMPC algorithm, we added a third set of tanks to the process and increased the couplings between the tanks by allowing the outputs of the upper tanks to distract not only the corresponding lower row of tanks but also the crossing tanks. For instant, output of the tank 5 affects tanks 1, 2, and 3. Through the valves 4, 5, and 6 we can increase or cancel the couplings in the states.

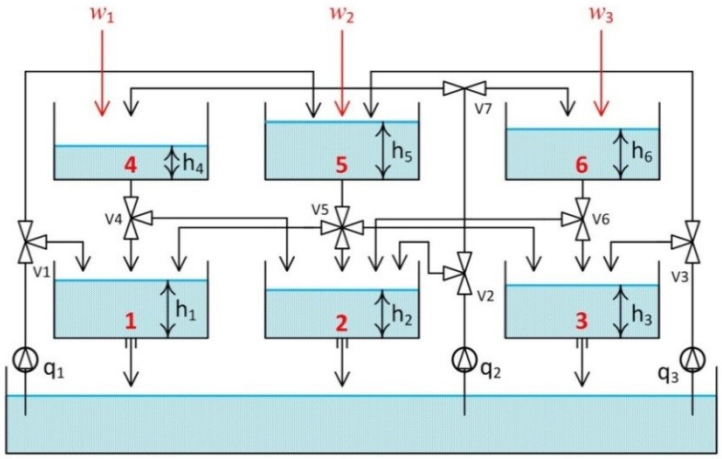



Figure 3.1: The six-tank process diagram used in chapter III

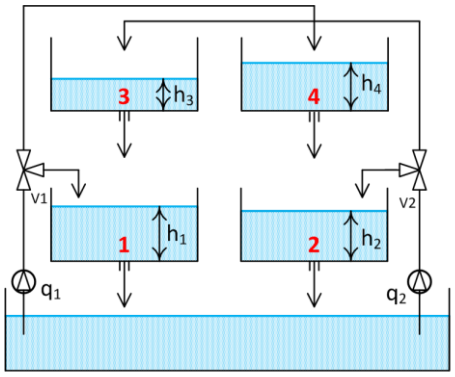



Figure 3.2: The four-tank process diagram used in literature

### 3.1.2 The Centralized Mathematical Model

The process inputs are $v_1, v_2,$ and $v_3$ (input voltages to the pumps) and the outputs are $y_1, y_2$ and $y_3$ (voltages from level measurement devices at lower tanks). From mass balances and Bernoulli's law the following differential equations approximate the dynamical model of the process:

$$\left.\begin{aligned}
\frac{dh_1}{dt} &= -\frac{\sigma_1(t)}{A_1} + \alpha_1\frac{\sigma_4(t)}{A_1} + \lambda_1\frac{\sigma_5(t)}{A_1} + \tau_1\frac{F_1}{A_1}\\
\frac{dh_2}{dt} &= -\frac{\sigma_2(t)}{A_2} + \alpha_2\frac{\sigma_4(t)}{A_2} + \lambda_2\frac{\sigma_5(t)}{A_2} + \rho_1\frac{\sigma_6(t)}{A_2} + \mu_1\frac{F_2}{A_2}\\
\frac{dh_3}{dt} &= -\frac{\sigma_3(t)}{A_3} + \lambda_3\frac{\sigma_5(t)}{A_3} + \rho_2\frac{\sigma_6(t)}{A_3} + \delta_1\frac{F_3}{A_3}\\
\frac{dh_4}{dt} &= -\frac{\sigma_4(t)}{A_4} + \mu_2\frac{F_2}{A_4} + w_1(t)\\
\frac{dh_5}{dt} &= -\frac{\sigma_5(t)}{A_5} + \tau_2\frac{F_1}{A_5} + \delta_2\frac{F_3}{A_5} + w_2(t)\\
\frac{dh_6}{dt} &= -\frac{\sigma_6(t)}{A_6} + \mu_3\frac{F_2}{A_6} + w_3(t)
\end{aligned}\right\} \qquad (3.1)$$

where $\sigma_i(t) = b_i\sqrt{2gh_i(t)}$, $i$ = 1, 2, … , 6 are the inlet and outlet flow rates and $F_j = k_j v_j$, $j$ = 1, 2, 3 are the controlled inlet flow rates form the pumps. $h_i(t)$, $b_i,$ and $\text{Å}_i$ refer to the water level, cross-section of the outlet hole, and cross-sectional area of tank $i$, respectively. $w_i(t), i = 1,2,3$ are the unmeasured disturbances. The flow parameters $\mathcal{B}, \lambda, \tau, \delta, \rho,$ and $\sigma$ determine how much water flows to the corresponding tank, therefore, $\sum_{i=1}^{2} x_i = 1$ for $x_i = \mathcal{B}_i,\ \rho_i,\ \tau_i,$ and $\delta_i$ and $\sum_{j=1}^{3} y_j = 1$ for $y_j = \lambda_j,$ and $\sigma_j$. Most of the parameter values used in (3.1) are taken from [102]. The values of the voltage constants $k_1, k_2,$ and $k_3$ for the pumps and flow parameters depend on the operating points.

For the LC-DMPC algorithm to be tested on the described six-tank process, a linear model has to be derived based on equation (3.1). The nonlinear dynamic model (3.1) is linearized around the operating point given in Table 3.1.

**Table 3.1: Physical Parameter Values for the Six-tank Process**

| **Description** | **Symbol and value** |
|---|---|
| Cross sectional area of the tanks ($cm^2$) | $Å_i$ = 28, i = 1, 3,5<br>$Å_i$ = 32, j = 2, 4, 6 |
| Cross section of the outlet hole ($cm^2$) | $b_i = 0.071$, i = 1, 3, 5<br>$b_j = 0.057$, j = 2, 4, 6 |
| Level sensor accuracy (V/cm) | $k_c = 0.5$ |
| Gravity constant ($cm/sec^2$) | 981 |
| **The Operating Point** | |
| Flow parameters | $\mathcal{B}_1 = \mathcal{B}_2 = 0.5$<br>$\lambda_1 = \lambda_2 = \lambda_3 = 0.333$<br>$\rho_1 = \rho_2 = 0.5$<br>$\tau_1 = 0.7, \tau_2 = 0.3$<br>$\sigma_1 = 0.7, \sigma_2 = \sigma_3 = 0.15$<br>$\delta_1 = 0.7, \delta_2 = 0.3$ |
| Voltage constant for pumps ($cm^3$/V. sec) | $k_1 = 3.33$, $k_2 = 3.35$, $k_3 = 3.34$ |
| Linearization level of tank $i$ (cm) ($i$ = 1, 2, …, 6) | $h_1^0 = 12.4, h_2^0 = 12.7$<br>$h_3^0 = 12.5, h_4^0 = 1.8$<br>$h_5^0 = 1.4, h_6^0 = 1.6$ |
| Linearization of control effort $u_j$ (V) ($j$ = 1, 2, 3) | $v_j^0 = 3.00$ |

By defining the deviation variables as:

$$x_i = h_i - h_i^0, \qquad i = 1, 2, \cdots, 6$$

$$u_i = v_j - v_j^0, \qquad j = 1, 2, 3$$

the following centralized continuous-time linear dynamic model can be obtained:

$$\left.\begin{array}{l} \frac{dx}{dt} = A_c x + B_u u + B_w w \\ y = C_c x \end{array}\right\} \tag{3.2}$$

where:

$$x = [x_1 \;\; x_2 \; x_3 \;\; x_4 \;\; x_5 \;\; x_6]^T$$

$$u = [u_1 \;\; u_2 \; u_3]^T$$

$$w = [w_1 \;\; w_2 \; w_3]^T$$

$$y = [y_1 \;\; y_2 \; y_3]^T$$

$$A_c = \begin{bmatrix} -1/T_1 & 0 & 0 & \alpha_1 A_4/A_1 T_4 & \lambda_1 A_5/A_1 T_5 & 0 \\ 0 & -1/T_2 & 0 & \alpha_2 A_4/A_2 T_4 & \lambda_2 A_5/A_2 T_5 & \rho_1 A_6/A_2 T_6 \\ 0 & 0 & -1/T_3 & 0 & \lambda_3 A_5/A_3 T_5 & \rho_2 A_6/A_3 T_6 \\ 0 & 0 & 0 & -1/T_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/T_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/T_6 \end{bmatrix}$$

$$B_u = \begin{bmatrix} \tau_1/A_1 & 0 & 0 \\ 0 & \mu_1/A_2 & 0 \\ 0 & 0 & \delta_1/A_3 \\ 0 & \mu_2/A_4 & 0 \\ \tau_2/A_5 & 0 & \delta_2/A_5 \\ 0 & \mu_3/A_6 & 0 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix}$$

$$C_c = [k_c I_{3\times3} \quad 0_{3\times3}]$$

where $T_i = \frac{A_i}{b_i}\sqrt{\frac{2h_i^0}{g}}, i = 1, 2, \cdots, 6.$

This new linearized model is used to implement the centralized predictive controller as well as the distributed controllers. The next section details the constructions of the LC-DMPC matrices and vectors for the local subsystems.

### 3.1.3 Proposed Subsystem Definitions

In this section the derivation of the local subsystems is detailed for the six-tank process in order to illuminate the application of the LC-DMPC algorithms. To implement the proposed algorithm, the centralized model is divided into three coupled plants. Subsystem $\Sigma_i$ consists of tanks $i$ and $i+3$, for $i = 1,2,3$. According to the upstream and downstream definition, the input and out disturbance vectors as well as the interaction matrices are derived as following.

The LC-DMPC subsystem $\Sigma_1$ is defined as:

$$\begin{aligned} \begin{bmatrix} \dot{x_1}(t) \\ \dot{x_4}(t) \end{bmatrix} &= A_{c_1} \begin{bmatrix} x_1(t) \\ x_4(t) \end{bmatrix} + B_{c_{u,1}} u_1(t) + B_{c_{v,1}} \begin{bmatrix} x_5(t) \\ u_2(t) \end{bmatrix} + B_w w_1 \\ y_1(t) &= C_{c_{y,1}} \begin{bmatrix} x_1(t) \\ x_4(t) \end{bmatrix} \end{aligned} \tag{3.3}$$

where $A_{c_1}, B_{c_{u,1}}, B_{c_{v,1}}$ and $C_{c_{y,1}}$ are easily derived from equation (3.2) while the regulated output is $x_1(t)$.

The vector that disturbs this subsystem is $[x_5(t)\ \ u_2(t)]^T$, which is coming from the upstream neighbor subsystem $\Sigma_2$, therefore:

$$v_1(t) = z_{2,1}(t) = \begin{bmatrix} x_5(t) \\ u_2(t) \end{bmatrix}$$

Also, this subsystem affects the downstream neighbor subsystem $\Sigma_2$ throughout state $x_4(t)$ and control action $u_1(t)$, therefore, the disturbance output for subsystem $\Sigma_2$ is $z_1(t) = [x_4(t)\ \ u_1(t)]^T$ and this can be realized as:

$$z_1(k) = C_{z,1} \begin{bmatrix} x_1(t) \\ x_4(t) \end{bmatrix} + D_{z,1} u_1(t) = v_{2,1}(t)$$

where: $C_{z,1} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, D_{z,1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

The LC-DMPC definition of subsystem $\Sigma_2$ is:

$$\begin{bmatrix} \dot{x_2}(t) \\ \dot{x_5}(t) \end{bmatrix} = A_{c_2} \begin{bmatrix} x_2(t) \\ x_5(t) \end{bmatrix} + B_{c_{u,2}} u_2(t) + B_{c_{v,2}} \begin{bmatrix} x_4(t) \\ u_1(t) \\ x_6(t) \\ u_3(t) \end{bmatrix} + B_w w_2$$
$$y_2(t) = C_{c_{y,2}} \begin{bmatrix} x_2(t) \\ x_5(t) \end{bmatrix} \tag{3.4}$$

Once again $A_{c_2}, B_{c_{u,2}}, B_{c_{v,2}}$ and $C_{c_{y,2}}$ are derived from the centralized model and here the regulated output is $x_2(t)$. There are two vectors that work as measured disturbances for $\Sigma_2$ from the upstream neighbors. These vectors are $[x_4(t)\ \ u_1(t)]^T$ from subsystem $\Sigma_1$ and $[x_6(t)\ \ u_3(t)]^T$ from subsystem $\Sigma_3$. Therefore:

$$v_2(t) = \begin{bmatrix} v_{2,1}(t) \\ v_{2,3}(t) \end{bmatrix} = \begin{bmatrix} x_4(t) \\ u_1(t) \\ x_6(t) \\ u_3(t) \end{bmatrix}$$

This subsystem affects its downstream neighbor subsystems $\Sigma_1$ and $\Sigma_3$ throughout state $x_5(t)$ and control action $u_2(t)$, hence, the disturbance output is given by:

$z_2(k) = \left[z_{2,1}(k)\ \ z_{2,3}(k)\right]^T = [x_5(k)\ \ u_2(k)\ \ x_5(k)\ \ u_2(k)]^T$ and the realization can be written as:

$$z_2(t) = \begin{bmatrix} z_{2,1}(t) \\ z_{2,3}(t) \end{bmatrix} = C_{z,2} \begin{bmatrix} x_2(t) \\ x_5(t) \end{bmatrix} + D_{z,2} u_1(t) = \begin{bmatrix} v_1(t) \\ v_3(t) \end{bmatrix}$$

where: $C_{z,2} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, D_{z,1} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$.

Finally, the LC-DMPC subsystem $\Sigma_3$ is defined in a similar way as for subsystem $\Sigma_1$:

$$\begin{bmatrix} \dot{x_3}(t) \\ \dot{x_6}(t) \end{bmatrix} = A_{c_3} \begin{bmatrix} x_3(t) \\ x_6(t) \end{bmatrix} + B_{c_{u,3}} u_3(k) + B_{c_{v,3}} \begin{bmatrix} x_5(t) \\ u_2(t) \end{bmatrix} + B_w w_3$$
$$y_3(t) = C_{c_{y,3}} \begin{bmatrix} x_3(t) \\ x_6(t) \end{bmatrix} \tag{3.5}$$

and as with subsystems $\Sigma_1$ and $\Sigma_2$, $A_{c_3}, B_{c_{u,3}}, B_{c_{v,3}}$ and $C_{c_{y,3}}$ are derived from (3.2). Subsystem $\Sigma_2$ acts as an upstream neighbor for subsystem $\Sigma_3$ through the disturbed vector $[x_5(t)\ \ u_4(t)]^T$ which gives the input disturbance as:

$$v_3(t) = z_{2,3}(t) = \begin{bmatrix} x_5(t) \\ u_2(t) \end{bmatrix}$$

The regulated output of this subsystem is $x_3(t)$. The state $x_6(t)$ and control action $u_3(t)$ disturb subsystem $\Sigma_2$, therefore, the output of $\Sigma_3$ for $\Sigma_2$ is given by $z_3(t) = [x_6(t)\ \ u_2(t)]^T$ which is realized as:

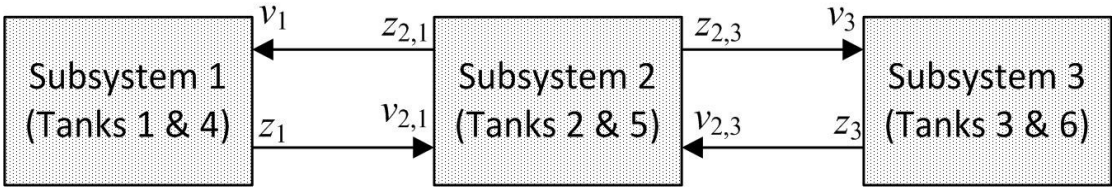
$$z_3(t) = C_{z,3} \begin{bmatrix} x_3(t) \\ x_6(t) \end{bmatrix} + D_{z,3} u_3(t) = v_{2,3}(t)$$

where: $C_{z,3} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, D_{z,3} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

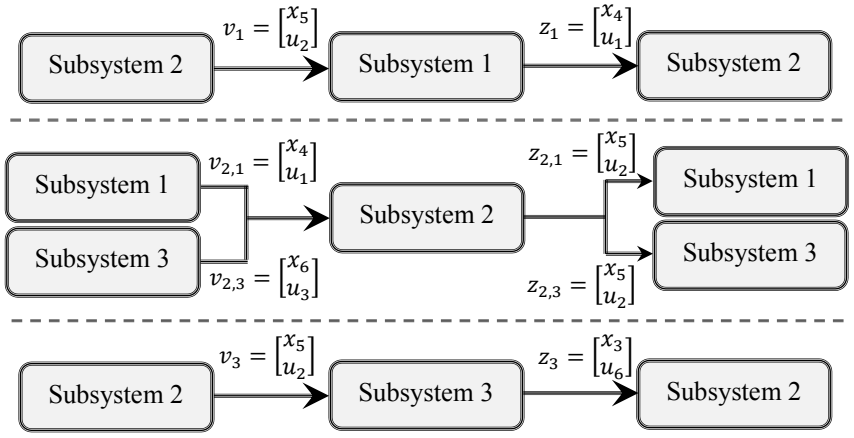
In all equations (3.3) through (3.5), $D_{u,i} = 0$ for $i = 1, 2, 3$ and:

$$B_w = [0 \quad 1]^T$$

In this partition of subsystems, there are direct coupling between subsystems $\Sigma_1$ & $\Sigma_2$, and between $\Sigma_2$ & $\Sigma_3$, but no direct coupling between subsystems 1 and 3 exists. Therefore, according to LC-DMPC architecture, only subsystems $\Sigma_1$ & $\Sigma_2$ and subsystems $\Sigma_2$ & $\Sigma_3$ exchange data. Figure 3.3 illustrates the LC-DMPC communications and upstream and downstream structures for the defined subsystems.




a) Data communications between the subsystems




b) Upstream and downstream subsystems for the six-tank process

Figure 3.3: Data exchange and subsystem structures for the six-tank process

The input and out disturbance vectors and the coupling matrices for the six-tank process subsystems are given in Table 3.2. Now for $N_p = 1$, the network output disturbance could be realized as:

$$\mathbf{Z} = [\mathrm{Z}_1^T \quad \mathrm{Z}_2^T \quad \mathrm{Z}_3^T]^T = \left[\mathrm{Z}_1^T \quad \left(\mathrm{Z}_{2,1}^T \ \mathrm{Z}_{2,3}^T\right) \quad \mathrm{Z}_3^T\right]^T$$

and the network input disturbance is related to $\mathbf{Z}$ through the interconnecting matrix $\mathbf{\Gamma}$ as:

$$\mathbf{V} = \begin{bmatrix} \mathrm{V}_1 \\ \mathrm{V}_2 \\ \mathrm{V}_3 \end{bmatrix} = \begin{bmatrix} \mathrm{V}_1 \\ \begin{bmatrix} \mathrm{V}_{2,1} \\ \mathrm{V}_{2,3} \end{bmatrix} \\ \mathrm{V}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I_{2\times 2} & 0 & 0 \\ I_{2\times 2} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{2\times 2} \\ 0 & 0 & I_{2\times 2} & 0 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} \mathrm{Z}_1 \\ \begin{bmatrix} \mathrm{Z}_{2,1} \\ \mathrm{Z}_{2,3} \end{bmatrix} \\ \mathrm{Z}_3 \end{bmatrix}$$

**Table 3.2: LC-DMPC Matrices and Vectors Defined for the Six-tank Process**

| Subsystem $\Sigma_i$ | Input Disturbance $v_i$ | Output Disturbance $z_i$ | $C_{z,i}$ | $D_{z,i}$ |
|---|---|---|---|---|
| 1, 3 | $v_1 = v_3 = \begin{bmatrix} x_5 \\ u_2 \end{bmatrix}$ | $z_1 = \begin{bmatrix} x_4 \\ u_1 \end{bmatrix}, z_3 = \begin{bmatrix} x_4 \\ u_1 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ |
| 2 | $\begin{bmatrix} x_4 \\ u_1 \\ x_6 \\ u_3 \end{bmatrix}$ | $\begin{bmatrix} x_5 \\ u_2 \\ x_5 \\ u_2 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ |

This six-tank process can also be coupled through the control actions only and the interconnecting matrix **Γ** can be manipulated to handle this type of coupling. Through the modification of **Γ** one can get the desired type of coupling. This concept can be extended for any $n$ number of coupled tanks with any value of $N_p$ (see appendix A).

### 3.1.4 Continuous to Discrete Time Conversion of the Local Subsystem Models

The defined subsystem dynamics (3.3 through 3.5) are continuous, and as a step forward, we need to discretize the continuous local models. For a subsystem dynamic free of process and measurement noises, a practical way to find the local discrete version model is to solve the following equation [84]:

$$\begin{bmatrix} A_i & B_{u,i} & B_{v,i} \\ 0 & I & I \end{bmatrix} = e^{\begin{bmatrix} A_{ci} & B_{cu,i} & B_{cv,i} \\ 0 & 0 & 0 \end{bmatrix} . t_s}$$

where $t_s$ is the sampling time (5 sec is used for simulation in this work).

The final discretized local linear model, with discrete Gaussian distributed process $w_i$ noise and measurement noise $\varepsilon_i$ disturbances, is written as:

$$
\begin{aligned}
x_i(k+1) &= A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_w w_i(k) \\
y_i(k) &= C_{y,i} x_i(k) + \varepsilon_i(k) \\
z_i(k) &= C_{z,i} x_i(k) + D_{z,i} u_i(k)
\end{aligned} \tag{3.6}
$$

## 3.2 Simulation Results

The six-tank process is simulated using algorithms 1 and 2 presented in chapter II. In the first group of simulations, Algorithm 2.1 is only applied assuming full state-feedback and noise free case. The second group of simulations considers the application of both algorithms for output feedback subsystems with disturbance and measurement noises.

### 3.2.1 Application of Algorithm 2.1 with State-Feedback and Noise Free Process

In this subsection, for the six-tank process, the LC-DMPC algorithm is simulated considering two types of coupling: couplings in controls and states, and in controls only. For both cases, full state-feedback is assumed and Laguerre sequences are used to parametrize the local control actions only. The distributed controllers are sharing the full size of vectors and Algorithm 2.1 is compared to a centralized MPC using the same values for weights, actuator saturations, and prediction horizons. Table 3.3 gives the values of the parameters used in simulations. Two different numbers of iterations are used. The local controllers are allowed to communicate for four iterations per sampling in a case while only one iteration is permitted in a different case. Figures 3.4 through 3.9 show the time response, time history of pump control efforts, and changes in cost values with time, respectively. The local controllers can track the centralized solution successfully for the case of four numbers of iterations through solving a reduced size

local optimal control problems. For one iteration per sampling, the controllers can still track the centralized MPC solution. The negative values and peaks in the local costs shown in Figures 3.6 and 3.9 are because of the third term in (2.20) and changes in the tracking local references, respectively. Figure 3.10 illustrates the number of communications per iteration required by the LC-DMPC algorithm versus sharing data with all agents for the case of extending the process to have twenty coupled tanks.

Finally, Figures 3.11 and 3.12 show the eigenvalues of the convergence matrix (2.27a) for both proposed types couplings with different values for the convex combination scalar $\beta$. For both cases, all of the eigenvalues are located inside the unit circle which indicates the stability or convergence of Algorithm 2.1 in chapter II for the coupled tank process. The maximum eigenvalues at the corresponding value of $\beta$ for the both coupling types are given in the table 3.4. As expected, for smaller value of $\beta$ the algorithm converge faster.

**Table 3.3: Parameter Values Used in Simulations for the Six-tank Process**

| Description | Symbol and value |
|---|---|
| Weight on predicted error | $q_1 = 10$, $q_2 = 15$, $q_3 = 14$ |
| Weight on control effort | $s_1 = 0.1$, $s_2 = 1$, $s_3 = 0.8$ |
| Constraints on pump action (deviation from $v = 3$V) | $-2 \le u_1 \le 2$<br>$-1.8 \le u_2 \le 1.8$<br>$-1.5 \le u_3 \le 1.5$ |
| Laguerre Functions Parameters for $u_i$ $j = 1,2,3$ | $N_j = 20, a_j = 0.8$ |
| Prediction and constraint horizons | $N_p = 100$ ($\approx 8.33$ min.)<br>$N_c = 25$ |
| Convex parameter | $\beta = 0.5$ |

**Table 3.4: The Maximum Eigenvalues for the Convergence Matrix (2.27a) with the Six-tank System**

| Coupling type | Maximum eigenvalue |
|---|---|
| **State and control** | |
| $\beta = 0.1$ | 0.0483+0.5793$i$ |
| $\beta = 0.5$ | 0.7741 |
| $\beta = 0.9$ | 0.9525 |
| **Control only** | |
| $\beta = 0.1$ | 0.6075 |
| $\beta = 0.5$ | 0.8362 |
| $\beta = 0.9$ | 0.9716 |

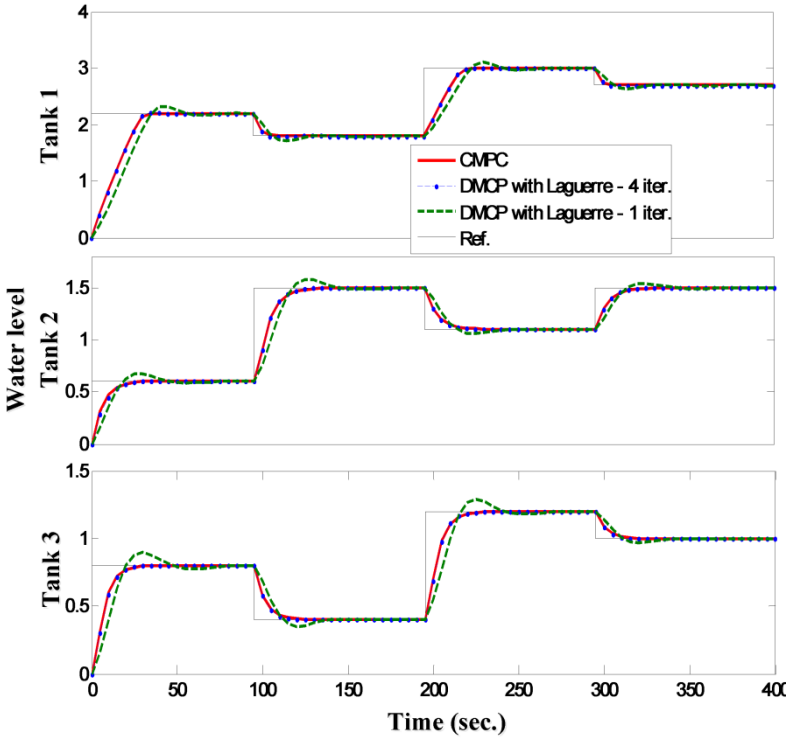



Figure 3.4: Water level responses in lower tanks - state and control coupling (State feedback)

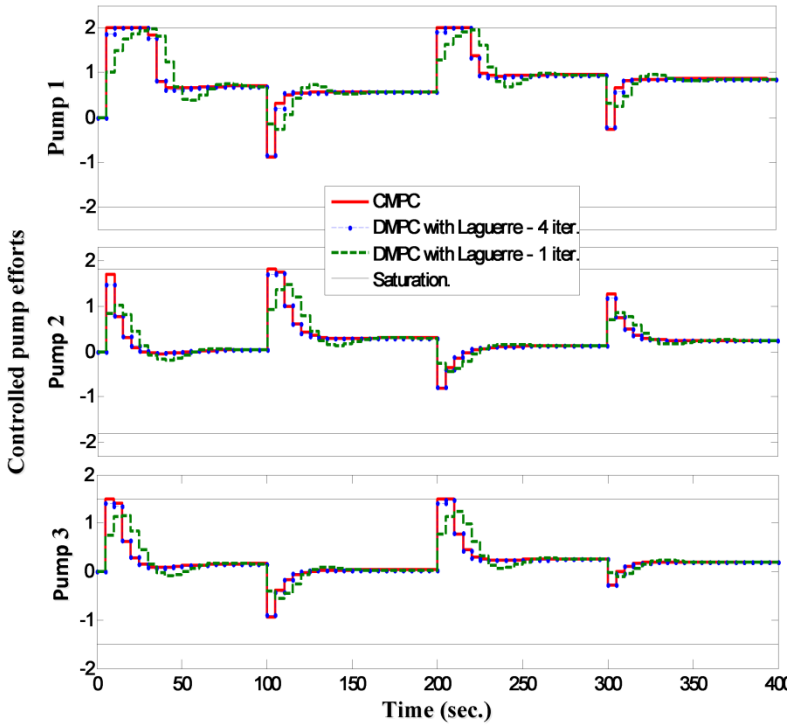



Figure 3.5: Pump control efforts - state and control coupling (State feedback)

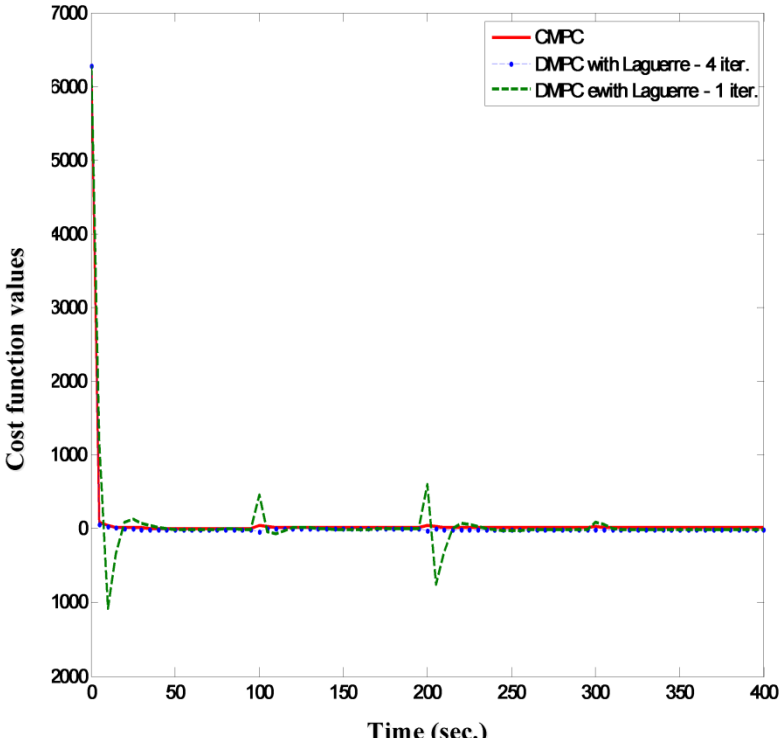



Figure 3.6: Cost function values - state and control coupling (State feedback)

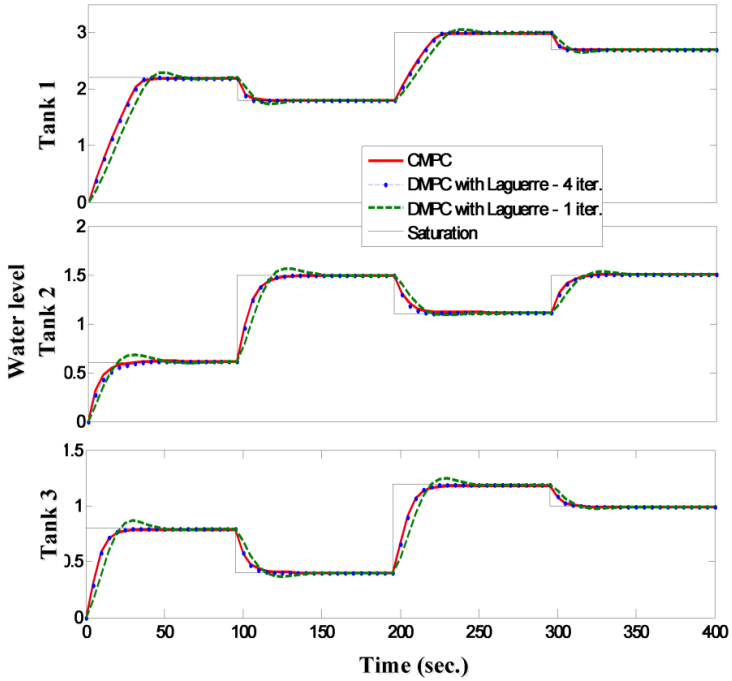



Figure 3.7: Water level responses in lower tanks - coupling in control only (State feedback)

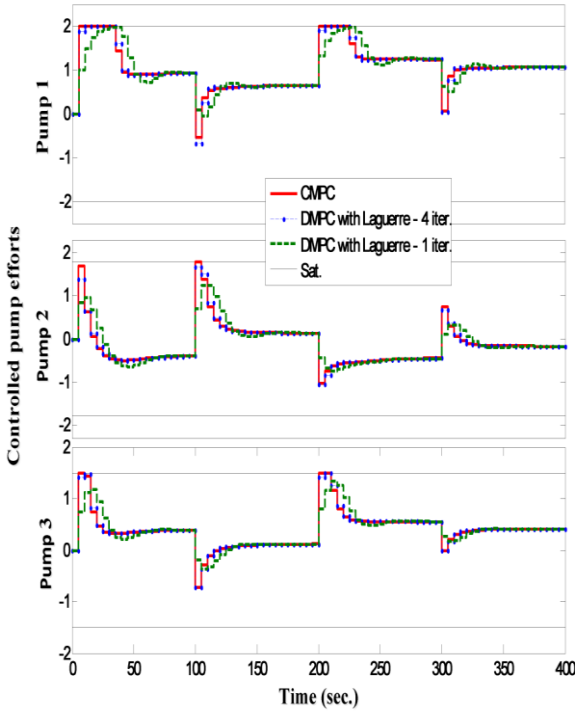



Figure 3.8: Pump control efforts - for coupling in control only (State feedback)

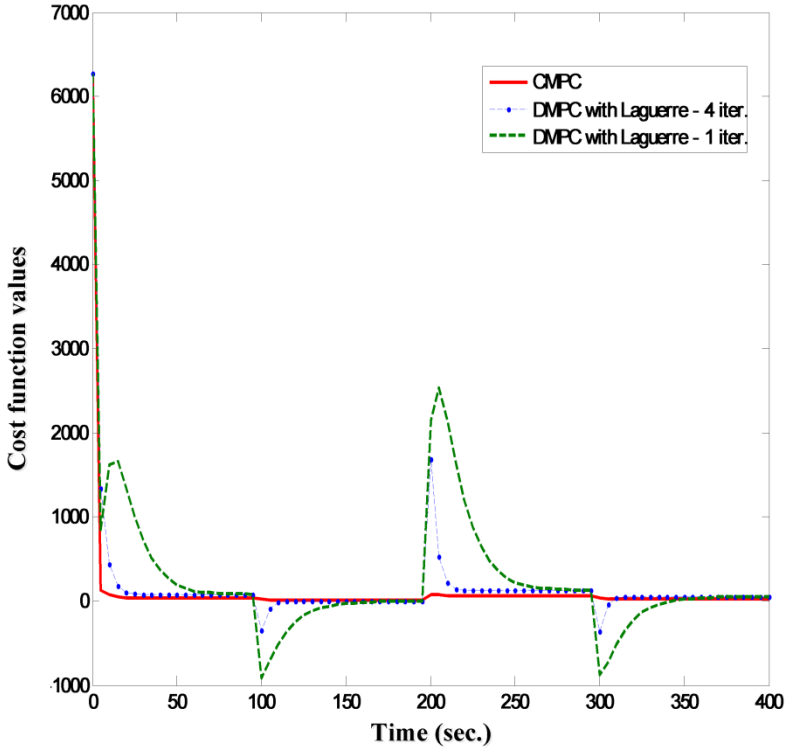



Figure 3.9: Cost function values - coupling in control only (State feedback)

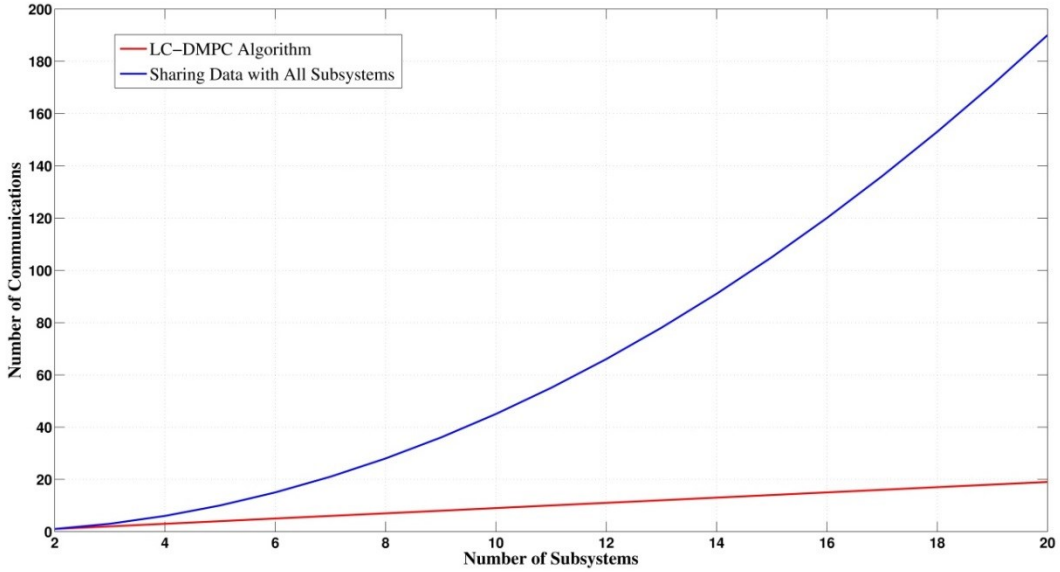



Figure 3.10: Number of communications per iteration for the LC-DMPC approach versus sharing information with all agents

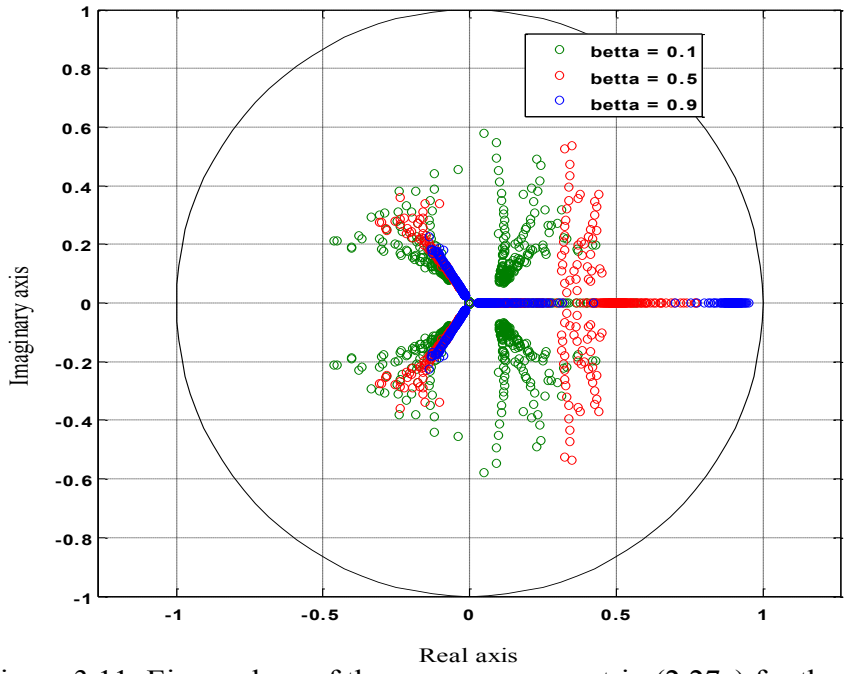



Figure 3.11: Eigenvalues of the convergence matrix (2.27a) for the six-tank process (Coupling in state and control)

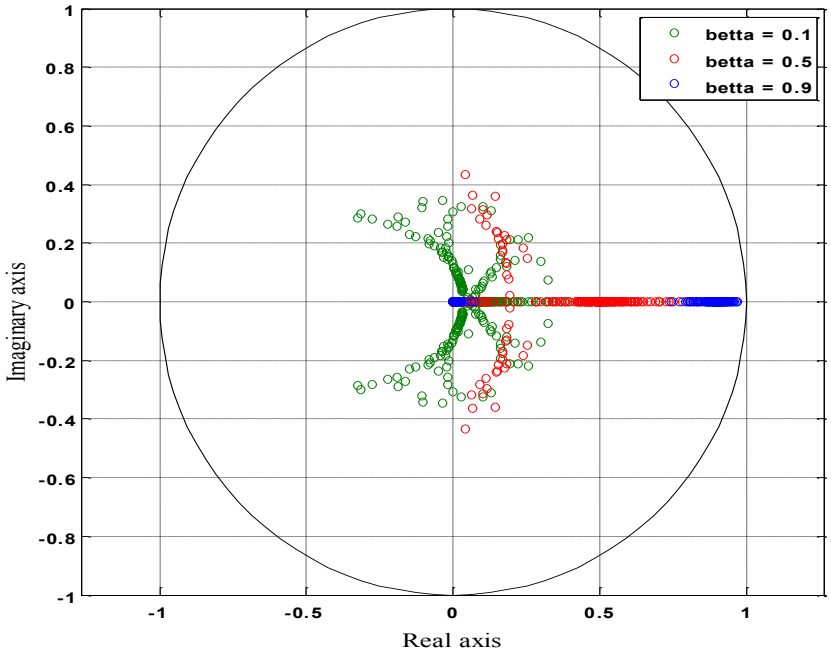



Figure 3.12: Eigenvalues of the convergence matrix (2.27a) for the six-tank process (Coupling in control only)

### 3.2.2 Application of Algorithms 2.1 & 2.2 with Output-Feedback

In this section, the Laguerre functions are used to parametrize the communicated vectors (the output disturbance and cost function sensitivity) along with parameterizing the local control actions. Algorithms 2.1 & 2.2 are used with the feedback of the lower tank states only. Upper tanks are subjected to unmeasured disturbances in addition to measurement noise in the feedback states. Both input noises are uncorrelated and independent identically distributed with known distributions. The disturbed noises are considered to be white with discrete Gaussian distributed and zero mean value. Table 3.5 gives the used covariance. With same parameter values in tables 3.2 and 3.3, the process is simulated considering coupling in states and control actions with four iterations per sampling in a case and one iteration in a different case.

**Table 3.5: Noise Covariance Used for the Six-tank Process**

| Description | Symbol and value |
|---|---|
| Process and measurement covariance | $Q_{w,1} = 0.210,\ Q_{v,1} = 0.130$<br>$Q_{w,2} = 0.164,\ Q_{v,2} = 0.210$<br>$Q_{w,3} = 0.340,\ Q_{v,3} = 0.110$ |

Once again, Figures 3.13 through 3.15 show the time response of the water levels in the lower tanks, the control efforts of the local pumps, and the decrease in the total cost function for both number of iterations, respectively. The local controllers are capable of tracking the centralized solution within the desired set points. The exchanged vectors are assumed to be inaccurate by 8% ($\epsilon = 0.08$) which allows reducing the communicated signal sizes significantly as shown in Figures 3.16 through 3.19.

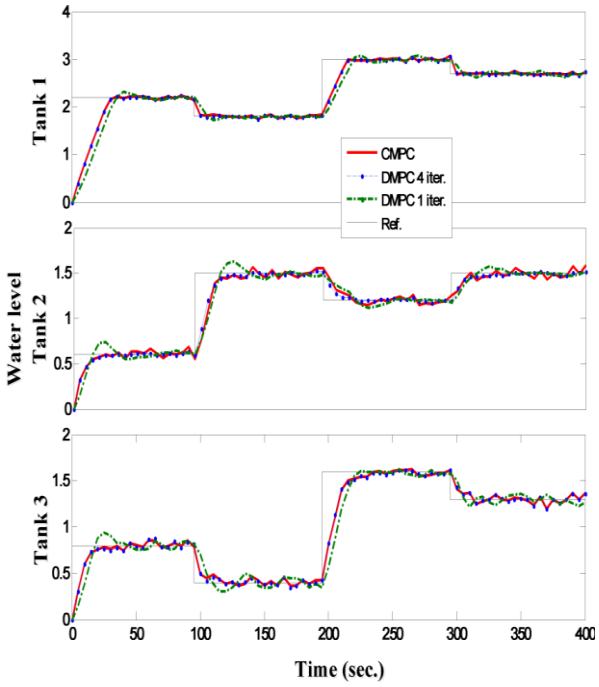



Figure 3.13: Water level responses in lower tanks - coupling in state and control (Output feedback)

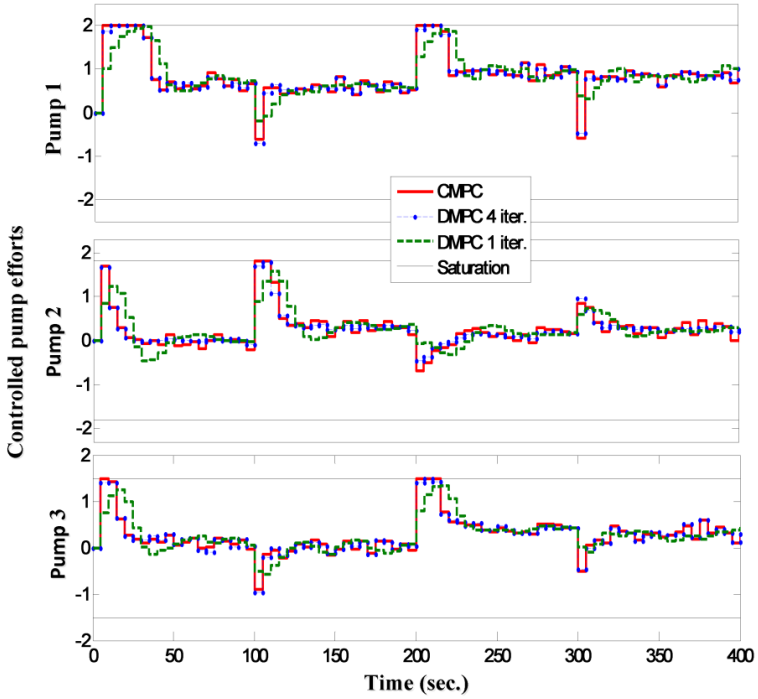



Figure 3.14: Pump control efforts - coupling in state and control (Output feedback)

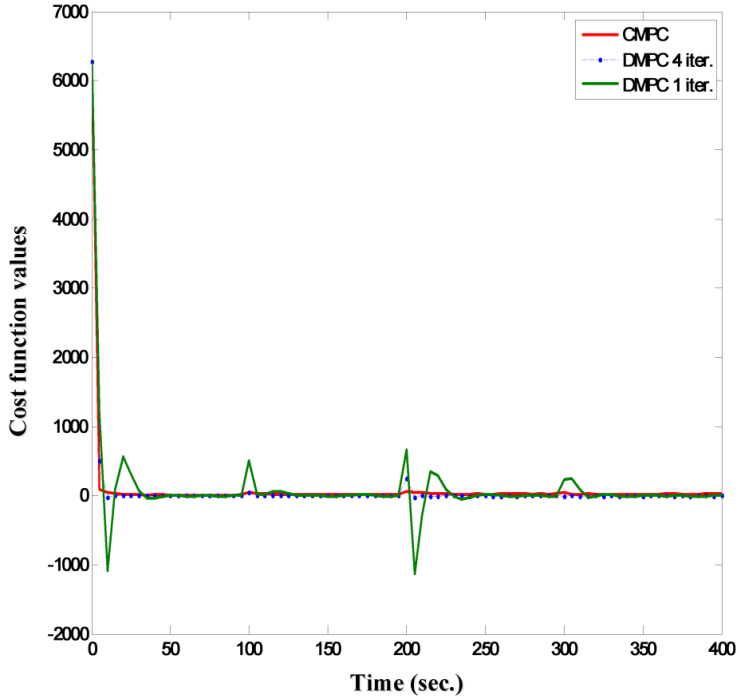



Figure 3.15: Cost function values - coupling in state and control (Output feedback)

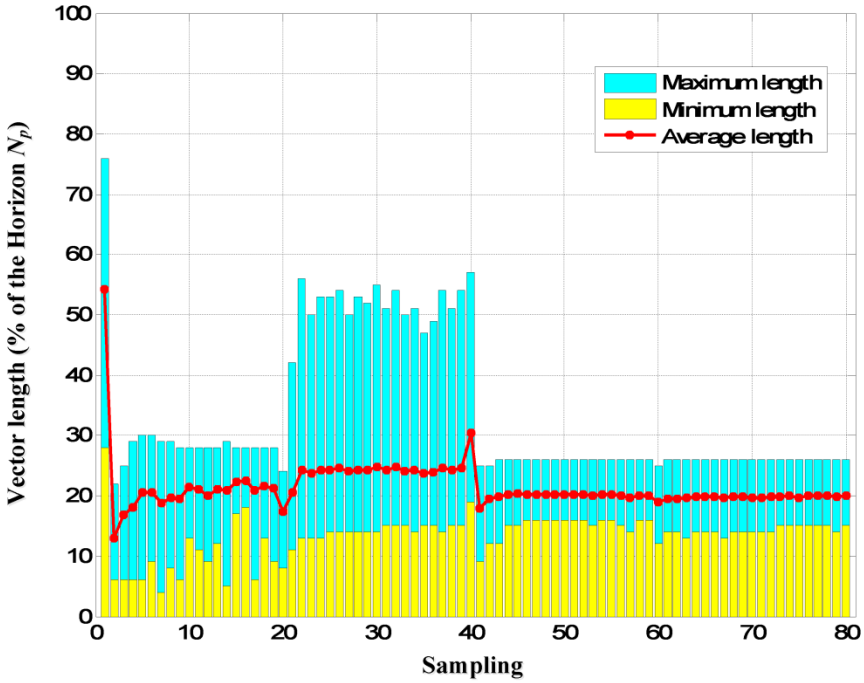



Figure 3.16: Maximum, minimum, and average shared lengths for $Z_i$, $i$ =1, 2, 3, (4 iterations per sampling) - coupling in state and control (Output feedback)

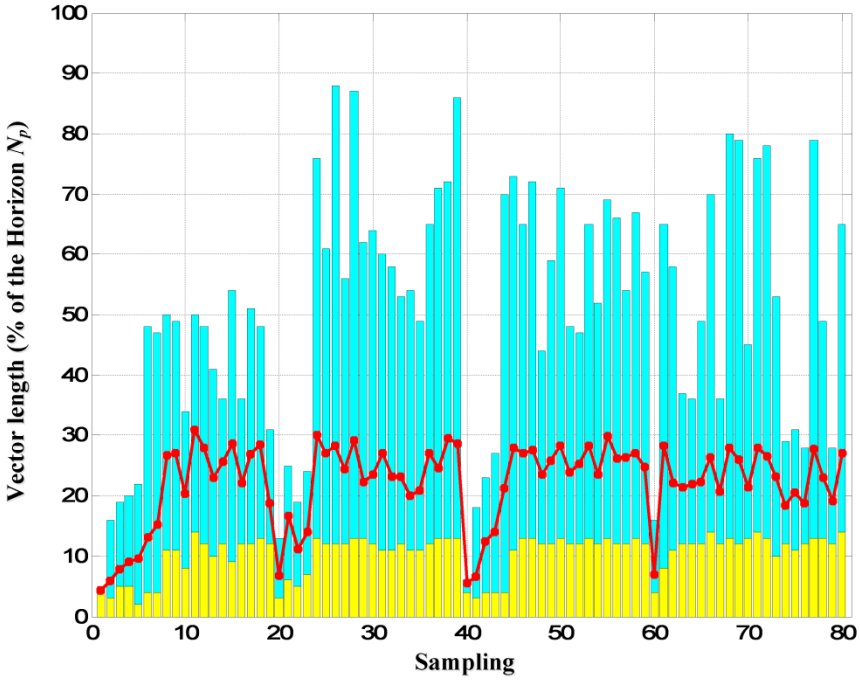



Figure 3.17: Maximum, minimum, and average shared lengths for $\gamma_i$, $i$ =1, 2, 3, (4 iterations per sampling) - coupling in state and control (Output feedback)

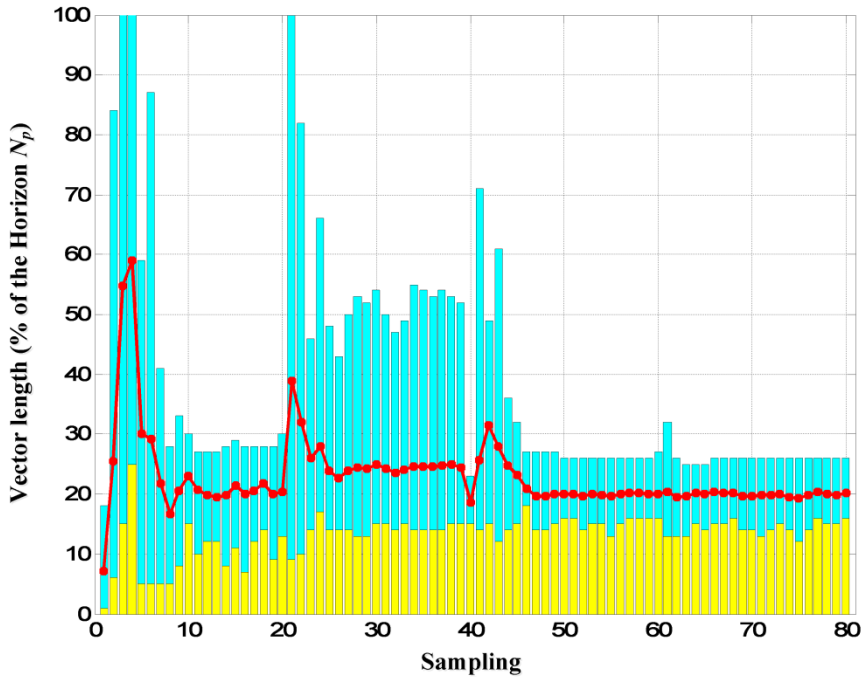



Figure 3.18: Maximum, minimum, and average shared lengths for $Z_i$, $i$ =1, 2, 3, (One iteration per sampling) - coupling in state and control (Output feedback)

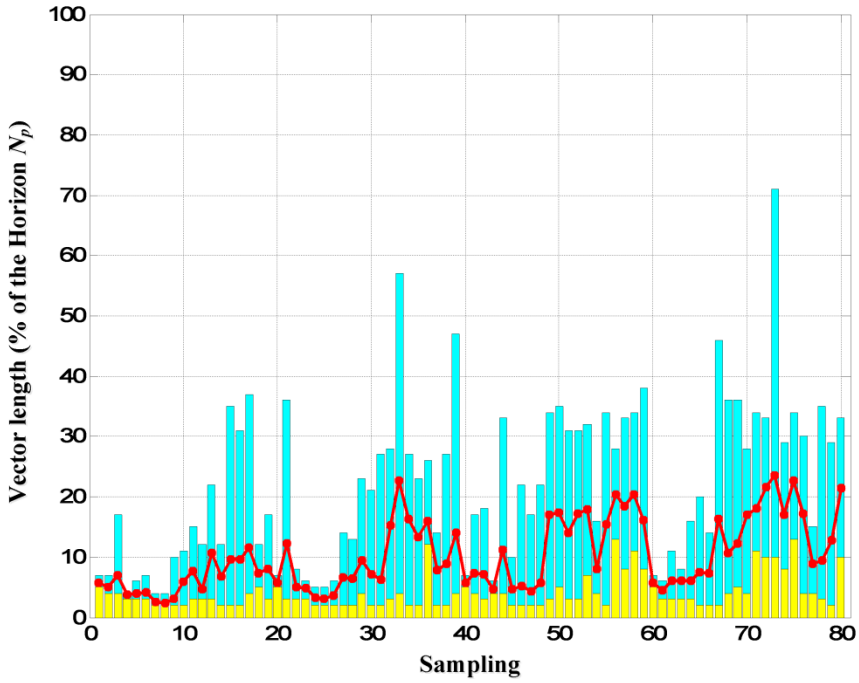



Figure 3.19: Maximum, minimum, and average shared lengths for $\gamma_i$, $i$ =1, 2, 3, (One iteration per sampling) - coupling in state and control (Output feedback)

Since the structure of the local dynamics and cost functions is the same as in the case of the full state-feedback (coupling in state and control), the LC-DMPC with the output feedback will have the same eigenvalues shown in Figure 3.12 with the convergence condition matrix (2.72a).

# 4. THE SUBOPTIMAL LC-DMPC ALGORITHM

The convergence condition for the LC-DMPC algorithm presented in Chapter II (Theorem I) is based on the availability of the all subsystems' dynamic and cost function information in a single matrix. In addition, Theorem II proves the closed-loop stability with the assumption of having sufficient number of iterations per sampling. As a result with the optimal LC-DMPC approach, a centralized supervisor with access to all network information is required and the complexities (or sizes) of the subsystems determine whether if the algorithm can be applied or not. This may restrict the real time application of the approach as subsystem local information change frequently which requires resolving the convergence condition in (2.27); moreover the solution of the convergence matrix becomes more expensive computationally as the subsystems in the network increases.

In this chapter, the suboptimal LC-DMPC algorithm is introduced. This approach eliminates the need for a centralized monitor by distributing the convergence condition through the subsystems which also reduces the computational loads significantly. Each subsystem is required to show dissipativity in the local information dynamics with a local gain less than one. This ensures the convergence of the algorithm without the need of solving a systemwide problem. However, this is a conservative problem. As an alternative method, a smaller systemwide problem based only on the finite gains of the local subsystems and network topology is proposed. This method gives a condition for the network information dynamic dissipativity that guarantees convergence of the

algorithm. For each subsystem to be dissipative with a finite gain comes on the price of the suboptimality of the distributed problems. This also affects the closed-loop stability as Theorem II depends on the assumption that the local control actions can approach the centralized solution per sampling. To solve the local subsystem closed-loop stability with the suboptimal LC-DMPC algorithm, a distributed synthesis of a local stabilizing terminal cost is introduced.

This chapter is outlined as follows: Section 4.1 introduces a network of coupled subsystems as an example to demonstrate the applications of the proposed concepts. In Section 4.2, the core idea of the suboptimal LC-DMPC algorithm is given where the suboptimal cost function is studied with new introduced local variables. Dissipativity definitions for the local subsystems and network dissipativity conditions are presented in Section 4.3. Section 4.4 illustrates the local closed-loop stability terminal cost synthesis while Section 4.5 details the steps of the suboptimal LC-DMPC algorithm.

## 4.1 A Simple Interconnected Subsystems Example

Throughout this chapter, two theorems and many definitions will be stated and in order to demonstrate the applications of these introduced statements, a small network example is used. Figure 4.1 shows the interconnected subsystem example. There are three subsystems with different dimensions, inputs, outputs, and couplings (states, control actions or both).

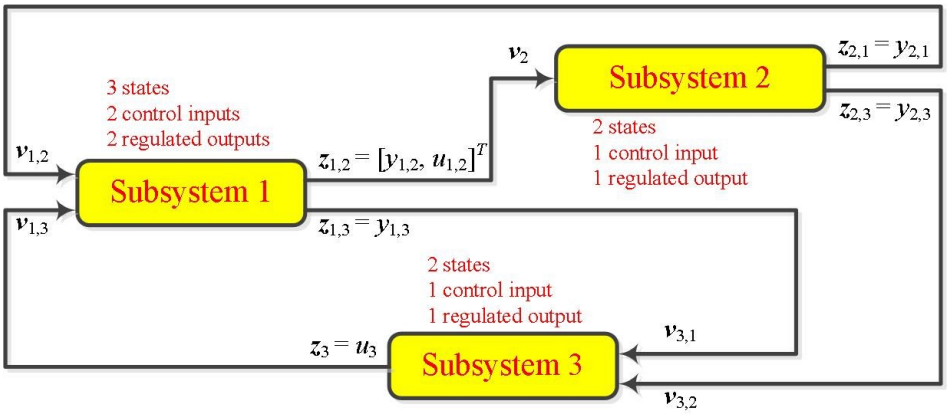



Figure 4.1: Three interconnected subsystems example

Subsystem 1 has two downstream outputs: The output $y_{1,2}$ and control action $u_{1,2}$ which affect subsystem 2 through $z_{1,2}$, and $z_{1,2} = y_{1,3}$ which disturbs subsystem 3. Subsystem 2 also has two downstream outputs: $z_{2,1} = y_{2,1}$ for subsystem 1, and $z_{2,3} = y_{2,3}$ for subsystem3. Finally, subsystem 3 disturbs subsystem 1 only through its control action $z_3 = u_3$. With this coupling and for prediction horizon equal to one ($N_p = 1$), we can write the following network relationship (Appendix B gives more details):

$$\begin{bmatrix} v_1 \begin{Bmatrix} v_{1,2} \\ v_{1,3} \end{Bmatrix} \\ v_2 \\ v_3 \begin{Bmatrix} v_{3,1} \\ v_{3,2} \end{Bmatrix} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\Gamma_{N_p=1}} \begin{bmatrix} z_1 \begin{Bmatrix} z_{1,2} \\ z_{1,3} \end{Bmatrix} \\ z_2 \begin{Bmatrix} z_{2,1} \\ z_{2,3} \end{Bmatrix} \\ z_3 \end{bmatrix}$$

The local dynamics and cost function matrices are given in appendix B. For $N_p = 20$ and $\beta = 0.1$, the eigenvalues of the convergence matrix (2.27a) are plotted in Figure 4.2. As can be seen, there are some eigenvalues outside the unit circle which violate the

convergence condition given by Theorem II in chapter II (even for $\beta = 0$), therefore, this network is not LC-DMPC stable.

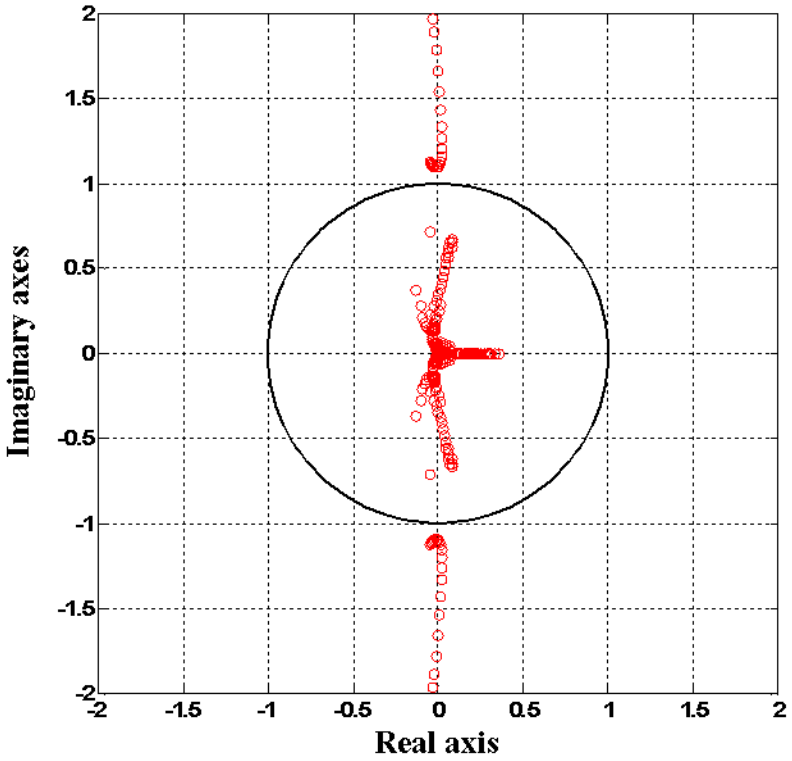



Figure 4.2: Eigenvalues of the convergence matrix (2.27a) for the interconnected network example

## 4.2 The Suboptimal Approach Core Idea

In order for us to drive a local convergence condition, we have used the definition of a dissipative system. At the local subsystem, the dynamic of the information exchange can be formulated as input-output system along the horizon as shown in Figure 4.3. The idea is to find some local variables that can make this input-output system be dissipative with respect to the iteration domain. Then, based on the dissipativity of each local agent, we can ensure the dissipativity of the information dynamics at the network level which

consequently ensures the convergence of the LC-DMPC algorithm. With the intention of having local free design variables, we require the LC-DMPC algorithm to operate at some suboptimal level. This gives us the ability to add two more design variables to the local information exchange dynamics. Next section details the addition of such variables.

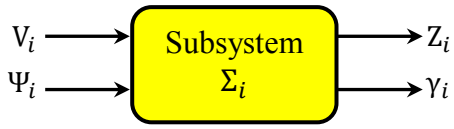



Figure 4.3: Input-output information dynamic system for an LC-DMPC local controller

### 4.2.1 Suboptimal Local Cost and Sensitivity Functions

To converge to the systemwide optimum, we require that the local MPC solves a modified cost function with three terms. The first two terms is same as the centralized cost but defined locally and the third term represents the penalty on the subsystem output for downstream systems. In order to have more freedom for designing the local convergence conditions, we relaxed the third term in the local cost with a scalar. The new local cost function is given by:

$$J_i = \sum_{m=1}^{N_p} \|e_i(k+m)\|_{q_i}^2 + \sum_{m=0}^{N_p-1} \|u_i(k+m)\|_{s_i}^2 + \alpha_i \sum_{m=0}^{N_p-1} \psi_i(k+m)^T z_i(k+m)$$

or

$$J_i = \mathrm{e}_i^T \mathrm{Q}_i \mathrm{e}_i + \mathrm{U}_i^T \mathrm{S}_i \mathrm{U}_i + \alpha_i \Psi_i^T \mathrm{Z}_i$$

where $\alpha_i \in \Re: 0 \leq \alpha_i \leq 1$

This means that now the local MPC will not fully consider its output effects for downstream systems in the cost function. The scalar $\alpha_i$ determines how much the local

agent should count for its disturbance for downstream neighbors before it is not dissipative. This, of course, will affect the optimality of the LC-DMPC algorithm as it will be at some level of suboptimality with respect to the centralized solution.

The second local free variable is added to the sensitivity equation. In chapter II, the following local sensitivity equation was used in the LC-DMPC algorithm (with $d_i = 0$):

$$\gamma_i = 2N_{y,i}^T \mathrm{Q}_i N_{y,i} \mathrm{V}_i + 2N_{y,i}^T \mathrm{Q}_i M_{y,i} \mathrm{U}_i + 2N_{y,i}^T \mathrm{Q}_i F_{y,i} x_{0,i} + N_{z,i}^T \alpha_i \Psi_i - 2N_{y,i}^T \mathrm{Q}_i \mathrm{r}_i$$

This local sensitivity equation is also relaxed with the variable $\mu_i$ as shown below:

$$\gamma_i = \mu_i \left\{ 2N_{y,i}^T \mathrm{Q}_i N_{y,i} \mathrm{V}_i + 2N_{y,i}^T \mathrm{Q}_i M_{y,i} \mathrm{U}_i + 2N_{y,i}^T \mathrm{Q}_i F_{y,i} x_{0,i} + N_{z,i}^T \alpha_i \Psi_i - 2N_{y,i}^T \mathrm{Q}_i \mathrm{r}_i \right\}$$

where $\mu_i \in \Re: 0 \leq \mu_i \leq 1$.

At this point, it is important to state the following:

1. In the optimal case where no $\alpha_i$ or $\mu_i$ is used, there are three free variables that can be manipulated in order to have a stable LC-DMPC algorithm. The local weighting matrices $q_i$ & $s_i$ and the convex combination parameter $\beta$. However, in some real applications, the cost matrices cannot be changed as they are related to operation costs or safety requirements, therefore the only available free design variable is $\beta$.
2. In the suboptimal mode, we have added two more free variables $\alpha_i$ and $\mu_i$ in addition, the convex combination parameter $\beta$ may also be treated as a local parameter, i.e. $\beta_i$.
3. The new suboptimal free variables $\alpha_i$ and $\mu_i$ determine the level of suboptimality as will be shown in the next section.
4. The suboptimal level of the LC-DMPC algorithm is not with respect to a decentralized solution. In a decentralized solution, the local controllers are not

considering the couplings within neighbors. However, even with $\alpha_i = \mu_i = 0$, for $i = 1, \dots, p$, the local agents with the LC-DMPC algorithm are still counting for the measured input disturbances $v_i$ from upstream neighbor subsystems.

### 4.2.2 Optimality

As it is stated above, the introduced free variables $\alpha_i$ and $\mu_i$ will drive the LC-DMPC algorithm to work in a suboptimal level. In this section, we will show how the centralized cost function changes with the variables $\alpha_i$ and $\mu_i$ at the network level.

The local suboptimal unconstrained MPC control action (with $d_i = 0$) is now given by:

$$\left(\mathrm{S}_i + M_{y,i}^T \mathrm{Q}_i M_{y,i}\right)\mathrm{U}_i^{QP} = \left[M_{y,i}^T \mathrm{Q}_i \mathrm{r}_i - M_{y,i}^T \mathrm{Q}_i F_{y,i} x_{0,i} - 0.5\alpha_i M_{z,i}^T \Psi_i - M_{y,i}^T \mathrm{Q}_i N_{y,i} \mathrm{V}_i\right]$$

where $\Psi_i = f(\mu_{i-1})$.

Following the same steps in chapter II, we can write the unconstrained control actions as function of the local defined free variables at the network level as:

$$\left[\mathbf{S} + \left(\mathbf{M_y} + \left(\boldsymbol{\alpha}\mathbf{M_z^T}[I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N_z^T}]^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N_y^T}\right)^T\right)^T \mathbf{Q}\left(\mathbf{M_y} + \mathbf{N_y}\mathbf{W}\mathbf{M_z}\right)\right]\mathbf{U^{QP}} =$$

$$\left[\mathbf{M_y} + \left(\boldsymbol{\alpha}\mathbf{M_z^T}[I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N_z^T}]^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N_y^T}\right)^T\right]^T \mathbf{Q}\mathbf{r} -$$

$$\left[\left(\mathbf{M_y} + \left(\boldsymbol{\alpha}\mathbf{M_z^T}[I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N_z^T}]^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N_y^T}\right)^T\right)^T \mathbf{Q}\left(\mathbf{F_y} + \mathbf{N_y}\mathbf{W}\mathbf{F_z}\right)\right]\mathbf{X_0} \qquad (4.1)$$

where

$\boldsymbol{\alpha} = diag(\alpha_1, \cdots, \alpha_p)$, $\boldsymbol{\mu} = diag(\mu_1, \cdots, \mu_p)$, and $\mathbf{W} = [I - \boldsymbol{\Gamma}\mathbf{N_z}]^{-1}\boldsymbol{\Gamma}$.

Recall that the optimum network centralized control action is:

$$\left[\mathbf{S}+\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)^{T}\mathbf{Q}\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]\mathbf{U}^{\mathbf{QP}}=\left[\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right]^{T}\mathbf{Q}\mathbf{r}-$$

$$\left[\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)^{T}\mathbf{Q}\left(\mathbf{F}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{F}_{\mathbf{z}}\right)\right]\mathbf{X}_{\mathbf{0}} \tag{4.2}$$

In (4.1), if $\boldsymbol{\alpha}=0$ & $\boldsymbol{\mu}=0$, the suboptimal control action would be:

$$\left[\mathbf{S}+\mathbf{M}_{\mathbf{y}}^{T}\mathbf{Q}\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]\mathbf{U}^{\boldsymbol{QP}}=\mathbf{M}_{\mathbf{y}}^{T}\mathbf{Q}\mathbf{r}-\left[\mathbf{M}_{\mathbf{y}}^{T}\mathbf{Q}\left(\mathbf{F}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{F}_{\mathrm{z}}\right)\right]\mathbf{X}_{\mathbf{0}} \tag{4.3}$$

and as $\boldsymbol{\alpha}\rightarrow I$ & $\boldsymbol{\mu}\rightarrow I$, equation (4.1) convergences to the optimal (4.2) solution.

As another way to show that (4.1) approaches (4.2) as $\boldsymbol{\alpha}\rightarrow I$ is by finding the derivative of the centralized cost function with respect to $\boldsymbol{\alpha}$ when $\boldsymbol{\mu}=I$ and prove that the derivative is equal to zero when $\boldsymbol{\alpha}=I$. This is illustrated below.

Recall that the centralized cost function is:

$$J_{\mathrm{CMPC}}=\ \mathbf{U}^{T}\left[\mathbf{S}+\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)^{T}\mathbf{Q}\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]\mathbf{U}+$$

$$2\left[\mathbf{X}_{\mathbf{0}}^{T}\left(\mathbf{F}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{F}_{\mathrm{z}}\right)^{T}\mathbf{Q}\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathrm{z}}\right)-\mathbf{r}^{T}\mathbf{Q}\left(\mathbf{M}_{y}+\mathbf{N}_{y}\mathbf{W}\mathbf{M}_{\mathrm{z}}\right)\right]\mathbf{U}+C_{0} \tag{4.4}$$

where

$$C_{0}=\mathbf{X}_{0}^{T}\left(\mathbf{F}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{F}_{\boldsymbol{z}}\right)^{T}\mathbf{Q}\left(\mathbf{F}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{F}_{\boldsymbol{z}}\right)\mathbf{X}_{0}-2\mathbf{X}_{0}^{T}\left(\mathbf{F}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{F}_{\boldsymbol{z}}\right)\mathbf{Q}\mathbf{r}+\mathbf{r}^{T}\mathbf{Q}\mathbf{r}$$

By substituting (4.1) in (4.4) for $\mathbf{U}$ and differentiating the result with respect to the variable $\boldsymbol{\alpha}$ (assuming $\boldsymbol{\mu}=I$), we will have the following (see appendix III for details):

$$\frac{\partial J_{\mathrm{CMPC}}}{\partial\boldsymbol{\alpha}}=2\left(\left[X_{\boldsymbol{\mu\alpha}}G_{\boldsymbol{\mu\alpha}}\right]^{T}H_{1}+H_{2}\right)\left[X_{\boldsymbol{\mu\alpha}}\frac{\partial G_{\boldsymbol{\mu\alpha}}}{\partial\boldsymbol{\alpha}}+\left(G_{\boldsymbol{\mu\alpha}}^{T}\frac{\partial X_{\boldsymbol{\mu\alpha}}}{\partial\boldsymbol{\alpha}}\right)^{T}\right] \tag{4.5}$$

where

$$\frac{\partial G_{\boldsymbol{\mu\alpha}}}{\partial\boldsymbol{\alpha}}=\left[\left(\left(\boldsymbol{\alpha}\mathbf{M}_{\boldsymbol{z}}^{T}[[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\mu\alpha}\mathbf{N}_{\boldsymbol{z}}^{\boldsymbol{T}}]^{-\mathbf{1}}[\boldsymbol{\Gamma}^{T}\boldsymbol{\mu}\mathbf{N}_{\boldsymbol{z}}^{T}][I-\boldsymbol{\Gamma}^{T}\boldsymbol{\mu\alpha}\mathbf{N}_{\boldsymbol{z}}^{T}]^{-\mathbf{1}}]\right.\right.$$

$$\left.\left.+[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\mu\alpha}\mathbf{N}_{\boldsymbol{z}}^{T}]^{-\mathbf{1}}\mathbf{M}_{\boldsymbol{z}}^{T})\boldsymbol{\Gamma}^{T}\boldsymbol{\mu}\mathbf{N}_{\mathbf{y}}^{T}\right)^{T}\right]^{T}\left[\mathbf{Q}\mathbf{r}-\mathbf{Q}\left(\mathbf{F}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{F}_{\mathbf{z}}\right)\mathbf{X}_{\mathbf{0}}\right]$$

$$\frac{\partial \mathrm{X}_{\boldsymbol{\mu\alpha}}}{\partial \boldsymbol{\alpha}} = -\left[\mathbf{S} + G_1^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right]^{-\mathbf{1}} \cdot$$

$$\left[\left((\boldsymbol{\alpha}\mathbf{M}_\mathbf{z}^T([I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N}_\mathbf{z}^T]^{-\mathbf{1}}\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N}_\mathbf{z}^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N}_\mathbf{z}^T]^{-\mathbf{1}}) + [I - \boldsymbol{\Gamma}^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N}_\mathbf{z}^T]^{-\mathbf{1}}\mathbf{M}_\mathbf{z}^T)\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N}_\mathbf{y}^T\right)^T\right]^T$$

$$\cdot \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\left[\mathbf{S} + \boldsymbol{G}_\mathbf{1}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right]^{-\mathbf{1}}$$

$$G_1 = \mathbf{M}_y + \left(\boldsymbol{\alpha}\mathrm{M}_z^T[I - \Gamma^T\boldsymbol{\mu}\boldsymbol{\alpha}\mathbf{N}_z^T]^{-1}\Gamma^T\boldsymbol{\mu}\mathrm{N}_\mathrm{y}^T\right)^T$$

$$G_{\boldsymbol{\mu\alpha}} = G_1^T\left[\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathrm{z}\right)\mathbf{X_0}\right]$$

$$\mathrm{X}_{\boldsymbol{\mu\alpha}} = \left[\mathbf{S} + G_1^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right]^{-1}$$

$$H_1 = \mathbf{S} + \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)$$

$$H_2 = \mathbf{X_0^T}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathrm{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathrm{z}\right) - \mathbf{r}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathrm{z}\right)$$

In order to show that the derivative of the centralized cost with respect to $\boldsymbol{\alpha}$ is zero when $\boldsymbol{\alpha} = I$, we can take the first term of the LHS of (4.5) and extend it as following:

$$\left[X_{\boldsymbol{\mu\alpha}} G_{\boldsymbol{\mu\alpha}}\right]^T H_1 + H_2 =$$

$$\left(G_1^T\left(\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right)\mathbf{X_0}\right)\right)^T \left(\left(\mathbf{S} + G_1^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right)^{-1}\right)^T \cdot$$

$$\left(\mathbf{S} + \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right) +$$

$$\mathbf{X_0^T}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right) - \mathbf{r}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)$$

Note that for $\boldsymbol{\mu}$ and $\boldsymbol{\alpha} = I$:

$$G_1 = \mathrm{M}_y + \left(\alpha\mathrm{M}_z^T[\mathrm{I} - \Gamma^T\mu\alpha\mathrm{N}_z^T]^{-1}\Gamma^T\mu\mathrm{N}_\mathrm{y}^T\right)^T = \mathrm{M}_y + \left(\mathrm{M}_z^T[I - \Gamma^T\mathrm{N}_z^T]^{-1}\Gamma^T\mathrm{N}_\mathrm{y}^T\right)^T$$

And we have the definition of $\mathbf{W}$ as:

$$\mathbf{W} = [I - \boldsymbol{\Gamma}\mathbf{N}_\mathbf{z}]^{-1}\boldsymbol{\Gamma}$$

which, with a matrix identity, can also be written as:

$$\mathbf{W}^T = [I - \mathbf{\Gamma}^T \mathbf{N}_\mathbf{z}^T]^{-1} \mathbf{\Gamma}^T$$

Then we can write:

$$G_1 = \mathrm{M}_y + \left(\mathrm{M}_z^T W^T \mathrm{N}_\mathrm{y}^T\right)^T = \mathrm{M}_y + \mathrm{N}_\mathrm{y} W \mathrm{M}_\mathrm{z}$$

This makes the following term equal to identity:

$$\left(\left(\mathbf{S} + G_1^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right)^{-1}\right)^T \left(\mathbf{S} + \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right) =$$

$$\left(\mathbf{S} + \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right)^{-T} \left(\mathbf{S}\right.$$

$$\left. + \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)\right) = I$$

which with $G_1 = \mathbf{M}_y + \mathbf{N}_\mathbf{y} \mathbf{W} \mathbf{M}_\mathbf{z}$ gives:

$$\left(\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \left[\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right) \mathbf{X}_\mathbf{0}\right]\right)^T +$$

$$\mathbf{X}_\mathbf{0}^\boldsymbol{T}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right) - \mathbf{r}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)$$

where the first term on LHS can be written as:

$$\left(\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \left(\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right) \mathbf{X}_\mathbf{0}\right)\right)^T =$$

$$\left(\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\mathbf{r} - \left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right) \mathbf{X}_\mathbf{0}\right)^T$$

or

$$\mathbf{r}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right) - \mathbf{X}_\mathbf{0}^\boldsymbol{T}\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right)$$

Therefore by summing both sides:

$$\mathbf{r}^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right) - \mathbf{X}_\mathbf{0}^T\left(\mathbf{F}_y + \mathbf{N}_y \mathbf{W} \mathbf{F}_\mathbf{z}\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y \mathbf{W} \mathbf{M}_\mathbf{z}\right) +$$

$$\mathbf{X_0^T}\left(\mathbf{F}_y + \mathbf{N}_y\mathbf{W}\mathbf{F_z}\right)^T\mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M_z}\right) - \mathbf{r}^T\mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M_z}\right) = 0$$

and consequently:

$\frac{\partial J_{\mathrm{CMPC}}}{\partial \alpha} = 0$, for $\boldsymbol{\mu}$ and $\boldsymbol{\alpha} = I$.

### 4.3 Convergence and Local Dissipativity

In this section the definition of a dissipative system as well as conditions for a system to be dissipative are given. The section begins by realizing the local dynamics of information sharing in a state-space representation. Then dissipativity conditions for local and network information dynamics are presented. For network dissipativity, three different methods are stated based on the dissipativity inequality. Finally, the interconnected subsystems example is used to show the applications of the presented methods.

#### 4.3.1 Local Information Dynamics and Dissipativity

Before stating the convergence theorem of the suboptimal LC-DMPC algorithm, the dynamics of information exchange at the level of a subsystem is first represented as state-space realization.

According to the LC-DMPC algorithm, the local information sharing dynamics (with $d_i = 0$) at each subsystem is given by the following difference equations where $j$ is the iteration index:

$\mathrm{U}_i(j+1) = \beta_i \mathrm{U}_i(j) + (1-\beta_i)\mathrm{U}_i{}^{QP}(j)$

$$\mathrm{Z}_i(j+1) = F_{z,i}x_{0,i}(k) + M_{z,i}\beta_i\mathrm{U}_i(j) + M_{z,i}(1-\beta_i)\mathrm{U}_i{}^{QP}(j) + N_{z,i}\mathrm{V}_i(j)$$

$$\gamma_i(j+1) = 2\mu_i N_{y,i}^T \mathrm{Q}_i\{F_{y,i}x_{0,i}(k) + M_{y,i}\beta_i\mathrm{U}_i(j) + M_{y,i}(1-\beta_i)\mathrm{U}_i{}^{QP}(j) + N_{y,i}\mathrm{V}_i(j) - \mathrm{r}_i(k)\} + \mu_i N_{z,i}^T \alpha_i \Psi_i(j)$$

and by introducing the local suboptimal unconstrained control action:

$$\mathrm{U}_i^{QP}(j) = \mathrm{X}_i\left[M_{y,i}^T\mathrm{Q}_i\mathrm{r}_i(k) - M_{y,i}^T\mathrm{Q}_iF_{y,i}x_{0,i}(k) - M_{y,i}^T\mathrm{Q}_iN_{y,i}\mathrm{V}_i(j) - 0.5\alpha_iM_{z,i}^T\Psi_i(j)\right]$$

where $\mathrm{X}_i = \left(\mathrm{S}_i + M_{y,i}^T\mathrm{Q}_iM_{y,i}\right)$ into these difference equations, the resulted difference equations would a function of $\mathrm{U}_i(j), \mathrm{V}_i(j)$, and $\Psi_i(j)$ that change with iteration $j$ and $\mathrm{r}_i(k)$ and $x_{0,i}(k)$ that change with the sampling $k$ as following:

$$\mathrm{U}_i(j+1) = \beta_i\mathrm{U}_i(j) + (1-\beta_i)\mathrm{X}_i\left[M_{y,i}^T\mathrm{Q}_i\mathrm{r}_i(k) - M_{y,i}^T\mathrm{Q}_iF_{y,i}x_{0,i}(k) - M_{y,i}^T\mathrm{Q}_iN_{y,i}\mathrm{V}_i(j) - 0.5\alpha_iM_{z,i}^T\Psi_i(j)\right]$$

$$\mathrm{Z}_i(j+1) = F_{z,i}x_i(k) + M_{z,i}\beta_i\mathrm{U}_i(j) + M_{z,i}(1-\beta_i)\mathrm{X}_i\left[M_{y,i}^T\mathrm{Q}_i\mathrm{r}_i(k) - M_{y,i}^T\mathrm{Q}_iF_{y,i}x_{0,i}(k) - M_{y,i}^T\mathrm{Q}_iN_{y,i}\mathrm{V}_i(j) - 0.5\alpha_iM_{z,i}^T\Psi_i(j)\right] + N_{z,i}\mathrm{V}_i(j)$$

$$\gamma_i(j+1) = 2\mu_iN_{y,i}^TQ_iN_{y,i}\mathrm{V}_i(j) + 2\mu_iN_{y,i}^TQ_iM_{y,i}\beta_i\mathrm{U}_i(j) + 2\mu_iN_{y,i}^TQ_iM_{y,i}(1-\beta_i)\mathrm{X}_i\left[M_{y,i}^TQ_i\mathrm{r}_i(k) - M_{y,i}^T\mathrm{Q}_iF_{y,i}x_{0,i}(k) - M_{y,i}^T\mathrm{Q}_iN_{y,i}\mathrm{V}_i(j) - 0.5\alpha_iM_{z,i}^T\Psi_i(j)\right] + \mu_i\alpha_iN_{z,i}^T\Psi_i(j) + 2\mu_iN_{y,i}^TQ_iF_{y,i}x_{0,i}(k) - 2\mu_iN_{y,i}^TQ_i\mathrm{r}_i(k)$$

To realize these difference equations as a state-space representation, let the state, input, output, and input disturbance variables be defined as:

$$w_i(j) = \begin{Bmatrix}\mathrm{U}_i(j)\\ \mathrm{Z}_i(j)\\ \gamma_i(j)\end{Bmatrix},\ \hbar_i(j) = \begin{Bmatrix}\mathrm{V}_i(j)\\ \Psi_i(j)\end{Bmatrix},\ \eta_i(j) = \begin{Bmatrix}\mathrm{Z}_i(j)\\ \gamma_i(j)\end{Bmatrix}, \text{ and } \mathscr{g}_i(k) = \begin{Bmatrix}x_{0,i}(k)\\ \mathrm{r}_i(k)\end{Bmatrix}$$

Then the following state-space realization can be written:

$$\begin{aligned} w_i(j+1) &= \hat{A}_i w(j) + \hat{B}_i \hbar_i(j) + \hat{H}_i g_i(k) \\ \eta_i(j) &= \hat{C}_i w_i(j) \end{aligned} \tag{4.6}$$

where

$$\hat{A}_i = \begin{bmatrix} \beta_i & 0 & 0 \\ M_{z,i}\beta_i & 0 & 0 \\ 2\mu_i N_{y,i}^T Q_i M_{y,i}\beta_i & 0 & 0 \end{bmatrix}$$

$$\hat{B}_i = \begin{bmatrix} -(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i N_{y,i} \\ N_{z,i} - M_{z,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i N_{y,i} \\ 2\mu_i N_{y,i}^T Q_i N_{y,i} - 2\mu_i N_{y,i}^T Q_i M_{y,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i N_{y,i} \end{bmatrix}$$

$$\begin{bmatrix} -0.5\alpha_i(1-\beta_i)\mathrm{X}_i M_{z,i}^T \\ -0.5\alpha_i M_{z,i}(1-\beta_i)\mathrm{X}_i M_{z,i}^T \\ \mu_i N_{z,i}^T \alpha_i - \mu_i N_{y,i}^T Q_i M_{y,i}(1-\beta_i)\mathrm{X}_i M_{z,i}^T \alpha_i \end{bmatrix}$$

$$\hat{H}_i = \begin{bmatrix} -(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i F_{y,i} \\ F_{z,i} - M_{z,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i F_{y,i} \\ 2\mu_i N_{y,i}^T Q_i F_{y,i} - 2\mu_i N_{y,i}^T Q_i M_{y,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i F_{y,i} \end{bmatrix}$$

$$\begin{bmatrix} (1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i \\ M_{z,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i \\ 2\mu_i N_{y,i}^T Q_i M_{y,i}(1-\beta_i)\mathrm{X}_i M_{y,i}^T Q_i - 2\mu_i N_{y,i}^T Q_i \end{bmatrix}$$

and

$$\hat{C}_i = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}.$$

Next the definition of a dissipative system will be stated as well as the dissipativity conditions. Then the stability of a dynamical system (in state-space form) in terms of these conditions is given. First let us start with this definition:

**Definition 1**. A discrete time dynamical system with input, output and state $u, y$ and $x$, respectively, is said to be dissipative if there exists a function defined on the input and output variables, called the supply rate $s(u,y)\colon \Re^{n_u} \times \Re^{n_y} \to \Re$ and positive semidefinite function defined on the state, called the storage function $V\big(x(t)\big)\colon \Re^n \to \Re$ such that the following dissipation inequality:

$$V(x(k+1) - V(x(k))) \leq s(u(k), y(k)) \tag{4.7}$$

is satisfied for any $u$ at any time steps $k$.

In literature, the common used supply rate has the quadratic structure given as [85]:

$$s(u,y,T) = \sum_{k=0}^{T} \left( y(k)^T q_d y(k) + 2y(k)^T s_d u(k) + u(k)^T r_d u(k) \right)$$

or

$$\mathrm{s(u,y)} = y^T Q_d y + 2y^T S_d u + u^T R_d u \tag{4.8}$$

where

$$\mathrm{u} = [u_0^T \quad \cdots \quad u_L^T]^T,$$

$$\mathrm{y} = [y_0^T \quad \cdots \quad y_L^T]^T$$

$$Q_d = diag(q_{d,0} \quad \cdots \quad q_{d,L}),$$

$$S_d = diag(s_{d,0} \quad \cdots \quad s_{d,L}), \text{ and}$$

$$R_d = diag(r_{d,0} \quad \cdots \quad r_{d,L})$$

and $Q_d, S_d$, and $R_d$ are some given constant matrices.

With the supply rate given in (4.8) the following points can be stated:

1. For $Q_d = Q_d^T \in \Re^{n_y \times n_y}, S_d \in \Re^{n_y \times n_u}$ and $R_d = R_d^T \in \Re^{n_u \times n_u}$, the system is said to be $(Q_d, S_d, R_d)$-dissipative if [86]:

$$\mathrm{s(u,y)} = y^T Q_d y + 2y^T S_d u + u^T R_d u \geq 0.$$

2. As a special case if: $Q_d = -I, S_d = 0, R_d = k^2 I$, (where for $0 < k \leq 1$ we will have the $H_\infty$ case) then with (4.8) we would have the following input-output relation:

$$-\mathrm{y}^T\mathrm{y} + k^2\mathrm{u}^T\mathrm{u} \geq 0 \Rightarrow \mathrm{y}^T\mathrm{y} \leq k^2\mathrm{u}^T\mathrm{u} \Rightarrow \frac{\|\mathrm{y}\|_2^2}{\|\mathrm{u}\|_2^2} \leq k^2 \Rightarrow \frac{\|\mathrm{y}\|_2}{\|\mathrm{u}\|_2} \leq k.$$

The dissipativity of the local information dynamics given in (4.6) with a supply rate can be ensured by solving a Linear Matrix Inequality (LMI) problem as following:
The local information dynamics in (4.6) without the disturbance $\mathscr{g}_i(k)$ and disturbance dynamic $\widehat{H}_i$ is given by:

$$\begin{aligned} w_i(j+1) &= \hat{A}_i w_i(j) + \hat{B}_i \hbar_i(j) \\ \eta_i(j) &= \hat{C}_i w_i(j) \end{aligned} \tag{4.9}$$

Assuming the following local supply rate with pre-defined $\left(Q_{d,i}, S_{d,i}, R_{d,i}\right)$ matrices:

$$\mathrm{s}_i\left(\eta_i(j), q_i(j)\right) = \eta_i^T(j) Q_{d,i} \eta_i(j) + 2\eta_i^T(j) S_{d,i} q_i(j) + \hbar_i^T(j) R_{d,i} \hbar_i(j) \tag{4.10}$$

Then we need to satisfy the dissipativity inequality in (4.7) which with (4.10) can be written as:

$$V_i\left(w_i(j+1)\right) - V_i\left(w_i(j)\right) \leq \mathrm{s}_i\left(\eta_i(j), \hbar_i(j)\right)$$

Let us assume the following Lyapunov function on the state $w_i$:

$$V_i\left(w_i(j)\right) = w_i^T(j) \hat{P}_i w_i(j)$$

then we can extend the dissipativity inequality as:

$$w_i^T(j+1) \hat{P}_i w_i(j+1) \; - w_i^T(j) \hat{P}_i w_i(j) \; \leq$$

$$\eta_i^T(j) Q_{d,i} \eta_i(j) + 2\eta_i^T(j) S_{d,i} \hbar_i(j) + \hbar_i^T(j) R_{d,i} \hbar_i(j)$$

By introducing the local information dynamic (4.9), the above inequality becomes:

$$\{w_i^T(j) \quad \hbar_i^T(j)\}\begin{bmatrix}\hat{A}_i^T\hat{P}_i\hat{A}_i - \hat{P}_i - \hat{C}_i^T Q_{d,i}\hat{C}_i & \hat{A}_i^T\hat{P}_i\hat{B}_i - \hat{C}_i^T S_{d,i} \\ \hat{B}_i^T\hat{P}_i\hat{A}_i - S_{d,i}^T\hat{C}_i & \hat{B}_i^T\hat{P}_i\hat{B}_i - R_{d,i}\end{bmatrix}\begin{Bmatrix}w_i(j)\\ \hbar_i(j)\end{Bmatrix} \leq 0$$

Taking the matrix part only in the above inequality, we will have the following new inequality:

$$\begin{bmatrix}\hat{P}_i + \hat{C}_i^T Q_{d,i}\hat{C}_i - \hat{A}_i^T\hat{P}_i\hat{A}_i & \hat{C}_i^T S_{d,i} - \hat{A}_i^T\hat{P}_i\hat{B}_i \\ S_{d,i}^T\hat{C}_i - \hat{B}_i^T\hat{P}_i\hat{A}_i & R_{d,i} - \hat{B}_i^T\hat{P}_i\hat{B}_i\end{bmatrix} \geq 0$$

To write this inequality as a LMI, it also can be formulated as:

$$\begin{bmatrix}\hat{P}_i & \hat{C}_i^T S_{d,i} \\ S_{d,i}^T\hat{C}_i & R_{d,i}\end{bmatrix} - \begin{bmatrix}\hat{A}_i^T\hat{P}_i\hat{A}_i - \hat{C}_i^T Q_{d,i}\hat{C}_i & \hat{A}_i^T\hat{P}_i\hat{B}_i \\ \hat{B}_i^T\hat{P}_i\hat{A}_i & \hat{B}_i^T\hat{P}_i\hat{B}_i\end{bmatrix} \geq 0$$

or as sum of a linear part and quadratic part:

$$\begin{bmatrix}\hat{P}_i & \hat{C}_i^T S_{d,i} \\ S_{d,i}^T\hat{C}_i & R_{d,i}\end{bmatrix} - \begin{bmatrix}\hat{A}_i^T\hat{P}_i & -\hat{C}_i^T Q_{d,i} \\ \hat{B}_i^T\hat{P}_i & 0\end{bmatrix}\begin{bmatrix}\hat{P}_i^{-1} & 0 \\ 0 & -Q_{d,i}^{-1}\end{bmatrix}\begin{bmatrix}\hat{P}_i\hat{A}_i & \hat{P}_i\hat{B}_i \\ -Q_{d,i}\hat{C}_i & 0\end{bmatrix} \geq 0$$

Using Schur complement, the following LMI with the variable $P_i$ can expressed:

$$\begin{bmatrix}\widehat{P}_i & \widehat{C}_i^T S_{d,i} & \widehat{A}_i^T\widehat{P}_i & -\widehat{C}_i^T Q_{d,i} \\ S_{d,i}^T\widehat{C}_i & R_{d,i} & \widehat{B}_i^T\widehat{P}_i & 0 \\ \widehat{P}_i\widehat{A}_i & \widehat{P}_i\widehat{B}_i & \widehat{P}_i & 0 \\ -Q_{d,i}\widehat{C}_i & 0 & 0 & -Q_{d,i}\end{bmatrix} > 0 \tag{4.11}$$

Stability of a dissipative system is given next but we need to state the following definition of local zero-state detectability for a dynamic system:

**Definition 2** [87]. A dynamic system Σ that is described by the nonlinear state-space:

$$x(k+1) = f\big(x(k)\big) + g\big(x(k)\big)u(k)$$
$$y(k) = h\big(x(k)\big) + q\big(x(k)\big)u(k)$$

is locally zero-state detectable if there exists a neighborhood $N$ of $x = 0$ such that, for all $x \in N$:

$$y(k)|_{u(k)=0} = h\big(\phi(k;x;0)\big) = 0 \text{ for all } k \in Z_+$$

which implies $\lim_{k\to\infty} \phi(k;x;0) = 0$.

where $\phi(k;x;0) = f^k(x)$ is a trajectory of the unforced system $x(k+1) = f(x(k))$ from $x(0) = x$. If $N = \Re^n$, the system is zero-state detectable.

Definitions 1 and 2 fulfill the following stability criterion of a dissipative system which is also stated in [88]: If the system Σ is zero-state detectable and dissipative with respect to the supply rate (4.8), then the system is Lyapunov stable if $Q_d \leq 0$ and asymptotically stable if $Q_d < 0$.

For the rest of this chapter we will assume that the dynamic (4.9) is zero-state detectable for $i = 1, \cdots, p$.

### 4.3.2 The Network Dissipativity

In the previous subsection we discussed the dissipativity of a dynamic system and presented a stability criterion based on this dissipativity. Using the same criteria, in this section we try to prove the stability of a networked system. This will give us different methods by which we can have new convergence conditions for the LC-DMPC algorithm. Theorem III presents a systemwide convergence condition for the suboptimal LC-DMPC algorithm based on local finite gains and network couplings only.

**Theorem III:** Let subsystem $\Sigma_i$ have a finite gain $k_i$ from the local information input $\hbar_i = [\mathrm{V}_i^T \quad \Psi_i^T]^T$ to the local information output $\eta_i = [\mathrm{Z}_i^T \quad \gamma_i^T]^T$ for $i = 1,2,\cdots p$, then if: $I - H^T K H \geq 0$ where $K = diag(k_1 I \quad \cdots \quad k_p I)$, and $H$ is mapping the network

output $\eta$ to the network input $\hbar$: $\hbar = H\eta$, the interconnected network system is stable i.e. the LC-DMPC algorithm is converging.

**Proof:**

For each subsystem let the matrices: $Q_i, S_i, R_i$ be given with $Q_i$ & $R_i$ symmetric, then we can say that the subsystem is $Q_i, S_i, R_i$ dissipative from the input $\hbar_i = [\mathrm{V}_i^T \quad \Psi_i^T]^T$ to the output $\eta_i = [\mathrm{Z}_i^T \quad \gamma_i^T]^T$ if:

$$w_i(\eta_i, \hbar_i) = \eta_i^T Q_i \eta_i + 2\eta_i^T S_i \hbar_i + \hbar_i^T R_i \hbar_i \geq 0 \tag{4.12}$$

By summing (4.12) over all $\Sigma_i$ in the network we get:

$$\mathrm{w}(\eta, \hbar) = \eta^T Q \eta + 2\eta^T S \hbar + \hbar^T R \hbar \tag{4.13}$$

where

$$\hbar = \left[\hbar_1^T, \cdots, \hbar_p^T\right]^T, \ \hbar = \left[\hbar_1^T, \cdots, \hbar_p^T\right]^T,$$

$$Q = diag\left(Q_1, \cdots, Q_p\right), \ S = diag\left(S_1, \cdots, S_p\right), \text{ and } \ R = diag(R_1, \cdots, R_p)$$

For a local subsystem we can have the following relation between the local subsystem input and the total network output:

$$\hbar_i = \begin{Bmatrix} \mathrm{V}_i \\ \Psi_i \end{Bmatrix} = H_i \eta$$

where $H_i$ is defined as: for $\hbar_i \in \Re^{l_i}$ and $\eta \in \Re^L$ then for $\hbar_i = \mathrm{col}(u_j^i)$ and $\eta = \mathrm{col}(\eta^h)$, $i = 1, \cdots l_i$, $h = 1, \cdots L$ and $\hbar_j^i = H_i(i,h)\eta^h$, $H_i(i,h) = 1$ if $\hbar_j^i = \eta^h$ and 0 otherwise.

Therefor for $p$ subsystems in the network:

$$\hbar = \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix} \eta = \mathrm{H}\eta$$

Then we can introduce this global relationship into the total supply rate (4.13) as:

$$\mathrm{w}(\eta, \hbar) = \eta^T Q\eta + 2\eta^T SH\eta + \eta^T H^T RH\eta$$

Thus the total supply rate is now only a function of the total network output $\eta$:

$$\mathrm{w}(\eta) = \eta^T [Q + 2SH + H^T RH]\eta = \eta^T [Q + 2SH + H^T RH]\eta$$

therefore for network stability:

$$\eta^T (Q + 2SH + H^T RH)\eta \leq 0$$

or

$$Q + 2SH + H^T RH \leq 0 \Rightarrow -Q - 2SH - H^T RH \geq 0$$

For a special case we have $Q = -I, S = 0, R = K = diag(k_1 \quad \cdots \quad k_p)$, then:

$$I - H^T KH \geq 0 \tag{4.14}$$

A similar result is also given by [86] (as Theorem 5) for stability of interconnected large-scale systems with single-input, single-output subsystems.

The network dissipativity condition given in (4.14) can be expressed as a different problem. It can be formulated as the problem of finding the diagonal matrix $K$ such that (4.14) is satisfied. This new problem can be easily stated as a LMI as following:

With Schur complement, (4.14) can be written as:

$$\begin{bmatrix} I & H^T K \\ KH & K \end{bmatrix} \geq 0$$

Which also can be formulated as an optimization problem of finding the maximum trace of the matrix $K = diag(k_1 \quad \cdots \quad k_p)$:

$$\text{Maximize trace } (K) \tag{4.15}$$

such that:

$$K \geq 0$$

$$\begin{bmatrix} I & H^T K \\ KH & K \end{bmatrix} \geq 0$$

In Theorem III (which is based on the small gain theorem) we still need to check the convergence at the centralized level; however, we do not require information of the local dynamics or cost functions. The only required information are the gains of the dissipativity for the local subsystems, i.e. $k_i$, for $i = 1,2,\cdots,p$, and the network topology $H$. A naive way is to choose a local gain $k_i$ less or equal to one and not checking for network dissipativity. This is because for any two dynamical systems $G_1$ and $G_2$, in a loop, we can write [89]:

$$\| G_1 \cdot G_2 \|_\infty \leq \| G_1 \|_\infty \cdot \| G_2 \|_\infty$$

Which, according to the small gain theorem, the stability (or convergence) around any loop in the network is guaranteed. Such an approach, however, would be conservative. It is desirable to make each subsystem to have a finite gain, as long as the network is satisfying theorem III.

At this point, to check the convergence of the LC-DMPC algorithm with suboptimal operation, we have the following three methods:

- For given local finite gains $k_i$, for $i = 1,2,\cdots,p$, and network topology H we can use Theorem III.
- We can solve the LMI problem given in (4.15) and then for $i = 1,2,\cdots,p$:
  1. $k_i = \min$(corresponding diagonal elements in $K$).
  2. Make the local subsystem to be dissipative with $k_i$, with the local free variables $\beta_i, q_i, s_i, \alpha_i$, and $\mu_i$.

- Or, we can select $k_i < 1$ for $i = 1,2,\cdots,p$.

For the example of the interconnected subsystems, we used last two methods to make each subsystem showing dissipativity behavior in the information dynamics with a finite gain. The local values for the variables $\alpha_i$ and $\mu_i$ for $i = 1,2,3$ for each method are shown in table 4.1**.** In both cases, we select $\beta_i = 0.1$ for all subsystems. Figure 4.4 illustrates the eigenvalues of convergence condition given by matrix (2.27a) with $\alpha_i$ and $\mu_i$ in table 4**.**1. All eigenvalues are inside the unit circle which indicates the stability (convergence) of the algorithm.

The following two points should be considered for the developed convergence methods:

- The presented methods for checking the convergence locally or using limited information (finite local gains and network topology) are still conservative where for the provided networked example, we can have a stable (i.e. converging) LC-DMPC algorithm just by choosing $\alpha_1 = 0.1$ and keep all other free local parameters values at one. Figure 4.5 shows the eigenvalues of (2.27a) with such case.
- The maximum value for $K$ that Theroem III can result is the identity matrix; however, this does not mean that it is better to choose the local gain $k_i = 1$ for all subsystems. For instance, in the network system shown in Figure 4.6, if we assume that $k_i = 1$ for $i = 1,\cdots,5$, Theorem III will not be satisfied while the results of problem (4.14) are: $k_1 = k_2 = k_4 = 1, k_3 = 0.4856$, and $k_5 = 0.8315$.

Figure 4.7 shows the values of the centralized cost function (4.4) using the resulted control actions with: $\alpha_1 = 0.1$ and all other free local parameters values at one, the free local parameters in table 1, and all free local parameters equal to zero, respectively.

The dissipativity inequality over the information dynamics may give an indication whether if two or more subsystems should be combined and treated as one subsystem. In case if the dissipativity condition (4.11) does not satisfied even when $\alpha_i$ and $\mu_i$ are either zeros or have very small values (close to zero), we may consider combining the undertaken subsystem with one or more of its neighbor subsystems. In a LC-DMPC network, two coupled subsystems $\Sigma_i$ and $\Sigma_{i-1}$ can be combined as following:

Let the dynamic of subsystem $\Sigma_i$ is given by:

$$\begin{aligned} x_i^+ &= A_i x_i + B_{u,i} u_i + B_{v,i} v_i \\ y_i &= C_{y,i} x_i \\ z_i &= C_{z,i} x_i + D_{z,i} u_i \end{aligned}$$

while subsystem $\Sigma_{i-1}$ is given by:

$$\begin{aligned} x_{i-1}^+ &= A_{i-1} x_{i-1} + B_{u,i-1} u_{i-1} + B_{v,i-1} v_{i-1} \\ y_{i-1} &= C_{y,i-1} x_{i-1} \\ z_{i-1} &= C_{z,i-1} x_{i-1} + D_{z,i-1} u_{i-1} \end{aligned}$$

Both subsystems are connected as shown in Figure 4.8a where we can write the local couplings as: $v_i = z_{i-1}$.

To combine $\Sigma_i$ and $\Sigma_{i-1}$ into one subsystem $\Sigma$, we can write:

$$\begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix}^+ = \begin{bmatrix} A_i & 0 \\ 0 & A_{i-1} \end{bmatrix} \begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix} + \begin{bmatrix} B_{u,i} & 0 \\ 0 & B_{u,i-1} \end{bmatrix} \begin{bmatrix} u_i \\ u_{i-1} \end{bmatrix} + \begin{bmatrix} B_{v,i} & 0 \\ 0 & B_{v,i-1} \end{bmatrix} \begin{bmatrix} v_i \\ v_{i-1} \end{bmatrix}$$

However, $v_i = z_{i-1} = C_{z,i-1} x_{i-1} + D_{z,i-1} u_{i-1}$, then:

$$\begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix}^+ = \begin{bmatrix} A_i & B_{v,i}C_{z,i-1} \\ 0 & A_{i-1} \end{bmatrix} \begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix} + \begin{bmatrix} B_{u,i} & B_{v,i}D_{z,i-1} \\ 0 & B_{u,i-1} \end{bmatrix} \begin{bmatrix} u_i \\ u_{i-1} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{v,i-1} \end{bmatrix} v_{i-1}$$

and dynamics of the regulated outputs and outputs for downstream neighbors are:

$$\begin{bmatrix} y_i \\ y_{i-1} \end{bmatrix} = \begin{bmatrix} C_{y,i} & 0 \\ 0 & C_{y,i-1} \end{bmatrix} \begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix}$$

$$z_i = [C_{z,i} \quad 0] \begin{bmatrix} x_i \\ x_{i-1} \end{bmatrix} + [D_{z,i} \quad 0] \begin{bmatrix} u_i \\ u_{i-1} \end{bmatrix}$$

Thus, subsystem Σ has the measured input disturbance $v_{i-1}$ (the input for $\Sigma_{i-1}$) and the downstream output $z_i$ (the output of $\Sigma_i$). Figure 4.8b shows the combined subsystem Σ.

Table 4.2 gives the three possible combinations of the subsystems for the example of the interconnected subsystems (more details are given in appendix B) with the corresponding local design parameters for dissipativity. As can be seen, combining subsystems 1 & 2 gives more reasonable results if it is compared with if subsystems 1 and 2 are dealt with as individual subsystems.

**Table 4.1: Local Convergence Parameters for the Three Interconnected Subsystems Example**

| Method | $k$ | $\alpha$ | $\mu$ | $\lambda(I - H^T K H)$ | |
|---|---|---|---|---|---|
| | | | | $\lambda_{max}$ | $\lambda_{min}$ |
| LMI Problem (4.14) | $k_1 = 1$<br>$k_2 = 1$<br>$k_3 = 1$ | $\alpha_1 = 0.80$<br>$\alpha_2 = 0.01$<br>$\alpha_3 = 0.80$ | $\mu_1 = 0.030$<br>$\mu_2 = 0.001$<br>$\mu_3 = 0.020$ | 0 | 0 |
| Selecting $k_i < 1$ | $k_1 = 0.98$<br>$k_2 = 0.92$<br>$k_3 = 0.99$ | $\alpha_1 = 0.80$<br>$\alpha_2 = 0.01$<br>$\alpha_3 = 0.80$ | $\mu_1 = 0.030$<br>$\mu_2 = 0.001$<br>$\mu_3 = 0.020$ | 0.08 | 0.01 |

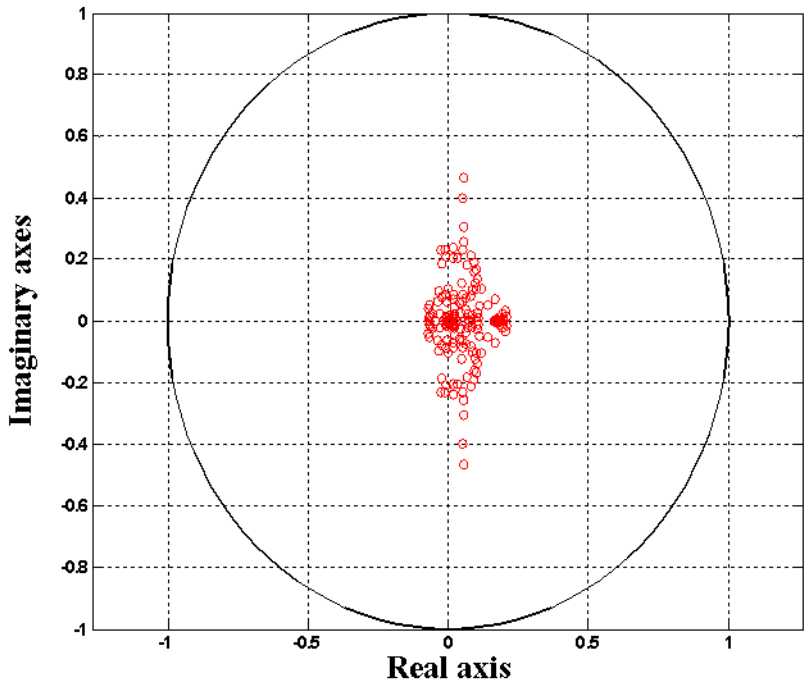



Figure 4.4: Eigenvalues of the convergence matrix (2.27a) for the interconnected network example with $\alpha_i$ and $\mu_i$ given in table 4.1

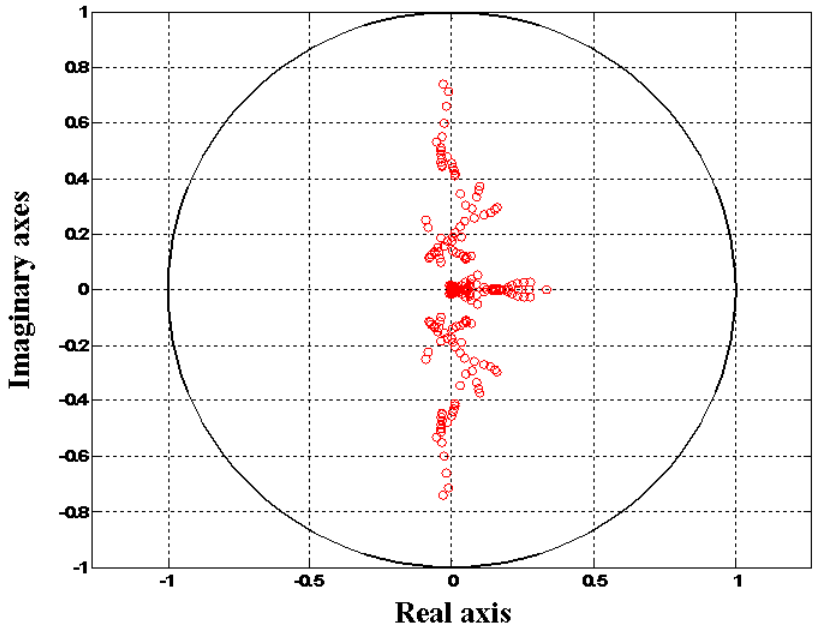



Figure 4.5: Eigenvalues of the convergence matrix (2.27a) for the interconnected network example with $\alpha_1 = 0.1$ and other local free parameters are hold at 1

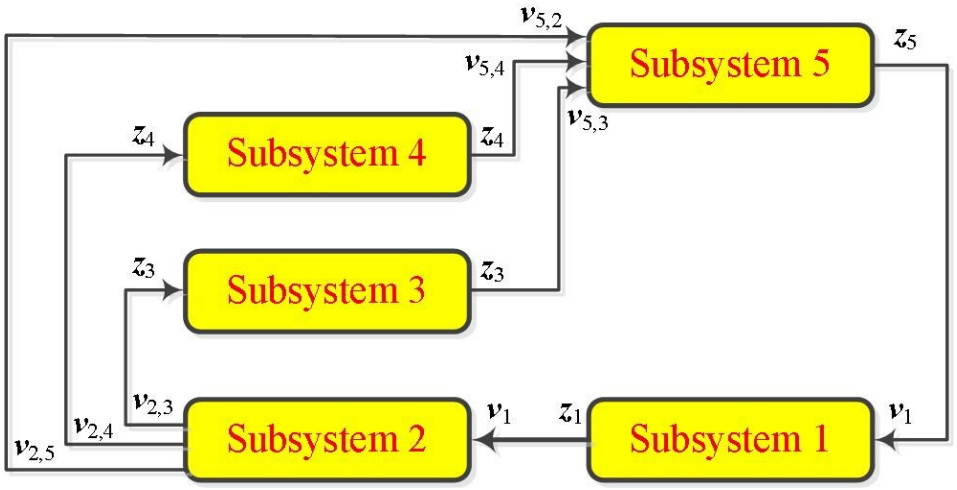



Figure 4.6: A network system with five interconnected subsystems with coupling in outputs (states) only

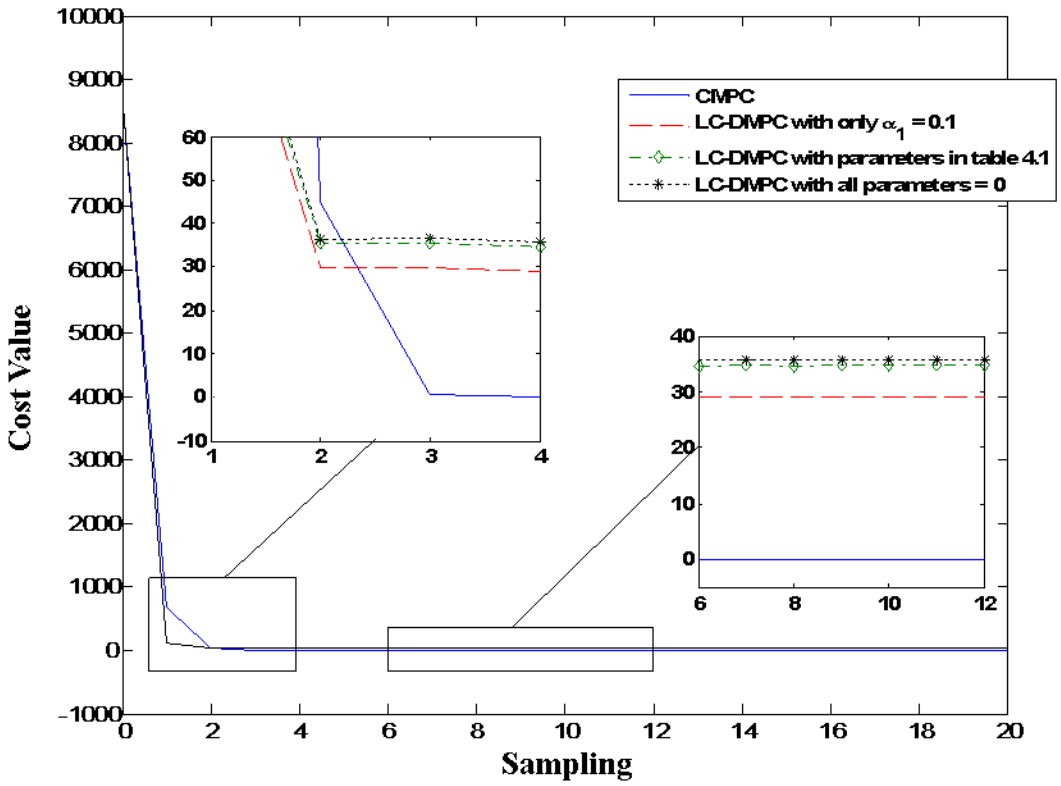
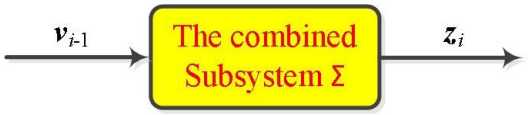



Figure 4.7: Cost values with solutions of a centralized MPC and the suboptimal LC-DMPC for the three interconnected subsystems example


(a): Two coupled subsystems in a LC-DMPC network




(b): The resulted combined subsystem

Figure 4.8: Combing two LC-DMPC subsystems

**Table 4.2: Combined Subsystems for the Three Interconnected Subsystems Example**

| Combined Subsystems | $k$ | $\alpha$ | $\mu$ |
|---|---|---|---|
| 1 & 2 | 1 | 1 | 0.15 |
| 1 & 3 | 1 | 1 | 0.038 |
| 2 & 3 | 1 | 1 | 0.012 |

### 4.4 Local Closed-Loop Stability

In Theorem II, the closed-loop stability is proven based on the assumption that the LC-DMPC converges to the centralized solution at each sampling. However, with theorem III, the subsystems are working in suboptimal levels meaning that the local control actions are not converging to the centralized control actions. Therefore we need to prove the closed-loop stability with a different method. Generally, the closed-loop stability of MPC is ensured by using a cost function with infinite horizon. However, solving an infinite constrained cost function is almost impossible. Therefore, researchers suggest combining a constrained finite horizon cost function with an infinite LQR problem as a terminal cost. Recent works also proposed to relax the LQR terminal weight by solving a Lyapunov stability inequality. For our problem, we suggest adding

terminal stabilizing costs at the end of the local distributed problems without terminal sets. With appropriate choice of the prediction horizon, adding a terminal cost would be sufficient for the MPC stability. To design such terminal costs in a distributed means, we modified the method that is given in [47] such that it can handle the distributed LC-DMPC cost functions.

### 4.4.1 MPC for Reference Tracking

As in this thesis, we are considering the reference tracking MPC (centralized as well as distributed) problems i.e. the servo problems, the undertaken systems will not converge to the origin but settle at some desired steady state point $(x_r, u_r)$. A combination of steady state and input control has to be selected to satisfy the desired tracking reference $r$ such that $x(k+1) = x(k) = x_r$ and $y(k) = r$.

For non-unique choices, the following simple optimization problem can be solved [90]:

$$\min_{x_r, u_r} u_r^T Q_r u_r \tag{4.16}$$

subject to:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{Bmatrix} x_r \\ u_r \end{Bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

$$u_{min} \leq u_r \leq u_{max}$$

To apply the standard MPC scheme with reference tracking case, a practical approach is to shift the origin of the problem by $(x_r, u_r)$ and then use the typical MPC methods (such as stability and feasibility) on the translated problem. In other words, penalize the

deviations of the original problem from the steady-state setpoint [91]. This means that for a standard regulation and unconstrained MPC problem given by:

$$\min_u \Theta_f\left(x\left(k+N_p\right)\right)+\sum_{i=0}^{N_p-1} l(x(k+i),u(k+i))$$

subject to:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ x(0) &= x_0 \end{aligned}$$

a reference tracking MPC problem can be formulated by shifting this problem such that it is centered at the steady-state setpoint resulted by problem (4.16). This new problem can be formulated as:

$$\min_u \Theta_f\left(x\left(k+N_p-x_r\right)\right)+\sum_{i=0}^{N_p-1} l(x(k+i)-x_r,u(k+i)-u_r)$$

subject to:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ x(0) &= x_0 \end{aligned}$$

In general, we assume that the tracking reference $r(k)$ is reachable where for a discrete LTI system that is given by:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k) \end{aligned}$$

and with constraints on control actions only ($u(k) \in \wp$), a reference signal $r(k)$ is said to be reachable if there exists a control signal such that:

- The control signal is admissible, i.e. $u(k) \in \wp$, and
- The control signal maps the above discrete system with the reference $r(k)$, i.e.

$$y(k) = r(k) = Cx(k)$$

Thus, the MPC problems will have feasible solutions only if the tracking signal is reachable.

In case if the reference signal is not reachable, then an approximate reachable reference that is closest to the original tracking reference can be calculated by a trajectory planner. Then the MPC problem is solved by so called two layer solver. In this structure, the first layer would be the trajectory planner which passes a reachable reference for the second layer where the MPC problem is.

At the trajectory planner layer, a reachable reference is computed by solving the following optimization problem:

$$\min_{x_r,u_r}(Cx_r - r)^T Q_r(Cx_r - r)$$

subject to:

$$x_r = Ax_r + Bu_r$$

$$u_r \in \wp$$

For the LC-DMPC approach, this trajectory planner is designed locally and each agent has to share its solution with neighbors in order to compute the reachable references locally. The following shows the optimization at the local trajectory planner level and a simple algorithm to compute the local reachable reference in a distributed fashion:
The local trajectory planner steady-state optimization problem is given as:

$$\min_{x_{r,i},u_{r,i}} \left(C_{y,i}x_{r,i} - r_i\right)^T Q_{r,i}\left(C_{y,i}x_{r,i} - r_i\right)$$

subject to:

$$x_{r,i} = A_i x_{r,i} + B_{u,i} u_{r,i} + B_{v,i} v_{r,i}$$

$$u_{min,i} \leq u_{r,i} \leq u_{max,i}$$

This problem can also be written as:

$$\min_{u_{r,i}} u_{r,i}^T \left[ M_{s,i}^T Q_{r,i} M_{s,i} \right] u_{r,i} + 2u_{r,i}^T \left[ M_{s,i}^T Q_{r,i} N_{s,i} v_{r,i} - M_{s,i}^T Q_{r,i} r_i \right]$$

subject to:

$$u_{min,i} \leq u_{r,i} \leq u_{max,i}$$

where

$$M_{s,i} = C_{y,i}(I - A_i)^{-1} B_{u,i}, \quad N_{s,i} = C_{y,i}(I - A_i)^{-1} B_{v,i}$$

While the following algorithm calculates the distributed reachable references:

**Algorithm 4.1: Distributed Reachable Reference Calculation**

**Input:** Number of iterations $\mathbb{N}_i$and set: $v_{r,i} = 0$.

**Start the Iterations:** For $j$ = 1 to $\mathbb{N}_i$:

Step 1: Exchange current information with local agents: $V_r(j+1) = \Gamma_s \mathrm{Z}_{\boldsymbol{r}}(j)$

Step 2: Solve the following optimization problem:

$$\min_{u_{r,i}} u_{r,i}^T \left[ M_{y,i}^T Q_{r,i} M_{y,i} \right] u_{r,i} + 2u_{r,i}^T \left[ M_{y,i}^T Q_{r,i} N_{y,i} v_{r,i} - M_{y,i}^T Q_{r,i} r_i \right]$$

subject to:

$$\begin{bmatrix} I_{r_{u,i}} \\ -I_{r_{u,i}} \end{bmatrix} u_{r,i} \leq \begin{Bmatrix} u_{max,i} \\ u_{min,i} \end{Bmatrix}$$

assign the solution as $u_{r,i}(j+1)$, and compute: $x_{r,i} = \left(I_{n_i} - A_i\right)^{-1} \left[ B_{u,i} u_{r,i} + B_{u,i} v_{r,i} \right]$.

Step 3: Update local output disturbance prediction:

$$z_{r,i}(j+1) = \left[C_{z,i}(I - A_i)^{-1}B_{u,i} + D_{z,i}\right]u_{r,i}(j+1) + \left[C_{z,i}(I - A_i)^{-1}B_{v,i}\right]v_{r,i}(j)$$

Next $j$

where $\Gamma_s$ is the interconnecting matrix for $N_p = 1$, and

$$V_r(j) = \begin{bmatrix} v_{r,1}^T(j) & v_{r,2}^T(j) & \cdots & v_{r,i}^T(j) \end{bmatrix}^T$$

$$Z_r(j) = \begin{bmatrix} z_{r,1}^T(j) & z_{r,2}^T(j) & \cdots & z_{r,i}^T(j) \end{bmatrix}^T$$

For non-reachable reference problems, we can simply replace the optimization problem in step 2 by (which is the distributed version of problem (4.16)):

$$\min_{x_{r,i}, u_{r,i}} u_{r,i}^T Q_{r,i} u_{r,i}$$

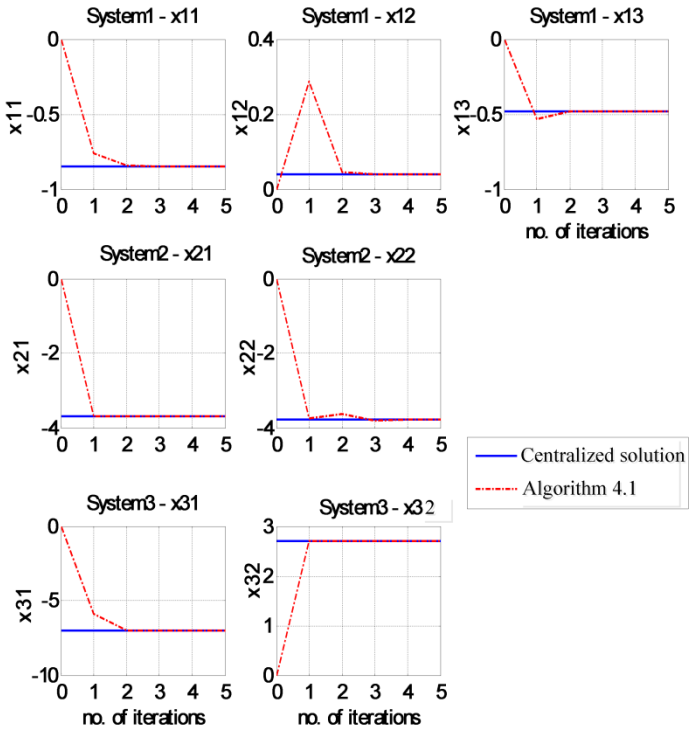
subject to:

$$\begin{bmatrix} A_i - I_i & B_{u,i} \\ C_i & 0 \end{bmatrix} \begin{Bmatrix} x_{r,i} \\ u_{r,i} \end{Bmatrix} = \begin{bmatrix} -B_{u,i}v_{r,i} \\ r_i \end{bmatrix}$$

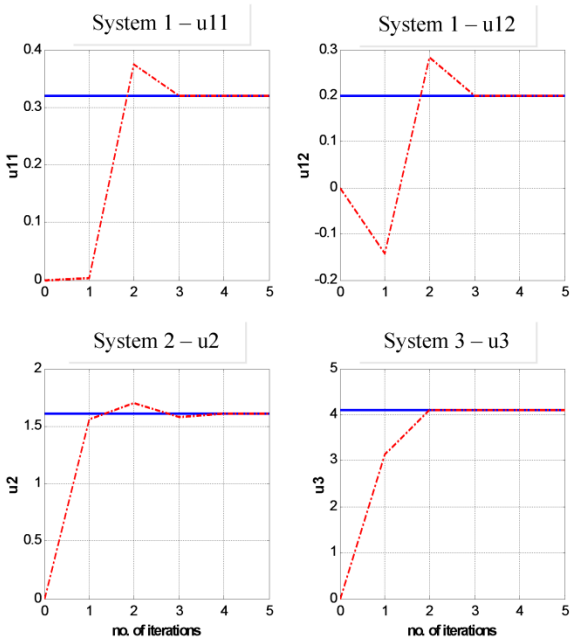
$$u_{min,i} \leq u_{r,i} \leq u_{max,i}$$

Figure 4.9 shows the application of Algorithm 4.1 versus a centralized solution for computing reachable references with the provided interconnected subsystems example.

The convergence of Algorithm 4.1 can be proven following the same method used with the main algorithm presented in chapter II (Algorithm 2.1).




(a) Steady-state states convergence




(b) Steady-state control actions convergence

Figure 4.9: Application of Algorithm 4.1 with the coupled subsystems example

### 4.4.2 Design of the Stabilizing Local Terminal Weights

As we simply can use standard state regulating MPC solutions with the reference tracking problems, we will focus on the state regulation MPC problem to formulate the terminal stabilizing cost synthesis for the stability of the suboptimal LC-DMPC algorithm.

First let us consider the following unconstrained state regulator MPC problem:

$$V^*(x) = \min_u V_f\left(x\big(k+N_p\big)\right) + \sum_{i=0}^{N_p-1} l(x(k+i), u(k+i))$$

subject to:

$$x(k+1) = Ax(k) + Bu(k)$$
$$x(0) = x_0$$

Here both the stage cost $l(x,u)$ and the terminal cost $V_f(x)$ are assumed to be convex functions. Let $U^*(x)$ be the optimal control trajectory that is the result of solving the problem, and let $u^*(x)$ be the first element of the sequence $U^*(x)$ that will be applied to the dynamical system. Then according to the receding horizon fashion we have:

$$x(k+1) = Ax(k) + Bu^*(x) \tag{4.17}$$

As sufficient conditions for stability of (4.17), is to have a terminal control law $u_f(x)$ and $\kappa_\infty$ functions $\alpha_1, \alpha_2$, and $\alpha_3$ [92] such that:

$$\alpha_1(\|x\|) \le V_f(x) \le \alpha_2(\|x\|)$$

$$V_f\big(Ax(k) + Bu_f(x(k))\big) - V_f\big(x(k)\big) \le -\alpha_3(\|x(k)\|)$$

In other words, $V_f\big(x(k)\big)$ is a Lyapunov function for the closed-loop dynamic given by:

$$x(k+1) = Ax(k) + Bu_f(x(k)) \tag{4.18}$$

For a distributed subproblem in the LC-DMPC algorithm we have the following local problem:

$$J_i = \min_{u_i} \Big\{ V_{f,i}\big(x_i, x_{r,i}, N_p\big) + \sum_{j=1}^{N_p-1} \big(x_i(k+j) - x_{r,i}\big)^T q_i \big(x_i(k+j) - x_{r,i}\big) + \sum_{j=0}^{N_p-1} \Big[ \big(u_i(k+j) - u_{r,i}\big)^T s_i \big(u_i(k+j) - u_{r,i}\big) + \alpha_i \psi_i^T(k+j) z_i(k+j) \Big] \Big\}$$

subject to:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) \\ y_i(k) &= C_{y,i} x_i(k) \\ z_i(k) &= C_{z,i} x_i(k) + D_{z,i} u_i(k) \end{aligned}$$

where we assumed that:

$$V_{f,i}\big(x_i, x_{r,i}\big) = \big(x_i(k+N_p) - x_{r,i}\big)^T P_i \big(x_i(k+N_p) - x_{r,i}\big)$$

We can write this problem in a more compact formula as:

$$J_i = \big(\mathrm{X}_i - \mathrm{X}_{r,i}\big)^T Q_i \big(\mathrm{X}_i - \mathrm{X}_{r,i}\big) + \big(\mathrm{U}_i - \mathrm{U}_{r,i}\big)^T S_i \big(\mathrm{U}_i - \mathrm{U}_{r,i}\big) + \alpha_i \Psi_i^T \mathrm{Z}_i$$

subject to:

$$\begin{aligned} \mathrm{X}_i &= F_{x,i} x_i(k) + M_{x,i} \mathrm{U}_i + N_{x,i} \mathrm{V}_i \\ \mathrm{Z}_i &= F_{z,i} x_i(k) + M_{z,i} \mathrm{U}_i + N_{z,i} \mathrm{V}_i \end{aligned}$$

where

$$X_{r,i} = \big[ x_{r,i}^T(k+1) \quad \cdots \quad x_{r,i}^T(k+N_p) \big],$$

$$U_{r,i} = \big[ u_{r,i}^T(k) \quad \cdots \quad u_{r,i}^T(k+N_p-1) \big]$$

$$Q_i = \begin{bmatrix} q_i & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P_i \end{bmatrix}$$

while other quantities are defined as in chapter II.

The new references $x_{r,i}$ & $u_{r,i}$ are the solution of Algorithm 4.1.

At this point we can state the following theorem that proves the stability of the local closed-loop stability with locally designed stabilizing terminal costs.

**Theorem IV:** If there exists functions $V_{f,i}(x_i), \Im_i(\psi_i), u_{f,i}(x_{\aleph,i})$ and $l_i\big(x_{\aleph,i}, u_{f,i}(x_{\aleph,i})\big)$ as well as $\alpha_{1,i}, \alpha_{2,i}$ and $\alpha_{3,i} \in \kappa_\infty$ such that for $i = 1,2,\cdots,p$:

$$\alpha_{1,i}(\|x_i\|) \leq V_{f,i}(x_i) \leq \alpha_{2,i}(\|x_i\|)$$

$$\alpha_{3,i}\big(\|x_{\aleph,i}\|\big) \leq l_i\big(x_{\aleph,i}, u_{f,i}(x_{\aleph,i})\big)$$

$$V_{f,i}\big(x_i(k+1)\big) - V_{f,i}\big(x_i(k)\big) \leq -l_i\left(x_{\aleph,i}, u_{f,i}\big(x_{\aleph,i}\big)\right) + \Im_i\big(\psi_i\big)$$

Then the function $V_{f,i}(x_i),$ is a Lyapunov function for the local distributed system under the control law $u_{f,i}(x_{\aleph,i})$, where $x_{\aleph,i}$ denotes the ordered set of neighbors' states of subsystem $i$ including the state of $i$.

**Proof:**

Before staring the proof, the following notes have been considered according to [47]:

- The following are defined as: the local feedback control law $u_{f,i}\big(x_{\aleph,i}\big) = K_{\aleph,i}x_{\aleph,i}$, the stage function $x_{\aleph,i}^T q_{\aleph,i} x_{\aleph,i} + {u_i}^T s_i u_i + \alpha_i \psi_i^T z_i$, $q_{\aleph,i} \in \Re^{n_{\aleph,i}}$ and the terminal cost $V_{f,i}(x_i) = x_i^T P_i x_i$.
- For each subsystem let: $\mho_i \in [0,1]^{n_i \times n}$ and $\zeta_i \in [0,1]^{n_{\aleph,i} \times n}$ be two mapping matrices such that: $x_i = \mho_i x,\ \ x_{\aleph,i} = \zeta_i x$ and consequently one can write: $x_i = \mho_i \zeta_i^T x_{\aleph,i}$.
- The coupling between two neighbor subsystems now can be defined as:

$$v_i = z_i = C_{z,i-1}x_{i-1} + D_{z,i-1}u_{i-1} = \big[C_{z,i-1} + D_{z,i-1}K_{\aleph,i-1}\zeta_{i-1}\mho_{i-1}^T\big]x_{i-1}$$

- With above definitions, the local subsystem dynamic would be given as:

$$x_i(k+1) = A_{\aleph,i}x_{\aleph,i}(k) + B_{u,i}u_i(k)$$
$$y_i(k) = C_{y,i}x_i(k)$$
$$z_i(k) = C_{z,i}x_i(k) + D_{z,i}u_i(k)$$

where

$$A_{\aleph,i} = \begin{bmatrix} A_i & B_{v,i}\left(C_{z,i-1} + D_{z,i-1}K_{\aleph,i-1}\zeta_{i-1}\mho_{i-1}^T\right) \end{bmatrix}$$

$$x_{\aleph,i} = \begin{Bmatrix} x_i \\ x_{i-1} \end{Bmatrix}$$

Now the last condition from theorem IV can be formulated as a set of the following inequalities for $i = 1,2,\cdots,p$:

$$\left[A_{\aleph,i}x_{\aleph,i} + B_{u,i}K_{\aleph,i}x_{\aleph,i}\right]^T P_i\left[A_{\aleph,i}x_{\aleph,i} + B_{u,i}K_{\aleph,i}x_{\aleph,i}\right] - x_{\aleph,i}^T\zeta_i\mho_i^T P_i\mho_i\zeta_i^T x_{\aleph,i} \leq$$

$$-\left[x_{\aleph,i}^T q_{\aleph,i}x_{\aleph,i} + x_{\aleph,i}^T K_{\aleph,i}^T s_i K_{\aleph,i}x_{\aleph,i} + \alpha_i{\psi_i}^T\left(C_{z,i}\mho_i\zeta_i^T x_{\aleph,i} + D_{z,i}u_i K_{\aleph,i}x_{\aleph,i}\right)\right] + \Im_i(\psi_i)$$

which also can be written as:

$$x_{\aleph,i}^T\left(\left[A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right]^T P_i\left[A_{N,i} + B_{u,i}K_{\aleph,i}\right] + q_{\aleph,i} + K_{\aleph,i}^T s_i K_{\aleph,i} - \overline{P}_i\right)x_{\aleph,i} +$$

$$\alpha_i{\psi_i}^T\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}u_i K_{\aleph,i}\right)x_{\aleph,i} - \Im_i(\psi_i) \leq 0$$

where: $\bar{P}_i = \zeta_i\mho_i^T P_i\mho_i\zeta_i^T$

Let us write this inequality in a different form as:

$$\Xi_i\left(x_{\aleph,i},\psi_i\right) - \Im_i(\psi_i) \leq 0 \tag{4.19}$$

where

$$\Xi_i\left(x_{\aleph,i},\psi_i\right) = x_{\aleph,i}^T\left(\left[A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right]^T P_i\left[A_{N,i} + B_{u,i}K_{\aleph,i}\right] + q_{\aleph,i} + K_{\aleph,i}^T s_i K_{\aleph,i} - \bar{P}_i\right)x_{\aleph,i} +$$

$$\alpha_i x_{\aleph,i}^T\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}u_i K_{\aleph,i}\right)^T\psi_i$$

We cannot solve inequality (4.19) as $x_{\aleph,i}$ appears in a non-quadratic form. Therefore, we have used a simple technique that has been imposed in [93] for stabilizing discrete-time

linear systems with unknown disturbances. The idea is that since inequality (4.19) must be satisfied for all $x_{\aleph,i}$, it also has to hold for the maximum value of $\Xi_i(x_{\aleph,i}, \psi_i)$ for any value of $\psi_i$. In order for $\Xi_i(x_{\aleph,i}, \psi_i)$ to have a maximum value at $\psi_i$, we assume that:

$$R_i = [A_{\aleph,i} + B_{u,i}K_{\aleph,i}]^T P_i [A_{N,i} + B_{u,i}K_{\aleph,i}] + q_{\aleph,i} + K_{\aleph,i}^T s_i K_{\aleph,i} - \bar{P}_i < 0$$

(which is guaranteed to be satisfied in the final result). This implies that $\Xi_i(x_{\aleph,i}, \psi_i)$ is a concave function of $x_{\aleph,i}$ with a maximum that can be computed as:

$$\frac{\partial}{\partial x_{\aleph,i}} \Xi_i(x_{\aleph,i}, \psi_i) = 0$$

where the maximum point is:

$$x_{\aleph,i}^*(\psi_i) = -0.5\alpha_i R_i^{-1}\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T \psi_i$$

Then the maximum value of the function $\Xi_i(x_{\aleph,i}, \psi_i)$ would be:

$$\Xi_i(x_{\aleph,i}^*, \psi_i) = -0.25{\alpha_i}^2 \psi_i^T \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) R_i^{-1}\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T \psi_i$$

Then (4.19) becomes:

$$\Xi_i(x_{\aleph,i}^*, \psi_i) - \Im_i(\psi_i) \leq 0 \tag{4.20}$$

We need now to solve inequality (4.20) for the stabilizing terminal costs, therefore let us assume that:

$$\Im_i(\psi_i) = \psi_i(k)^T \Phi_i \psi_i(k), \quad \Phi_i = \varphi_i I$$

for some unknown scalar $\varphi_i \geq 0$. Then (4.20) will be an inequality with $\psi_i$ appears in a quadratic form that can be written as:

$$\psi_i^T \left[\Phi_{\delta_i} + 0.25\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) R_i^{-1}\left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T\right] \psi_i \geq 0$$

where

$$\Phi_{\delta_i} = \frac{\Phi_i}{{\alpha_i}^2} = \frac{\varphi_i}{{\alpha_i}^2} I$$

Considering the definition of $R_i$ and applying Schur complement, the above inequality is equivalent to:

$$\begin{bmatrix} \Phi_{\delta_i} & \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) \\ \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T & -4\left(\left[A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right]^T P_i\left[A_{N,i} + B_{u,i}K_{\aleph,i}\right] + q_{\aleph,i} + K_{\aleph,i}^T s_i K_{\aleph,i} - \bar{P}_i\right) \end{bmatrix} \geq 0$$

which also can be reformatted as:

$$\begin{bmatrix} \Phi_{\delta_i} & \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) \\ \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T & 4\left(\bar{P}_i - q_{\aleph,i} - K_{\aleph,i}^T s_i K_{\aleph,i}\right) \end{bmatrix} -$$

$$\begin{bmatrix} 0 \\ 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right)^T \end{bmatrix} P_i \begin{bmatrix} 0 & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right) \end{bmatrix} \geq 0$$

Applying Schur complement a second time:

$$\begin{bmatrix} \Phi_{\delta_i} & \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) & 0 \\ \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T & 4\left(\bar{P}_i - q_{\aleph,i} - K_{\aleph,i}^T s_i K_{\aleph,i}\right) & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right)^T \\ 0 & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right) & {P_i}^{-1} \end{bmatrix} \geq 0$$

Once again this inequality can be written as:

$$\begin{bmatrix} \Phi_{\delta_i} & \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) & 0 \\ \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T & 4\bar{P}_i & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right)^T \\ 0 & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right) & {P_i}^{-1} \end{bmatrix} -$$

$$\begin{bmatrix} 0 & 0 \\ 2{q_{\aleph,i}}^{1/2} & 2K_{\aleph,i}^T {s_i}^{1/2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & 2{q_{\aleph,i}}^{1/2} & 0 \\ 0 & 2{s_i}^{1/2}K_{\aleph,i} & 0 \end{bmatrix} \geq 0$$

Through Schur complement, the final LIM inequality is given by:

$$\begin{bmatrix} \Phi_{\delta_i} & \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right) & 0 \\ \left(C_{z,i}\mho_i\zeta_i^T + D_{z,i}K_{x_{\aleph,i}}\right)^T & 4\bar{P}_i & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right)^T \\ 0 & 2\left(A_{\aleph,i} + B_{u,i}K_{\aleph,i}\right) & {P_i}^{-1} \\ 0 & 2{q_{\aleph,i}}^{1/2} & 0 \\ 0 & 2{s_i}^{1/2}K_{\aleph,i} & 0 \end{bmatrix}$$

$$\begin{matrix} 0 & 0 \\ 2{q_{\aleph,i}}^{1/2} & 2K_{\aleph,i}^T{s_i}^{1/2} \\ 0 & 0 \\ I & 0 \\ 0 & I \end{matrix}\Bigg] \geq 0$$

To finalize the proof, let us define the following: $\mathcal{E}_i = {P_i}^{-1}$ and $\mathcal{E} = diag(\mathcal{E}_1 \quad \cdots \quad \mathcal{E}_p)$, which give: $\bar{\mathcal{E}}_i = \zeta_i\mho_i^T{P_i}^{-1}\mho_i\zeta_i^T$, and $\mathcal{E}_{\aleph,i} = \zeta_i\mathcal{E}\zeta_i^T$. Also let $\lambda_{\aleph,i} = K_{\aleph,i}\mathcal{E}_{\aleph,i}$, then by pre- and post-multiplying the last LMI by $diag\left(I, \mathcal{E}_{\aleph,i}, I, I, I\right)$, and applying the change of variables, the final result would be:

$$\begin{bmatrix} \Phi_{\delta_i} & C_{z,i}\mho_i\zeta_i^T\mathcal{E}_{\aleph,i} + D_{z,i}\lambda_{\aleph,i} & 0 \\ \mathcal{E}_{\aleph,i}\zeta_i\mho_i^T C_{z,i}^T + \lambda_{\aleph,i}^T D_{z,i}^T & 4\bar{\mathcal{E}}_i & 2\mathcal{E}_{\aleph,i}A_{\aleph,i}^T + 2\lambda_{\aleph,i}^T B_{u,i}^T \\ 0 & 2A_{\aleph,i}\mathcal{E}_{\aleph,i} + 2B_{u,i}\lambda_{\aleph,i} & \mathcal{E}_i \\ 0 & 2{q_{\aleph,i}}^{1/2}\mathcal{E}_{\aleph,i} & 0 \\ 0 & 2{s_i}^{1/2}\lambda_{\aleph,i} & 0 \end{bmatrix}$$

$$\begin{matrix} 0 & 0 \\ 2\mathcal{E}_{\aleph,i}{q_{\aleph,i}}^{1/2} & 2\lambda_{\aleph,i}^T{s_i}^{1/2} \\ 0 & 0 \\ I & 0 \\ 0 & I \end{matrix}\Bigg] \geq 0 \quad (4.21)$$

This ends the proof and the final LMI has the following variables: $\Phi_{\delta_i}, \mathcal{E}_i, \bar{\mathcal{E}}_i,$ and $\mathcal{E}_{\aleph,i}$.

Each LC-DMPC agent needs to solve LMI (4.21) locally and off-line and for that it needs the dynamic of couplings. Now each subsystem needs to share the coupling matrices $C_{z,i}$ and $D_{z,i}$ as well as the matrices of $\mho_i$ and $\zeta_i$ with the local neighbors. If a

subsystem has a coupling through the control action with upstream subsystems, it also needs to know the control gain $K_{\aleph,i-1}$. Sharing of such information is limited as only coupled systems need to exchange information. Any changes in local dynamics, cost functions, or constraints do not require updating the neighbors as long as the coupling dynamics are not changed.

For subsystems with coupling in control actions as well as in states the LMI (4.21) has to be solved twice, first assuming coupling in state only (as $K_{\aleph,i-1}$ is unknown), and second time when $K_{\aleph,i-1}$ is available for exchange. To explain more, in the interconnected subsystems example, subsystem 1 for instance has two disturbances: $v_{1,2} = y_{2,1}$ the output from subsystem 2, and $v_{1,3} = u_3$ the control action of subsystem 3. Therefore:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mho_1 x = [I_3 \quad 0_{3\times 4}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

where states of subsystems 1, 2, and 3 are: $[x_1 \quad x_2 \quad x_3]^T$, $[x_4 \quad x_5]^T$, and $[x_6 \quad x_7]^T$, respectively. To compute the LMI in (4.21), first we assume coupling in states only (i.e. $v_1 = v_{1,2} = y_2 = C_{z,2}x_2$) thus:

$$x_{\aleph,1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \zeta_1 x = [I_5 \quad 0_{5\times 2}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Following the same procedure for subsystems 2 and 3 (in 3 there are only couplings in states) we will have the following:

$$\begin{bmatrix} x_4 \\ x_5 \end{bmatrix} = \mho_2 x = [0_{2\times 3} \quad I_2 \quad 0_{2\times 2}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}, x_{\aleph,2} = \begin{bmatrix} x_4 \\ x_5 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \zeta_2 x = \begin{bmatrix} 0_{3\times 2} & I_2 & 0_{2\times 2} \\ I_3 & 0_{3\times 2} & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix},$$

$$\begin{bmatrix} x_6 \\ x_7 \end{bmatrix} = \mho_3 x = [0_{2\times 5} \quad I_2] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}, \text{ and } x_{\aleph,3} = \begin{bmatrix} x_6 \\ x_7 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \zeta_3 x = \begin{bmatrix} 0_{2\times 5} & I_2 \\ I_5 & 0_{5\times 2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}.$$

After solving the LMI problem in (4.21), $K_{\aleph,i}, i = 1,2,3$ will be available then subsystems 1 and 2 need to share $K_{\aleph,1}$ and $K_{\aleph,2}$ and subsystem 1 has to update the matrix of $\zeta_1$ as states of subsystem 3 will affect it through $v_{1,3} = u_3 = K_{\aleph,3}\zeta_3\mho_3^T \begin{bmatrix} x_6 \\ x_7 \end{bmatrix}$, therefore $\zeta_1 = \zeta_{u,1} = I_7$. Subsystem 2 needs only to know $K_{\aleph,1}$ as the only states that are disturbing it are those of subsystem 1 which are already appearing in the disturbance $y_{1,1}$. Finally, subsystem 3 is not updating any information since it is not being affected on by any control action. Now subsystems 1 and 2 will need to solve the LIM problem one more time to consider the effects in control actions.

This procedure is summarized as following:

For coupling in states only:

Subsystem 1:

$$v_1 = \begin{bmatrix} v_{1,2} \\ v_{1,3} \end{bmatrix} = \begin{bmatrix} z_{2,1} \\ z_{3,1} \end{bmatrix} = \begin{bmatrix} y_{2,1} \\ u_3 \end{bmatrix} = \begin{bmatrix} y_{2,1} \\ 0 \end{bmatrix} = \begin{bmatrix} C_{z,2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$A_{\aleph,1} = [A_1 \quad B_{v,1}C_{z,2}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Subsystem 2:

$$v_2 = z_{1,2} = \begin{bmatrix} y_{1,2} \\ u_{1,2} \end{bmatrix} = \begin{bmatrix} y_{1,2} \\ 0 \end{bmatrix} = \begin{bmatrix} C_{z,1} \\ 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$A_{\aleph,2} = [A_2 \quad B_{v,2}C_{z,1}] \begin{bmatrix} x_4 \\ x_5 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Subsystem 3:

$$v_3 = \begin{bmatrix} v_{3,1} \\ v_{3,2} \end{bmatrix} = \begin{bmatrix} z_{1,3} \\ z_{2,3} \end{bmatrix} = \begin{bmatrix} y_{1,3} \\ y_{2,3} \end{bmatrix} = \begin{bmatrix} C_{z,1} & 0 \\ 0 & C_{z,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

$$A_{\aleph,3} = \left[A_3 \quad B_{v,3} \begin{pmatrix} C_{z,1} & 0 \\ 0 & C_{z,2} \end{pmatrix}\right] \begin{bmatrix} x_6 \\ x_7 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

For coupling in states and control actions:

Subsystem 1:

$$v_1 = \begin{bmatrix} v_{1,2} \\ v_{1,3} \end{bmatrix} = \begin{bmatrix} z_{2,1} \\ z_{3,1} \end{bmatrix} = \begin{bmatrix} y_{2,1} \\ u_3 \end{bmatrix} = \begin{bmatrix} C_{z,2} & 0 \\ 0 & D_{z,3} K_{\aleph,3} \zeta_3 \mho_3^T \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$A_{\aleph,1} = \begin{bmatrix} A_1 & B_{v,1} \begin{pmatrix} C_{z,2} & 0 \\ 0 & D_{z,3} K_{\aleph,3} \zeta_3 \mho_3^T \end{pmatrix} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

and for solving the LMI (4.21) for second time: $\zeta_1 = \zeta_{u,1} = I_7$

Subsystem 2:

$$v_2 = z_{1,2} = \begin{bmatrix} y_{1,2} \\ u_{1,2} \end{bmatrix} = [C_{z,1} + D_{z,1} K_{\aleph,1} \zeta_1 \mho_1^T] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$A_{\aleph,2} = \begin{bmatrix} A_2 & B_{v,2} \left( C_{z,1} + D_{z,1} K_{\aleph,1} \zeta_1 \mho_1^T \right) \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

While subsystem 3 will have the same matrices and structures as it does not have to solve the LMI problem (4.21) twice.

## 4.5 The Suboptimal LC-DMPC Algorithm

Based on the theorems and definitions that have been presented throughout this chapter, the following suboptimal LC-DMPC algorithm is stated. It has two subsection calculations: The off-line calculations where the local terminal stabilizing costs and

converging parameters are computed, and on-line calculations which is the real application of the algorithm. For constant (step function) references, Algorithm 4.1 can also be added to the off-line calculations. For dynamic references, however, Algorithm 4.1 should be integrated in the on-line calculation part.

**Algorithm 4.2: Suboptimal Unconstrained LC-DMPC Algorithm**

**Off-line Mode:** *Reachable references, terminal stabilizing costs, and network and local dissipativity.*

**For Reachable References:** Run Algorithm 4.1.

**For Terminal Costs and Network dissipativity:** Do the following:

**Input:** Local values of $q_i, s_i, \alpha_i, \mu_i, k_i, \zeta_i$ and $\mho_i$

**Step 1:** Share the coupling matrix dynamics: $C_{z,i}$ and $D_{z,i}$ with neighbors.

**Step 2:** Compute the LMI given in (4.21), assuming coupling in states only, i.e.

$$A_{\aleph,i} = [A_i \quad B_{v,i} diag(C_{z,1} \quad \cdots \quad C_{z,p})]$$

**Step 3:** For subsystems with coupling in control actions, update $\zeta_i$, if necessary, and share the initial computed $K_{\aleph,i}$. Re-compute the LMI problem in (4.21) with updated coupling dynamics:

$$A_{\aleph,i} = \left[A_i \quad B_{v,i} diag\left(C_{z,1}+D_{z,1}K_{\aleph,1}\zeta_1\mho_1^T \quad \cdots \quad C_{z,p}+D_{z,p}K_{\aleph,p}\zeta_p\mho_p^T\right)\right]$$

**Step 5:** Check the network dissipativity through Theorem III for given local finite gains, or solve the optimization problem (4.14) and distribute the results.

**Step 4:** Check the dissipativity of the local information dynamics by solving for $\hat{P}_i$ from the LMI problem (4.11). If not dissipative, change local values of $\alpha_i$ and/or $\mu_i$.

**Step 6:** If local information dynamics is not dissipative with given dissipativity matrices, change the dissipativity matrices or cost weighting matrices and go to step 2.

**<u>On-line Mode</u>:** *Real time implementation of the suboptimal LC-DMPC algorithm.*

**<u>Input</u>:** Number of iterations $\mathbb{N}_i$, initial values for states $x_{0,i}$ and error covariance $\mathfrak{H}_i$ and from the off-line mode get: $q_i, s_i, \alpha_i$, and $\mu_i$.

**Set:** initial values for $V_i = 0,\ U_i = 0,\ \Psi_i = 0$ or take previous values.

**Start of iteration:** For $j$ = 1 to $\mathbb{N}_i$ Do:

**Step 1:** Exchange current information with local agents:

$$\boldsymbol{\Psi}(j+1) = \left[\Psi_1(j+1), \cdots, \Psi_f(j+1)\right]^T = \boldsymbol{\Gamma}^T\left[\gamma_1(j), \cdots, \gamma_f(j)\right]^T = \boldsymbol{\Gamma}^T\boldsymbol{\gamma}(j)$$

$$\mathbf{V}(j+1) = \left[V_1(j+1), \cdots, V_f(j+1)\right]^T = \boldsymbol{\Gamma}\left[Z_1(j), \cdots, Z_f(j)\right]^{\boldsymbol{T}} = \boldsymbol{\Gamma}\mathbf{Z}(j)$$

**Step 2**: Set:

$$\mathrm{U}_i^{QP} = -\mathrm{X}_i\left[M_{x,i}^T Q_i F_{x,i} x_{0,i} + M_{x,i}^T Q_i N_{x,i} \mathrm{V}_i(j) + 0.5\alpha_i M_{z,i}^T \Psi_i(j) - M_{x,i}^T Q_i \mathrm{X}_{r,i} - S_i \mathrm{U}_{r,i}\right]$$

with $\mathrm{X}_i = \left[M_{x,i}^T Q_i M_{x,i} + S_i\right]^{-1}$

**Step 3**: For $\beta_i \in [0,1)$, set:

$$\mathrm{U}_i(j+1) \leftarrow \beta_i U_i(j) + (1-\beta_i)\mathrm{U}_i{}^{QP}(j)$$

**Step 4**: Update local output disturbance:

$$\mathrm{Z}_i(j+1) \leftarrow F_{z,i} x_{0,i} + M_{z,i}\mathrm{U}_i(j+1) + N_{z,i}\mathrm{V}_i(j)$$

**Step 5**: Update local sensitivity for input disturbance:

$$\gamma_i(j+1) \leftarrow \mu_i\big(2N_{x,i}^T Q_i N_{x,i}\mathrm{V}_i(j) + 2N_{x,i}^T Q_i F_{x,i} x_{0,i} + \alpha_i N_{z,i}^T \Psi_i(j) +$$

$$2N_{x,i}^T Q_i M_{x,i}\mathrm{U}_i(j+1) - 2N_{x,i}^T Q_i \mathrm{X}_{r,i}\big)$$

Next $j$

**End of iteration**

**Output:** First value of the computed control action $U_i$. (Inject this value into the local subsystem).

**Get:** new measurements for $y_i$, $u_i$, and $v_i$ then estimate the state.

**Go to: Start of iteration**.

**End of Algorithm 4.2**

Figure 4.10 shows the time response of the interconnected subsystems example with Algorithm 4.2 and parameter values given in table 4.1.

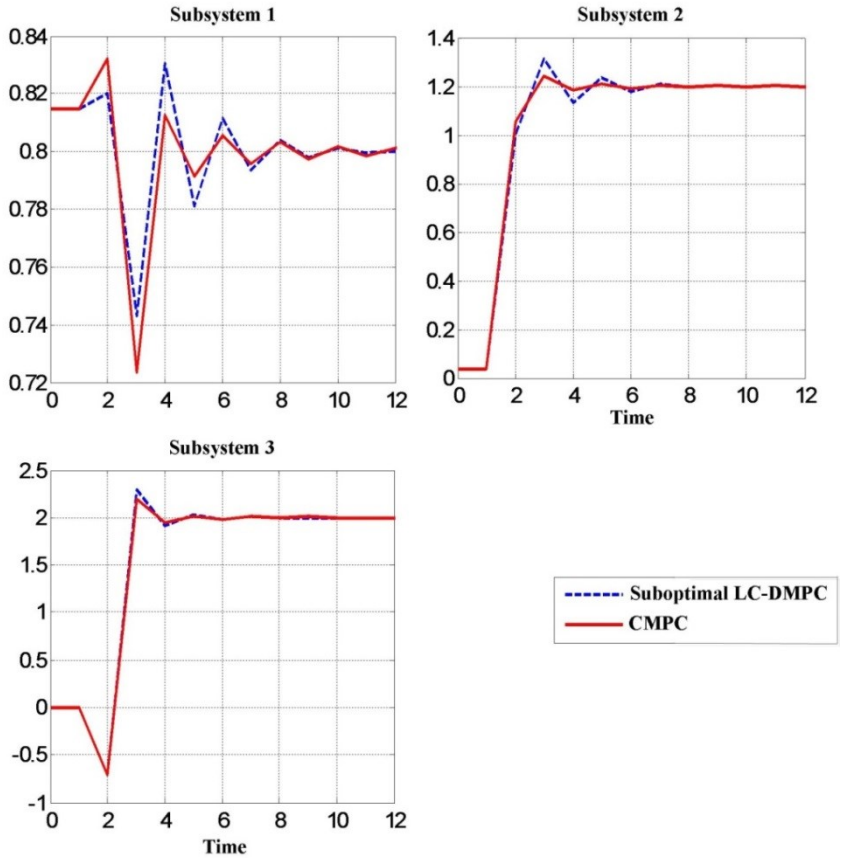



Figure 4.10: Time response of the interconnected subsystems example

## 4.6 Summary of the Chapter Statements

Throughout this chapter many theories, definitions, and algorithms were stated for the suboptimal LC-DMPC approach. This section summarizes all of the introduced statements through three points:

1- The suboptimality of the approach:

In order to distribute the convergence condition over the local controllers, two design variables are defined for each distributed agent. The values of these design variables are selected such that the local information dynamics conduct a dissipative behavior along the iteration domain. Each subsystem will have a finite gain associated with the solution of the local dissipativity inequality. However, these designed values will make LC-DMPC algorithm to operate in a suboptimal level compared with the systemwide performance.

2- The dissipativity of the network (or the convergence of the algorithm):

Using the theory of a dissipative system, the convergence of the suboptimal LC-DMPC algorithm can be proven or achieved by the following methods:

a) With the given finite gains for each subsystems in the network, Theorem III can be used using the network topology,
b) The LMI problem given in (4.14) is solved first with the network topology information. Then the result is distributed throughout the local agents. Each subsystem now can calculate the values for the design local variables such that the local information dynamics are dissipative by the corresponding distributed gain, or

c) Each subsystem is dissipative in its local information dynamics with gain less than one (the local free variables are selected to realize such dissipativity).

With the first two methods above, a centralized monitor is required but with less information (only local finite gains and network topology); however, last method can be implemented locally with local information only.

3- The closed-loop stability:

To design a stabilizable terminal cost for the distributed MPC, the LMI problem in (4.21) has to be solved locally by each agent. This requires the local agents to share the coupling dynamics with their neighbors. The introduced stabilizing problem is developed for state regulator MPC. The local problem with an LC-DMPC agent, however, is a reference tracking problem. Therefore, this last problem is shifted from the original in order for stander state regulation MPC techniques to be applied. Further work is introduced to compute reachable references that a local LC-DMPC agent can track without involving feasibility problems. Algorithm 4.1 is propped for computing reachable references locally where each agent shares its solution with downstream neighbors. This eliminates the need for a centralized monitor for the computation of reachable references.

As a final step, Algorithm 4.2 is introduced with two modes of calculations. The off-line mode involves the computations of reachable references, terminal stabilizing costs, and network convergence. And the on-line mode where the real time application of the suboptimal LC-DMPC approach is given.

# 5. APPLICATION OF THE LC-DMPC ALGORITHM FOR BUILDING HVAC SYSTEMS

In chapter II the optimal LC-DMPC algorithm is presented for networks of linear discrete coupled subsystems. In this chapter the algorithm is applied to control an HVAC system in a building with ten temperature controlled zones and the details of applying the algorithm is introduced and discussed. The chapter begins by presenting the software that is used to model the building zones and HVAC system and run and interface the algorithm. Then the LC-DMPC decomposition of the building HVAC system into a network of subsystems is introduced. The following section details the implemented local dynamic models, local economic cost functions, and distributed controller design. Finally the application of the LC-DMPC algorithm is demonstrated through a series of simulations.

## 5.1 Modeling and Actual Control Structure of the Simulated Building

### 5.1.1 Modeling of the Building Zones and HVAC System with EnergyPlus

EnergyPlus (EP) is a software available from the U.S. Department of Energy [94] which allows the users to model a whole building with complete HVAC systems. It also analyzes the energy consumption and gives the opportunity to apply and investigate different control strategies for buildings with the help of interfacing programs. The user can model the building with the fenestrations in 3-D using the freeware program SketchUp [95] as a first step. The SketchUp model and EP can be connected to each

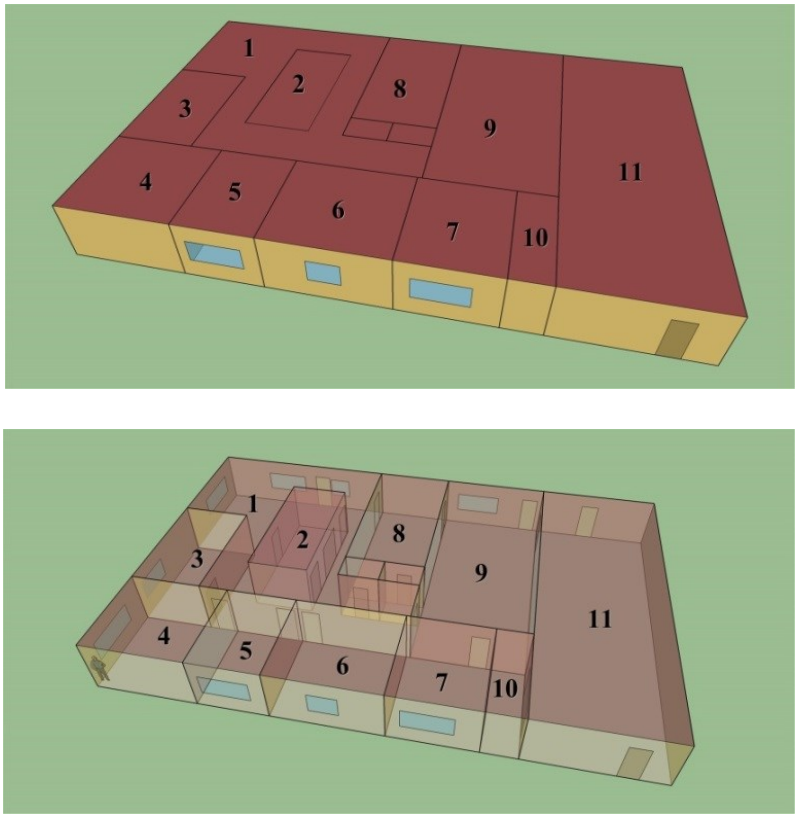
other through the software suite OpenStudio [96] to simplify the setup of the building. EP is a simulation and evaluation media only and in order to create the local LC-DMPC controllers for the building zones and HVAC systems, Matlab is used. In Matlab, the distributed controllers compute the setpoints and exchange information and at the beginning of a sampling, they inject the setpoints into EP through an interfacing software program. The Building Controls Virtual Test Bed (BCVTB) software [97] (developed at the Lawrence Berkeley National Laboratory in Berkeley, CA) is used as the interfacing media. However, this software is not user friendly as it does not provide an easy way for interfacing Matlab with EP, therefore, MLEP Matlab/Simulink [98] toolbox is sued as the main interfacing software. MLEP is an open source that is developed by University of Pennsylvania for users who are familiar with Matlab and Simulink.

The building used in simulation is the Utilities Business Office (UBO) at Texas A&M University in College Station, Texas. The building has eleven rooms with a separate Variable Air Volume (VAV) box serving each room (in this work a zone refers to a room with the corresponding VAV box). Figure 5.1 shows the SketchUp views. Real dimensions are used to draw and simulate the building HVAC control. All rooms are controlled with a PI controller except room 10 which is not controlled (a server room).

With the HVAC System Templates capability in EP, a basic and quick HVAC system is modeled for the building. The HVAC system has two main components: The main Air Handling Unit (AHU) and distributed Variable Air Volume (VAV) boxes per each room.

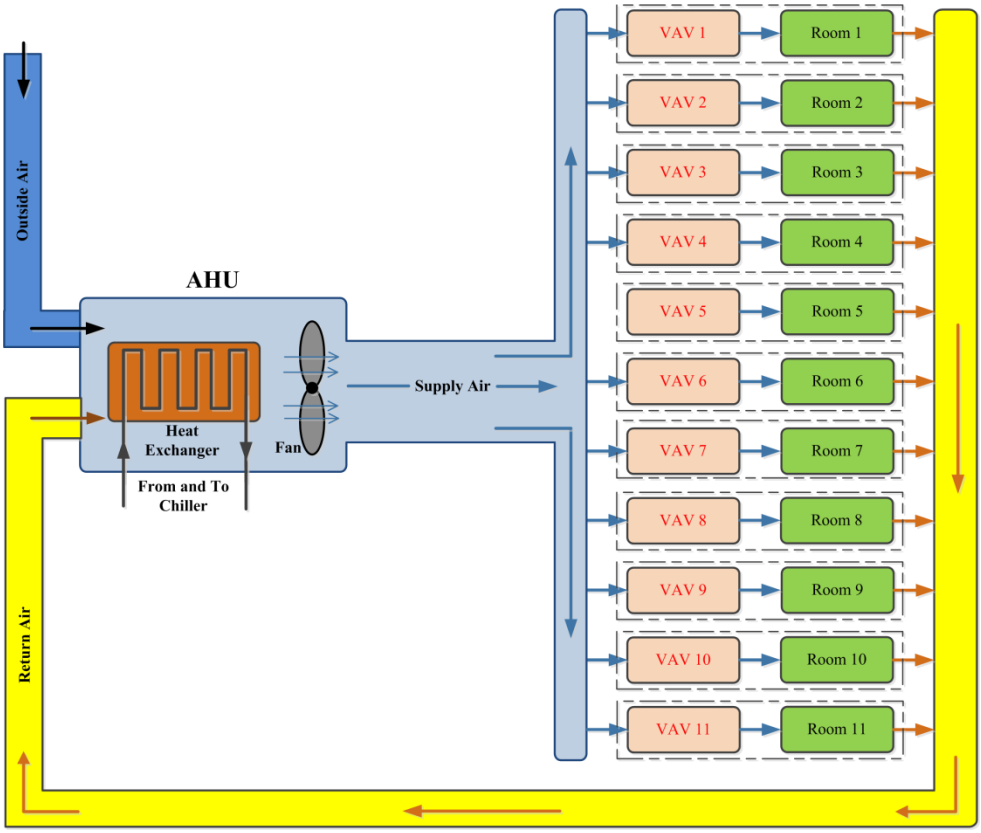
The AHU has a variable speed fan and heat exchanger. The fan provides the VAV boxes with treated supply air and the total volume of the supplied air can be controlled through the speed of the main fan. The supply air (which is the return air from rooms with some ratio of outside air) is regulated to the desired temperature through the heat exchanger using recycling chilled water from the main cooling tower. The amount of cooling or heating that can be provided by the heat exchanger is controlled through regulations of a valve position. Each zone has a VAV box, which adjusts the room temperature to a pre-user defined set-point through a damper. Figure 5.2 shows a basic schematic of the zones and HVAC systems and control structures of the UBO building.


Actual view

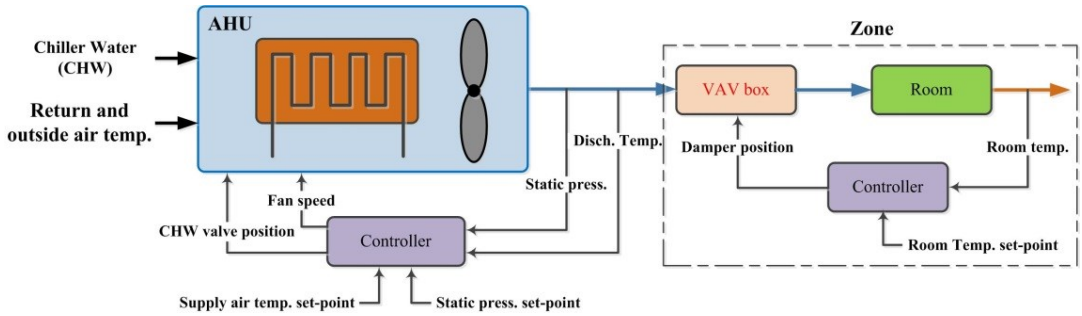



SketchUp side views

Figure 5.1: Views of the actual and simulation UBO building




The UBO HVAC system components




UBO HVAC control systems

Figure 5.2: UBO HVAC components and control systems

### 5.1.2 Demand Calculations and HVAC Control Systems in the UBO Building

In the real UBO building, at the AHU control structure, the following methods are used to calculate the demands for discharge air temperature and static pressure ($P_{EDS}$) using the local zone demands. These demands are then used as set-points for the AHU lower level controllers.

**Discharge air temperature demand**:

For each room, the cooling demand is calculated through a PI/PID control loop that is being fed with the difference between a user defined set-point and the actual room temperature as shown in Figure 5.3. Using all zone demands, the building level demand is then computed through the following equation:

$$\text{Discharge Air Demand} = 0.6 \cdot \text{average}(\text{cooling demand of zones}) + 0.4 \cdot \max(\text{cooling demand across zones}) \quad (5.1)$$

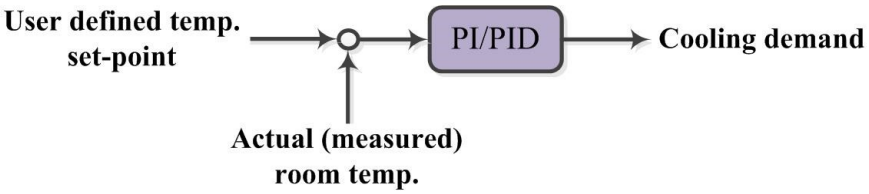



Figure 5.3: Local calculations of the cooling demand

**Static pressure ($P_{EDS}$) demand**:

For the $P_{EDS}$ demand a similar equation is sued as (5.1) but with local damper position information:

$$P_{EDS}\ \text{Demand} = 0.4 \cdot \text{average}(\text{zones VAV damper positions}) + 0.6 \cdot \max(\text{damper position across zone VAV boxes}) \quad (5.2)$$

The differences between the results from (5.1) and (5.2) and predefined set-points are then fed to the control loop at the Air Handing Unit (AHU) level to regulate the fan speed and supply air temperature. Figures 5.4 and 5.5 show the actual architecture of the UBO building HVAC system control.

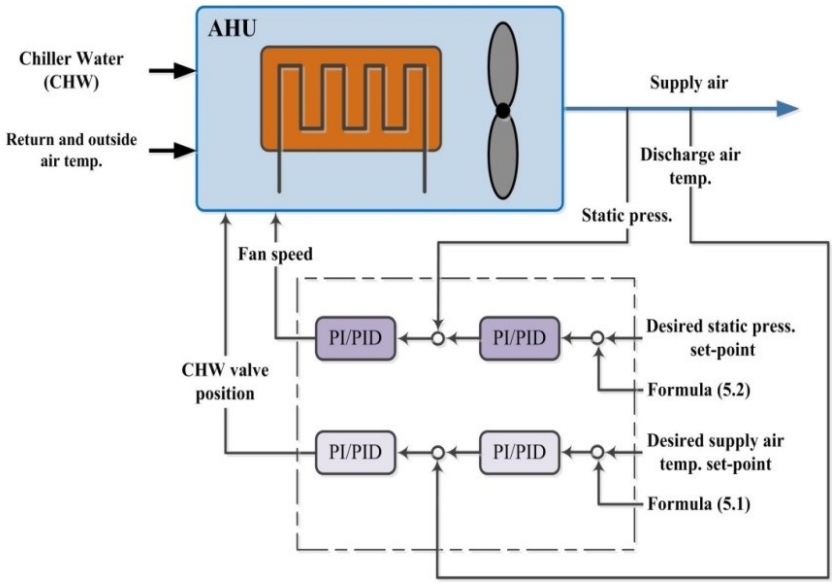



Figure 5.4: Local AHU controllers in the real UBO HVAC system

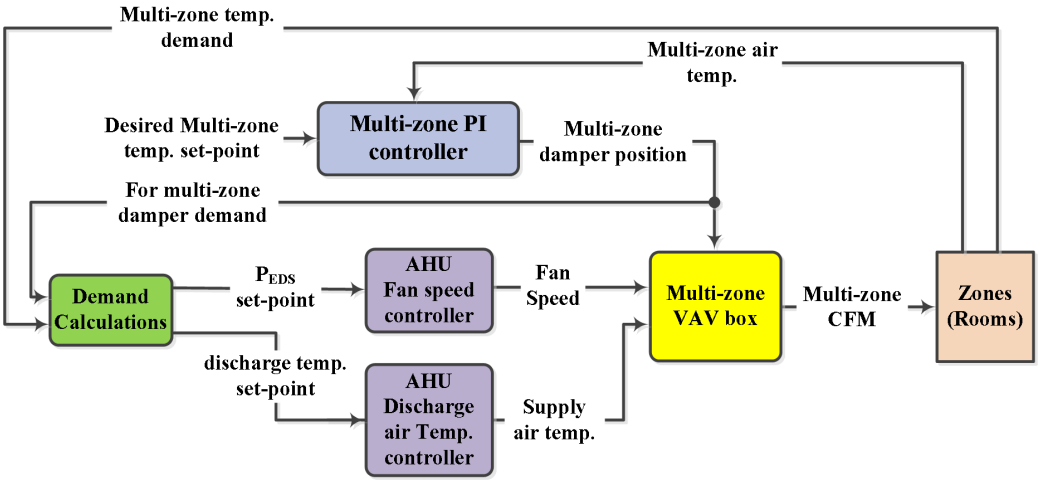



Figure 5.5: Actual control architecture in the UBO HVAC system

## 5.2 Application of the LC-DMPC Algorithm with the UBO Building

In this section we will explain how the subsystems are subdivided according to the LC-DMPC algorithm and show the local exchanged signals. Then the local cost functions as well as sensitivity calculations are detailed. Finally, the LC-DMPC distributed problems and algorithm are introduced.

### 5.2.1 Partitions of Local Subsystems

According to the upstream and downstream subsystem division followed in the LC-DMPC algorithm, we can subdivide the UBO building and HVAC systems into the following subsystems: Each room with corresponding VAV box (i.e. a zone) represents a subsystem in total of nine zone subsystems (rooms 10 and 11 have no control). The AHU will only have a static optimizer as no dynamical models are used to represent the unit. This optimizer computes the optimal $P_{EDS}$ and discharge air temperature set-points for AHU local controllers. Figure 5.6 shows the proposed subsystems.

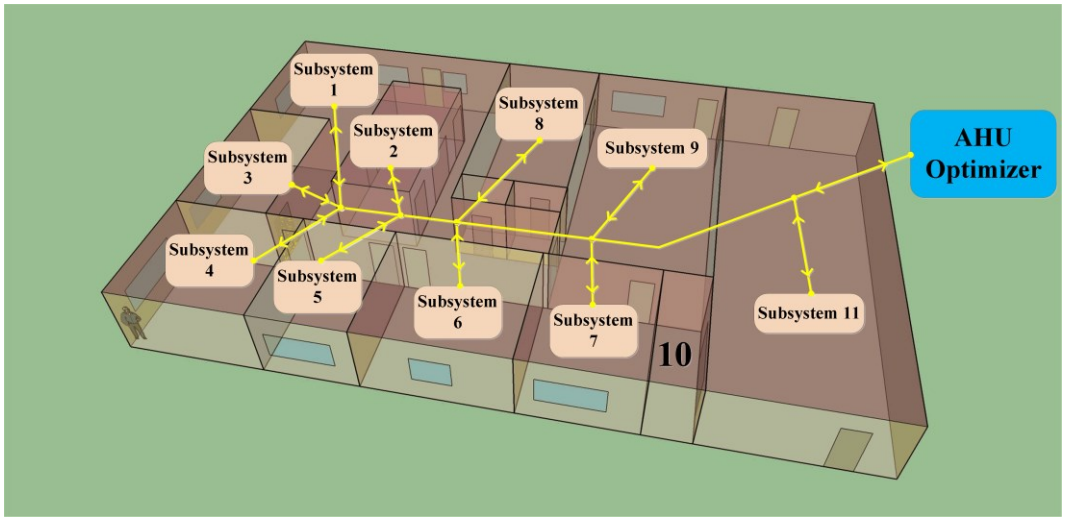



Figure 5.6: The LC-DMPC subsystems structure for the UBO building

#### 5.2.1.1 Local Room Model, Control Structure, and Zone Subsystem

The actual model of the lower level PI controllers, VAV box, and room is shown in Figure 5.7. There are two PI control loops. The first (or outer) loop produces a set-point for the inlet flow rate through the difference between the desired and actual room temperatures. This flow rate set-point is regulated against the supplied air flow through the second (or inner) PI control loop that adjusts the damper position inside the corresponding VAV box. The input disturbances to the room dynamics are the AHU discharge air temperature $T_{AHU}$, surrounding room temperatures, outside temperature $T_{outside}$, and relative humidity $RH_{outside}$.

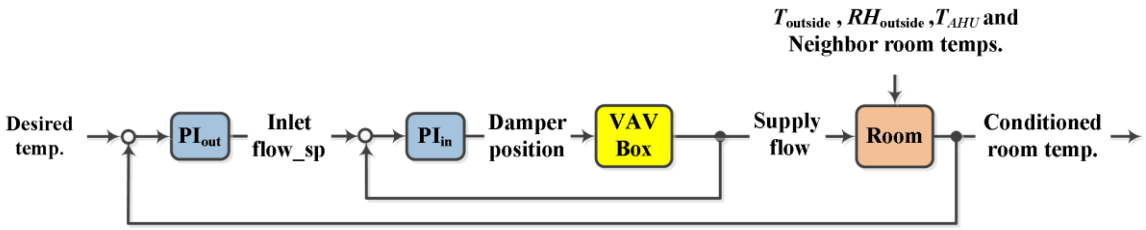



Figure 5.7: The actual control structure of a zone inside the UBO building

In this work, however, the dynamic between the "Inlet flow_sp" and "Supply flow" (i.e. the inner PI closed control loop) signals is assumed to be instant and perfect so that the "Inlet flow_sp" is equal to "Supply flow". The identification algorithm proposed in [99] was used to identify the individual room dynamics with the following state-space representation:

$$x_i(k+1) = A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_{d,i} d_i(k)$$

$$y_i(k) = C_i x_i(k)$$

where

$x_i$ is the state of room $i$,

$u_i$ is the fraction of the inlet flow rate

$v_i$ is the measured disturbing neighbor temperatures, and

$d_i$ is the unmeasured disturbances (outside temperature and relative humidity).

However, this realization does not consider the discharge air temperature of the AHU as one of the inputs. This is because the control input (cooling load) to the room dynamics is the product of the inlet flow are and the AHU discharge air temperature. This gives a bilinear room dynamic which the identification algorithm cannot identify (the algorithm can give only linear modes for the rooms). For us to consider the effect of the AHU discharge air temperature in the rooms and to have linear models at the same time, the following modification for the identified models was implemented:

Consider a simplified energy balance for a room under cooling demand as shown in Figure 5.8 where only the input and output heat energy are considered.

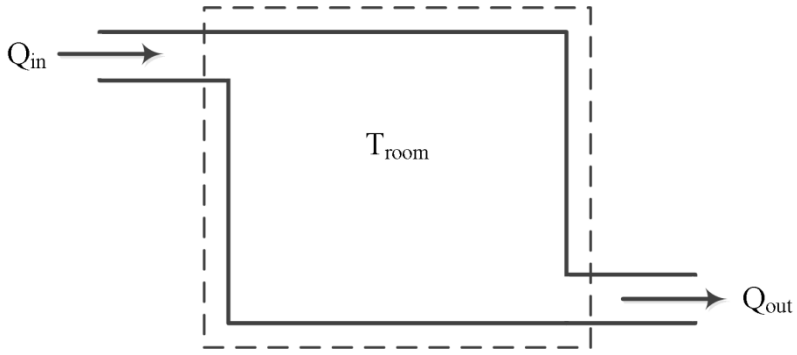



Figure 5.8: Simplified heat balance for a typical room

The heat balance can be written as:

$$\frac{d}{dt}\left(C_p T_{room} m_{air}\right) = \dot{Q}_{in} - \dot{Q}_{out} = \dot{m}_{in} C_p T_{AHU} - \dot{m}_{out} C_p T_{room}$$

or

$$\frac{d}{dt}T_{room} = \frac{\dot{m}_{fan}}{m_{air}}(T_{AHU} - T_{room})$$

where

$C_p$ is the heat capacity of the room

$m_{air}$ is the mass of the air inside the room (constant with time)

$\dot{m}_{fan}$ is the supplied inlet flow rate.

Now, the changes in $\dot{T}_{room}$ with respect to changes in $\dot{m}_{fan}$ and $T_{AHU}$ can be written as:

$$d\dot{T}_{room} = B_1 \cdot d\dot{m}_{fan}\big|_{T_{AHU}=cons} + B_2 \cdot dT_{AHU}|_{\dot{m}_{fan}=cons}$$

where

$$B_1 = \frac{1}{m_{air}}(T_{AHU} - T_{room})$$

$$B_2 = \frac{\dot{m}_{fan}}{m_{air}}$$

The coefficient $B_1$ (or $B_{u,i}$) is identified by the identification algorithm and for small variation in $T_{AHU}$, $B_2$ can be calculated from the relation:

$$B_2 = B_1 \frac{\dot{m}_{fan}}{(T_{AHU} - T_{room})}$$

Therefore, the variation in $\dot{T}_{room}$ for room $i$ can be expressed as:

$$d\dot{T}_{room_i} = B_{u,i}\left(d\dot{m}_{fan} + \frac{\dot{m}_{fan}}{(T_{AHU} - T_{room})}dT_{AHU}\right)$$

Note that the values of $\dot{m}_{fan}, T_{AHU}$, and $T_{room}$ are measurable. This simple modification changes the local zone control problem from a bilinear problem into a linear one.

The coefficient $B_2$ can be updated at each time step, or through some pre-defined schedule. In this work, however, we preferred to take the average of $B_2$ across the simulation of three months. With this modification, the final state space form of a room dynamic would be:

$$x_i(k+1) = A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_{t,i} T_{AHU}(k) + B_{d,i} d_i(k)$$

$$y_i(k) = C_i x_i(k) \tag{5.3}$$

where $B_{t,i}$ is the calculated coefficient $(B_2)$ for room $i$, and the AHU discharge air temperature $(T_{AHU})$ is treated as a measured disturbance.

Figure 5.9 shows the simplified local zone control structure sued for simulation.

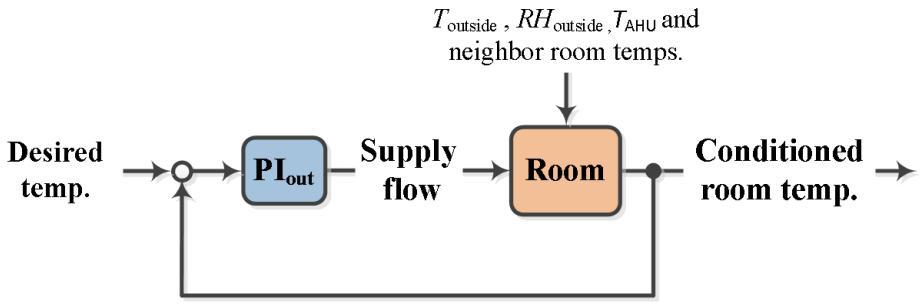



Figure 5.9: The simplified local zone control structure for the UBO building

Through the applied flow rate, the $P_{EDS}$ and AHU discharge air temperature disturb the room dynamics; however, the lower PI controller rejects this disturbance as long as the damper is not fully open. In this case, the inlet flow rate is a function of the damper position only and room temperature dynamics have no sensitivity for changes in the $P_{EDS}$ or $T_{AHU}$. For a fully opened damper, the entering air flow rate will be a function of the $P_{EDS}$ only (the damper is not controlling the flow rate any more) and room dynamics

will be sensitive for any variations in the fan $P_{EDS}$ and AHU discharge temperature $T_{AHU}$.

With the simplified room control and dynamics, the LC-DMPC partition for the room zones is now defined. A zone subsystem accepts an optimal temperature set-point from the local LC-DMPC agent. The local lower level PI control system regulates the room temperature around the optimum set-point while the local MPC modulates the occupant desired set-point with a pre-defined comfort level. The upstream disturbance inputs are the predicted neighbor room temperatures and AHU discharge air temperature. On the other hand, the outputs for downstream subsystems (neighbor zones and AHU) are the predicted room temperature and cos sensitivities. Outside air temperature and relative humidity are treated as exogenous disturbances. Figure 5.10 illustrates a zone subsystem with a local LC-DMPC agent as well as the lower level PI controller.

The rooms are simulated inside the EP and the identified models are used to design the LC-DMPC controllers. Local PI controllers and LC-DMPC agents are simulated inside the Matlab environment.

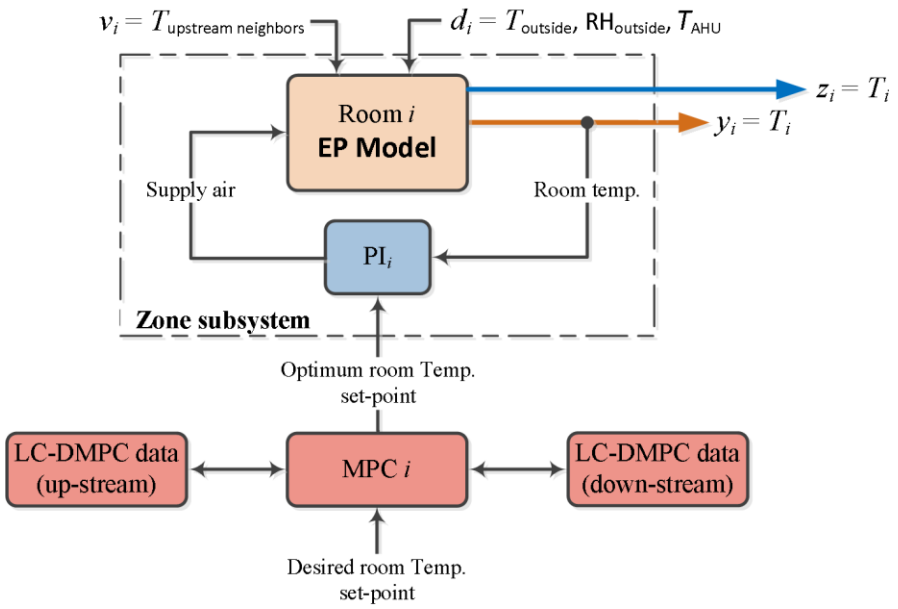



Figure 5.10: A zone subsystem in the UBO building with local LC-DMPC agent and PI controller

#### 5.2.1.2 The AHU Optimizer

Figure 5.11 shows the AHU lower level controllers (PI) with the set-point optimizer for the AHU discharge air temperature and fan speed. In this work, we assumed that the AHU outputs (i.e. the $P_{EDS}$ and discharge air temperature) will reach the optimum set-points through the local PI controllers in the next sampling. Therefore, from the LC-DMPC perspective, the AHU will only have a set-point optimizer where the dynamics are not considered in order to show the general concept of the HVAC system application for the LC-DMPC approach in a simple and trackable way.

The AHU optimizer computes the optimum set-points for the fan speed and AHU discharge using the sensitivities information fed from the zone local MPCs. The cost

function used by the optimizer is the summation of two economic costs in terms of fan $P_{EDS}$ and AHU discharge air temperature $T_{AHU}$.

The next section details the economic cost functions as well as the calculations of the zone sensitivities.

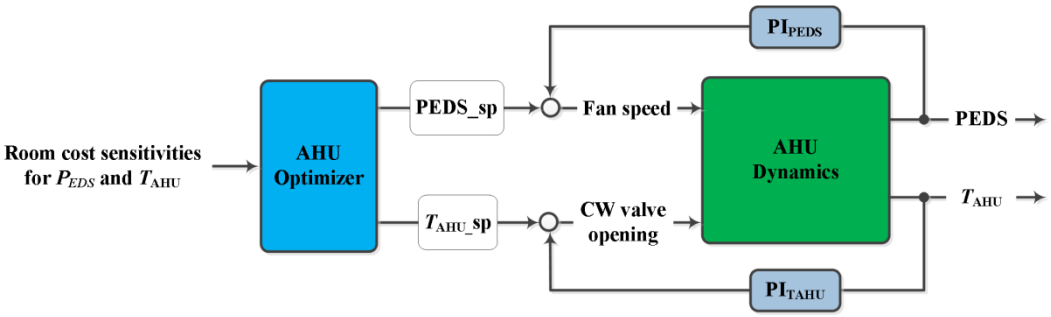



Figure 5.11: The AHU set-point optimizers and lower level controllers

### 5.2.2 Cost Functions and Sensitivities

In this section, the cost functions for zones and AHU optimizers as well as sensitivity calculations are derived. The cost functions used in this work are economic costs which mean that the local LC-DMPC controllers try to minimize actual financial costs while maintaining the comfort levels inside the rooms. The cost functions are not derived by the author of this dissertation and the reader is referred to appendix D for more details.

#### 5.2.2.1 Zone Cost Functions

At the level of a zone, the cost function for the local distributed MPC will have the following form:

$$J_i = \sum_{j=1}^{N_p} (e_i^T(k+j) \cdot \mathbb{Z}_i(k+j)) + \sum_{j=0}^{N_p-1} \psi_i^T(k+j) \cdot z_i(k+j) \qquad (5.4)$$

where

$e_i = T_{room,i} - T_{desired,i}$,

$z_i$ is the output disturbing temperatures for downstream neighbors that are weighted by the cost sensitivity of the neighbor zones $\psi_i$.

The variable $\mathbb{Z}_i$ is derived using the Predicted Mean Vote (PMV) equation [101] as well as Loss Of Productivity (LOP) relationships that are detailed in the appendix D. $\mathbb{Z}_i$ will be a function of the room *i* temperature $\left(T_{room,i}\right)$ with the following final form:

$$\mathbb{Z}_i\left(T_{room,i}\right) = \phi_i \cdot T_{room,i} + \varphi_i \tag{5.5}$$

where the derivations of $\phi_i$ & $\varphi_i$ are detailed in appendix D.

The term $\mathbb{Z}_i$ has a unit of $\$/℃$ where it relates the occupancy productivities ($) with the required comfort levels in terms of temperature (℃). Thus, it gives a tool for measuring and relating the activity of occupancy and the HVAC performance. The LOP measures how much the occupancies lose their productivities as they get more uncomfortable due to inadequate HVAC system.

Equation (5.5) is an affine linear expression that can be extended along the prediction horizon $N_p$ as shown below:

$$\begin{bmatrix} \mathbb{Z}_i(k+1) \\ \mathbb{Z}_i(k+2) \\ \vdots \\ \mathbb{Z}_i(k+N_p) \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_i & 0 & 0 & 0 \\ 0 & \phi_i & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \phi_i \end{bmatrix}}_{\mathcal{M}_i} \begin{bmatrix} T_{room,i}(k+1) \\ T_{room,i}(k+2) \\ \vdots \\ T_{room,i}(k+N_p) \end{bmatrix} + \underbrace{\begin{bmatrix} \varphi_i \\ \varphi_i \\ \vdots \\ \varphi_i \end{bmatrix}}_{\aleph_i} \tag{5.6}$$

Using the local predicted zone temperature vector $\mathrm{Y}_i$ where:

$$\mathrm{Y}_i = \begin{bmatrix} T_{room,i}(k+1) \\ T_{room,i}(k+2) \\ \vdots \\ T_{room,i}(k+N_p) \end{bmatrix}$$

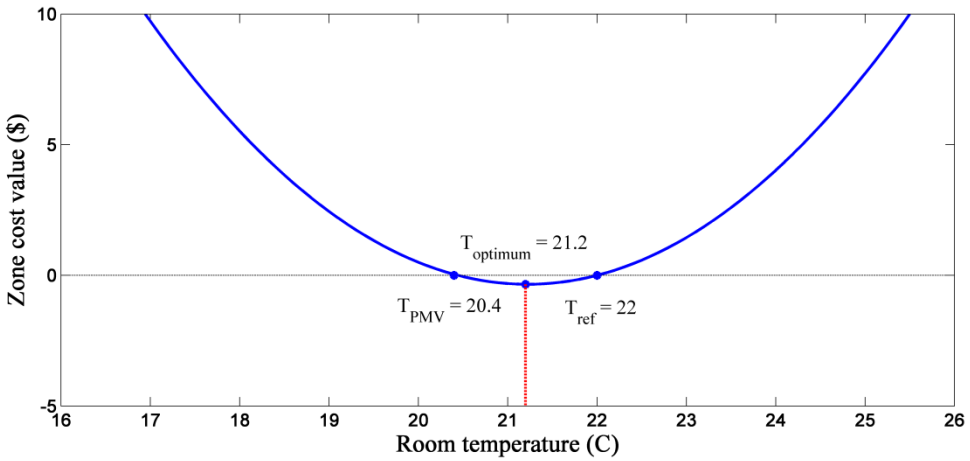
and the local desired reference $\mathrm{r}_i$, the zone cost $J_i$ (not considering the downstream costs) with (5.6) can be re-written as:

$$J_i = (\mathrm{Y}_i - \mathrm{r}_i)^T(\mathcal{M}_i\mathrm{Y}_i + \aleph_i) \tag{5.7}$$

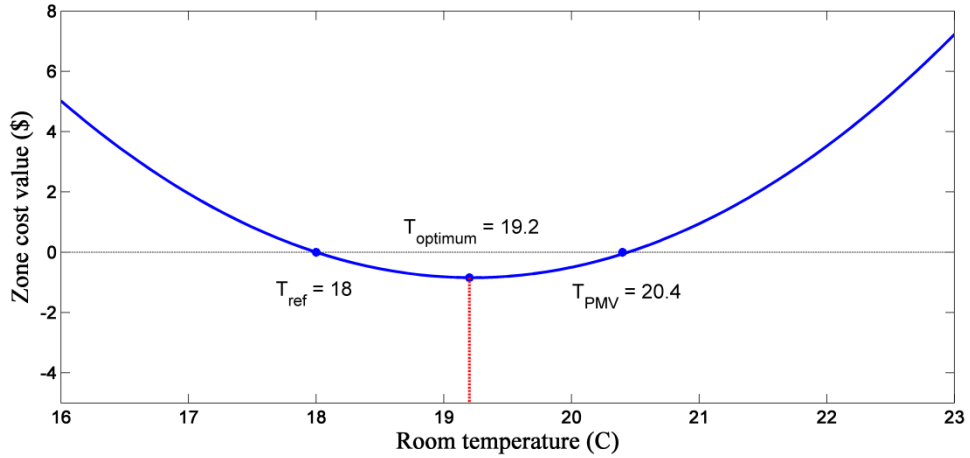
Generally speaking, there are two references in (5.7): the user defined reference $\mathrm{r}_i$ and the reference that is associated with the aggregate occupant comfort (i.e. PMV). The optimum reference, however, is some point between these two temperature references. Therefore, based on the desired room reference and comfort level, there will be two different cases for the optimum reference that the zone MPC attempts to approach. The first case is when both reference temperatures match. In this case the optimum reference will match the desired or comfort reference. The second case is when both references are different. Here, the zone MPC tries to track the average of the two references and the associated cost will be negative. The negative cost value is because either the desired reference (the room temperature is below the user-defined reference and the MPC still requiring more air) or the comfort reference (loss in productivity) is not met.

Figure 5.12 shows the cost function (5.7) with the second case for an office activity of Metabolic rate $M = 85\ W/m^2$ (the activity level of the occupancy) and clothing insulation $I_{cl} = 0.1162\ m^2\mathrm{k/W}$ (light business suit). The relative humidity is 50%, $N_p = 1$, productivity cost = \$1.0016 (typical yearly payment of \$25000), and with two different desired reference temperatures 22 and 18 ℃. In Figure 5.12 a, the optimum room temperature (that will be tracked by the local MPC) is 21.2 ℃ which is the average of the desired tempearure 22 ℃ and the PMV comfort reference 20.4 ℃. On the other

hand, the optimum reference for the local MPC is 19.2 ℃ when the desired reference changes to 18 ℃ as shown Figure 5.12 b.




(a) When desired room temperature is greater than the PMV comfort reference




(b) When PMV comfort reference is greater than the desired room temperature

Figure 5.12: Cost function (5.7) with different desired room temperature and same comfort level

With (5.7), using the local predicted temperature dynamics $Y_i$ and downstream output temperature dynamics $Z_i$ given by:

$$\mathrm{Y}_i = F_{y,i}x_{0,i} + M_{y,i}\mathrm{U}_i + N_{y,i}\mathrm{V}_i + P_{y,i}\mathrm{D}_i + P_{ty,i}\mathrm{D}_{TAHU}$$

$$\mathrm{Z}_i = F_{z,i}x_{0,i} + M_{z,i}\mathrm{U}_i + N_{z,i}\mathrm{V}_i + P_{z,i}\mathrm{D}_i + P_{tz,i}\mathrm{D}_{TAHU}$$

where $P_{ty,i}$ and $P_{tz,i}$ are computed in the same way that $P_{y,i}$ is computed but with $B_{t,i}$, and $\mathrm{D}_{TAHU}$ is the AHU discharge temperature along $N_p$, then the zone cost along $N_p$ can be expressed as:

$$J_i = \mathrm{U}_i^T\left[M_{y,i}^T\mathcal{M}_i M_{y,i}\right]\mathrm{U}_i +$$

$$2\mathrm{U}_i^T\left[M_{y,i}^T\mathcal{M}_i F_{y,i}x_{0,i} + M_{y,i}^T\mathcal{M}_i N_{y,i}\mathrm{V}_i + M_{y,i}^T\mathcal{M}_i P_{y,i}\mathrm{D}_i + M_{y,i}^T\mathcal{M}_i P_{ty,i}\mathrm{D}_{TAHU} + 0.5M_{z,i}^T\Psi_i\right.$$

$$\left.+0.5M_{y,i}^T\aleph_i - 0.5M_{y,i}^T\mathcal{M}_i\mathrm{r}_i\right] +$$

$$\mathrm{V}_i^T\left[N_{y,i}^T\mathcal{M}_i N_{y,i}\right]\mathrm{V}_i + 2\mathrm{V}_i^T\left[N_{y,i}^T\mathcal{M}_i F_{y,i}x_{0,i} + N_{y,i}^T\mathcal{M}_i P_{y,i}\mathrm{D}_i + N_{y,i}^T\mathcal{M}_i P_{tz,i}\mathrm{D}_{TAHU} +\right.$$

$$\left.0.5N_{z,i}^T\Psi_i + 0.5N_{y,i}^T\aleph_i - 0.5N_{y,i}^T\mathcal{M}_i\mathrm{r}_i\right] +$$

$$\mathrm{D}_{TAHU}^T\left[P_{ty,i}^T\mathcal{M}_i P_{ty,i}\right]\mathrm{D}_{TAHU} +$$

$$2\mathrm{D}_{TAHU}^T\left[P_{ty,i}^T\mathcal{M}_i F_{y,i}x_{0,i} + P_{ty,i}^T\mathcal{M}_i P_{y,i}\mathrm{D}_i + 0.5P_{tz,i}^T\Psi_i + 0.5P_{ty,i}^T\aleph_i - 0.5P_{ty,i}^T\mathcal{M}_i\mathrm{r}_i\right] +$$

$$\text{constants} \qquad (5.8)$$

#### 5.2.2.2 AHU Optimizer Cost Functions

To compute the optimum set-points for the AHU fan speed and discharge air temperature, the following economic cost functions are suggested at the AHU level, respectively:

$$J_{fan} = \sum_{j=0}^{N_p-1}\left(u_{fan}^T(k+j)\cdot\delta_{fan}(k+j) + \psi_{fan}^T(k+j)\cdot z_{fan}(k+j)\right) \qquad (5.9)$$

and

$$J_{T_{AHU}} = \sum_{j=0}^{N_p-1}\left(u_{T_{AHU}}^{T}(k+j)\cdot\delta_{T_{AHU}}(k+j) + \psi_{T_{AHU}}^{T}(k+j)\cdot z_{T_{AHU}}(k+j)\right) \tag{5.10}$$

where

$u_{fan} = P_{EDS}$ and $u_{T_{AHU}} = T_{AHU}$

$\delta_{fan}$ & $\delta_{T_{AHU}}$ are detailed in appendix D,

$\psi_{fan} = \gamma_{Room\_PEDS}$ and $\psi_{T_{AHU}} = \gamma_{Room\_TAHU}$, , and finally

$z_{fan} = P_{EDS}$ and $z_{T_{AHU}} = T_{AHU}$.

Both cost functions are quadratic because of the way the $\delta_{fan}$ and $\delta_{T_{AHU}}$ are defined. The first term in the cost function (5.9) is minimizing the fan power while the second term represents the sensitivities of the local zone costs for the $P_{EDS}$ set-point. Similarly, the first term in (5.10) is based on minimizing the cost of producing chilled water and the second term is the sensitivities that the local zone costs have for the AHU discharge air temperature set-point. All of the presented cost functions (zones and AHU costs) have the unit of ($).

#### 5.2.2.3 Zone Cost Function Sensitivities

From the real data of the VAV box inside the UBO building, the damper flow characteristic is found. All of the 9 controlled zones are assumed to have the same flow characteristic. Figure 5.13 shows the relations between the damper position (%), $P_{EDS}$ (in. water), and air flow (cfm) at the VAV box of each zone.

The $P_{EDS}$ value will be the same for all VAV boxes while the air flow will have different values. From the damper flow characteristic of the real UBO building VAV box, the value of $\partial cfm_i/\partial P_{EDS}$ can be computed as following: The air flow rate at VAV box $i$ is a function of the $P_{EDS}$ and the corresponding damper position $\varrho_i$, i.e. $cfm_i = f(P_{EDS}, \varrho_i)$, therefore, the changes in $cfm_i$ with respect to changes in $P_{EDS}$ and $\varrho_i$ can be written as:

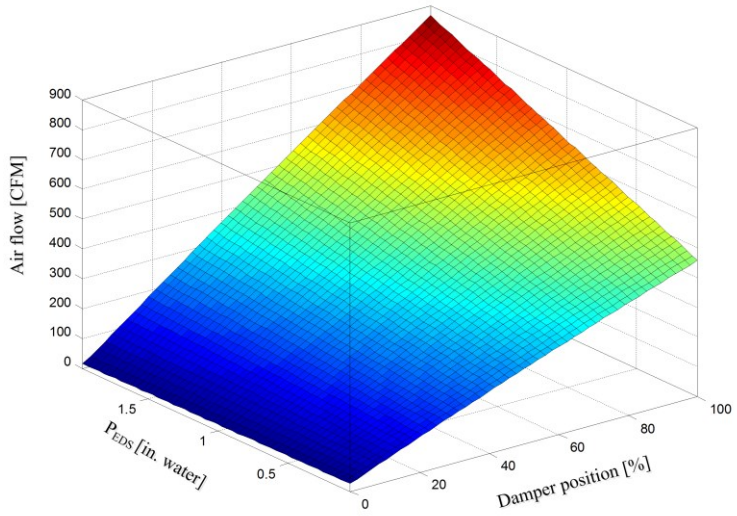



Figure 5.13: The VAV damper flow characteristic of the UBO building - reprinted with permission from [100]

$$dcfm_i = \frac{\partial f}{\partial P_{EDS}} dP_{EDS}\Big|_{\varrho_i = cons} + \frac{\partial f}{\partial \varrho_i} d\varrho_i\Big|_{P_{EDS} = cons}$$

and as we need the derivative of $cfm_i$ with $P_{EDS}$ when the damper position is constant, then $\frac{\partial f}{\partial \varrho_i} d\varrho_i = 0$, thus:

$$\frac{dcfm_i}{dP_{EDS}} = \frac{\partial f}{\partial P_{EDS}}\Big|_{\varrho_i = cons}$$

Using this partial derivative, the value of the zone cost function sensitivity with respect to $P_{EDS}$ can be computed as following: since we are interested in the sensitivity value of the zone cost function only when the damper is full opened, then we can write:

$$\frac{\partial \mathrm{J}_i}{\partial P_{EDS}} = \frac{\partial \mathrm{J}_i}{\partial u_i} \cdot \frac{\partial u_i}{\partial P_{EDS}}\Big|_{\varrho_i = 100\%}$$

where $u_i$ (which is $cfm_i$) is the first move in the computed optimum flow rate (control action) by the zone local MPC. This can be extended along the horizon; however, in this work we only considered the first applied local control action. For $\partial J_i / \partial u_i$ one can simply differentiate the cost (5.8) with respect to $u_i$. The identification algorithm used to produce the local zone dynamics is assuming that the input flow rate is fractional. That is the input flow rate is a fraction of the maximum flow arte specified inside the energyplus. Therefore, ($u_i = u_{actual\ flow} / u_{max\_flow\_EP}$) and this has to be taken care of in the numerical differentiation of $f(P_{EDS}, \varrho_i)$.

We are also interested in the zone cost sensitivity with respect to the AHU discharge air temperature when the damper is fully opened. This value can also be calculated by differentiating (5.8) with respect to $TAHU$. Once again we, in this work, are only considering the first value of the cost sensitivity with respect to $TAHU$ as with $P_{EDS}$.

### 5.2.3 Zone and AHU LC-DMPC Local Problems

This section introduces the LC-DMPC local problems and details the implementation of the LC-DMPC approach for the UBO building HVAC system.

#### 5.2.3.1 Zone Local LC-DMPC Problem

As it was shown in Figure 5.10, a zone subsystem has two local controllers: the lower PI controller and the LC-DMPC MPC. The design of the local MPC includes the room dynamics only. This MPC problem computes a sequence of optimum flow rate fractions from the inputs: desired room temperature, room actual temperature, upstream neighbor disturbing temperatures, downstream neighbor sensitivities, AHU discharge air temperature, and unmeasured outside disturbances. The first element of the optimum flow fractions is then fed into the dynamic of the local PI controller which gives the optimum temperature set-point of the room. Figure 5.14 shows the local zone MPC problem.

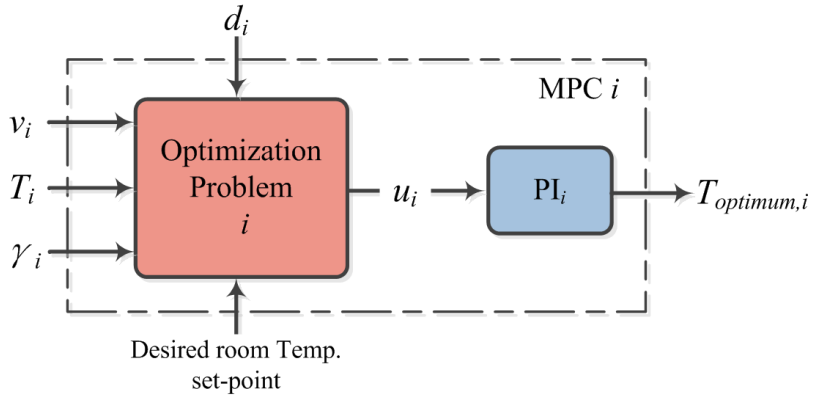



Figure 5.14: Zone level LC-DMPC controller

From Figure 5.14, the optimization problem $i$ is given as:

$$\min_{u_{i(j):j=0,\cdots N_p-1}} \left\{ \sum_{j=1}^{N_p} [e_i^T(k+j) \cdot \mathbb{Z}_i(k+j)] + \sum_{j=0}^{N_p-1} \psi_i^T(k+j) \cdot z_i(k+j) \right\} \tag{5.11}$$

subject to:

$$x_i(k+1) = A_i x_i(k) + B_{u,i} u_i(k) + B_{v,i} v_i(k) + B_{t,i} T_{AHU}(k) + B_{d,i} d_i(k)$$

$$y_i(k) = C_{y,i} x_i(k)$$

$$z_i(k) = C_{z,i} x_i(k)$$

$$0.07 \leq u_i(k+j) \leq u_{max}(k), \qquad j = 0, \cdots N_p - 1$$

and from the $\mathrm{PI}_i$ controller dynamics, the optimum applied temperature set-point is computed using the first move from the optimization problem (5.11).

In (5.11) the outside temperature, relative humidity, and AHU discharge air temperature values are assumed to be constant along the horizon $N_p$ and are updated periodically at each sampling. The value of the constraint $u_{max}(m)$ changes per the sampling $k$ as the $P_{EDS}$ changes and is computed as:

$$u_{max}(k) = f(P_{EDS}(k), \varrho_i)/u_{max\_flow\_EP}$$

where $\varrho_i$ is kept at 100% (i.e. the maximum flow rate when the damper is fully open) and $u_{max\_flow\_EP}$ is specified to be $0.5\, m^3/sec$ inside the EP for all zones.
Finally, a flow rate of 5 cfm (corresponds to 0.07) is imposed as the minimum limit on the control action in order to satisfy the natural ventilation requirement.

#### 5.2.3.2 AHU Local Optimizer LC-DMPC Problem

The LC-DMPC optimization problem for computing the AH set-points is illustrated in Figure 5.15.

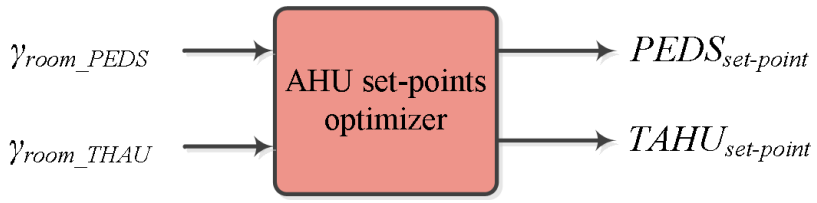



Figure 5.15: The AHU set-points optimizer

As the LC-DMPC controller injects only the first control action into the corresponding zone dynamics, we are only computing the AHU set-points to the changes in zone cost sensitivities with respect to the change in the applied flow rate fraction. Thus, the AHU optimization problem for one time step is given as:

$$\min_{u_{fan},\ u_{T_{AHU}}} \left[\left(u_{fan}^T \cdot H_{fan} \cdot u_{fan} + f_{fan}(k) \cdot u_{fan}\right) + u_{T_{AHU}}^T \cdot H_{T_{AHU}} \cdot u_{T_{AHU}} + f_{T_{AHU}}(k) \cdot u_{T_{AHU}}\right]$$

or

$$\min_{u_{fan},u_{T_{AHU}}} \left\{\begin{pmatrix} u_{fan}^T & u_{T_{AHU}}^T \end{pmatrix} \begin{bmatrix} H_{fan} & 0 \\ 0 & H_{T_{AHU}} \end{bmatrix} \begin{pmatrix} u_{fan} \\ u_{T_{AHU}} \end{pmatrix} + \begin{pmatrix} f_{fan}(k) & f_{T_{AHU}}(k) \end{pmatrix} \begin{pmatrix} u_{fan} \\ u_{T_{AHU}} \end{pmatrix}\right\}$$

subject to:

$$\begin{bmatrix} 0.2 \\ 14.45 \end{bmatrix} \leq \begin{pmatrix} u_{fan} \\ u_{T_{AHU}} \end{pmatrix} \leq \begin{bmatrix} 1.8 \\ 17.78 \end{bmatrix} \tag{5.12}$$

where $f_{fan}(k)$ and $f_{T_{AHU}}(k)$ are detailed in appendix D.

In problem (5.12), the constraint values for the control moves are taken for the normal operation of the real UBO building. We also assumed that the lower level PI controllers at the AHU level are perfect and fast such that at the next sampling time the discharge air temperature and fan speed hit the computed set-points, i.e. there is one time delay between the AHU outputs and computed set-points.

As the values of $P_{ESD}$ and $T_{AHU}$ are constant for all zones at instance $k$ and each zone has different cost sensitivities, the question now is which values for $\psi_{fan}^{T}$ and $\psi_{T_{AHU}}^{T}$ should be selected by the AHU optimizer? To answer this question, we need to know $\partial J_i/\partial P_{EDS}$ and $\partial J_i/\partial T_{AHU}$. When a zone has a fully opened damper, it means that the damper is no longer regulating the temperature inside the room. At this point, the room asks the AHU to improve its zone level cost performance throughout the changes in the fan speed and/or the discharge air temperature. The $\partial J_i/\partial P_{EDS}$ must have a negative value which means that any increasing in the fan speed (or the $P_{EDS}$) will result in decreasing in the cost value of $J_i$. On the other hand, the value of $\partial J_i/\partial T_{AHU}$ should be positive, i.e. the zone asks for inlet air with cooler temperature which reduces its local cost value. Based on above we propose that the AHU optimizer selects the values for $\psi_{fan}^{T}$ and $\psi_{T_{AHU}}^{T}$ as following:

$$\psi_{fan}^{T} = \min_{i=1,2,\cdots,10}(\partial J_i/\partial P_{EDS})$$

$$\psi_{T_{AHU}}^{T} = \max_{i=1,2,\cdots,10}(\partial J_i/\partial T_{AHU})$$

#### 5.2.3.3 The Main LC-DMPC Algorithm for the UBO Building

In this section, the main control algorithm for controlling the HVAC system in the UBO building is stated. This control algorithm is based on the optimum LC-DMPC algorithm proposed in chapter 2 (Algorithm 2.1).

**Algorithm 5.1: The main UBO building control**

**Input:** Number of iterations $N_a$ and initial values for states $x_{0,i}$ and error covariance $\mathfrak{H}_i$.

**Set:** initial values for $V_i = 0,\ U_i = 0,\ \Psi_i = 0$,or take previous values from last step and measure $T_{out}(k)$, $RH_{out}(k)$, $P_{EDS}(k)$, and $T_{AHU}(k)$.

**Start of Iteration:** For $j$ = 1 to $N_a$ Do:

**Step 1:** Exchange current information with local agents:

$$\boldsymbol{\Psi}(j+1) = \left[\Psi_1(j+1), \cdots, \Psi_p(j+1)\right]^T = \boldsymbol{\Gamma}^T\left[\gamma_1(j), \cdots, \gamma_p(j)\right]^T = \boldsymbol{\Gamma}^T\boldsymbol{\gamma}(j)$$

$$\mathbf{V}(j+1) = \left[\mathrm{V}_1(j+1), \cdots, \mathrm{V}_p(j+1)\right]^T = \boldsymbol{\Gamma}\left[\mathrm{Z}_1(j), \cdots, \mathrm{Z}_p(j)\right]^{\boldsymbol{T}} = \boldsymbol{\Gamma}\mathbf{Z}(j)$$

**Step 2**: Solve the optimization problem given in (5.11):

$$\mathrm{U}_i{}^{QP}(j) \leftarrow \text{argmin Problem}(5.11)$$

**Step 3**: For $\beta \in [0,1)$, set:

$$\mathrm{U}_i(j+1) \leftarrow \beta U_i(j) + (1-\beta)\mathrm{U}_i{}^{QP}(j)$$

**Step 4**: Update local output disturbance:

$$\mathrm{Z}_i(j+1) \leftarrow F_{z,i}x_{0,i}(k) + M_{z,i}\mathrm{U}_i(j+1) + N_{z,i}\mathrm{V}_i(j) + P_{y,i}\mathrm{D}_i(k) + P_{tz,i}\mathrm{D}_{TAHU}(k)$$

**Step 5**: Update sensitivity for input disturbance. Set:

$$\gamma_i(j+1) \leftarrow -N_{y,i}^T\mathcal{M}_i\mathrm{r}_i(k) + 2N_{y,i}^T\mathcal{M}_iM_{y,i}\mathrm{U}_i(j+1) + 2N_{y,i}^T\mathcal{M}_iN_{y,i}\mathrm{V}_i(j) +$$
$$2N_{y,i}^T\mathcal{M}_iP_{y,i}\mathrm{D}_i(k) + 2N_{y,i}^T\mathcal{M}_iP_{ty,i}\mathrm{D}_{TAHU}(k) + N_{z,i}^T\Psi_i(j) + 2N_{y,i}^T\mathcal{M}_iF_{y,i}x_{0,i}(k) + N_{y,i}^T\aleph_i$$

Next $j$

**Step 6:** Compute the zone sensitivities for $P_{EDS}$ and $T_{AHU}$ as:

For $\forall\ \Sigma_i \in p$:

If $\varrho_i = 100\%$ then:

$$\gamma_{room,i_{P_{EDS}}}(k) = \partial\mathrm{J}_i/\partial P_{EDS}(k) = \partial\mathrm{J}_i/\partial u_i \cdot \partial u_i/\partial P_{EDS}(k)|_{\varrho_i=100\%}$$

$$\gamma_{room,i_{T_{AHU}}}(k) = \partial\mathrm{J}_i/\partial T_{AHU}(k)$$

Otherwise

$$\gamma_{room,i_{P_{EDS}}}(k) = 0,\ \gamma_{room,i_{T_{AHU}}}(k) = 0$$

**Step 7:** With the values of:

$$\psi_{fan}^{T} = \min_{i=1,2,\cdots,9}\left(\gamma_{room,i\_P_{EDS}}\right), \qquad \psi_{T_{AHU}}^{T} = \max_{i=1,2,\cdots,9}\left(\gamma_{room,i\_T_{AHU}}\right)$$

Solve the AHU optimizer problem given in (5.12) and set:

$$\begin{Bmatrix} P_{EDS}(k) \\ T_{AHU}(k) \end{Bmatrix} = \text{argmin Problem}(5.12)$$

**Output:** $U_i$ (first value), $P_{EDS}(k)$, and $T_{AHU}(k)$ (Inject into EnergyPlus).

**Get:** new measurements for $T_{out}$, $RH_{out}$, $T_i$, $u_i$, and $v_i$ and estimate current room state.

**Update:** the flow rate constraint for the local MPCs:

$$u_{max}(k+1) = f(P_{EDS}(k),100\%)/u_{\max\_flow\_EP}$$

**Go to: Start of Iteration**.

Figure 5.16 shows the implementation of the Algorithm 5.1 with all communication details between Matlab and EnergyPlus through MLEP.

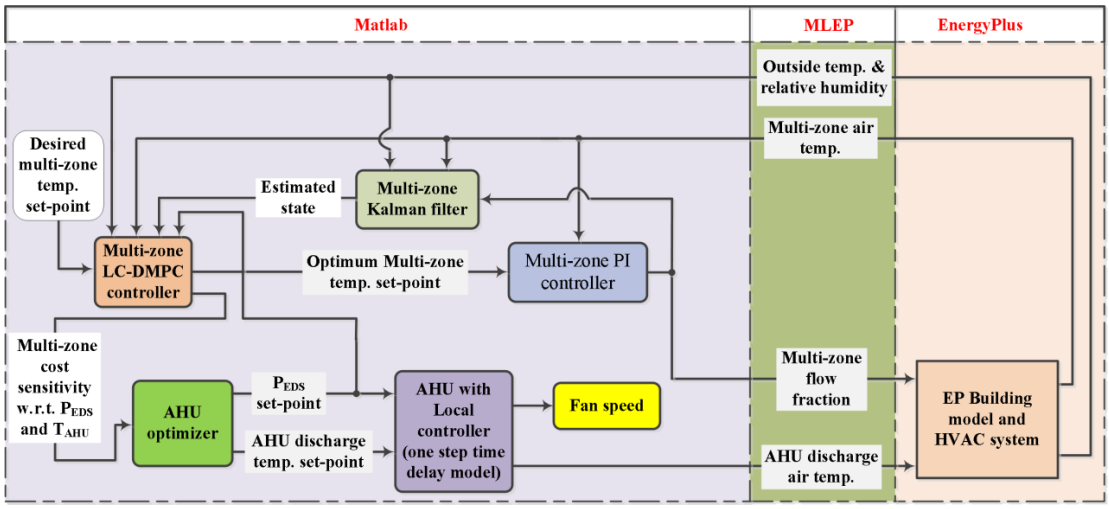



Figure 5.16: The implementation of the Algorithm 5.1

## 5.3 Simulation Results

The UBO building is modeled inside the EnergyPlus while Algorithm 5.1 is computed in Matlab to regulate the air temperature inside the zones and calculate the set-points for the end static pressure and discharge air temperature of the AHU. The following data is used for the simulations:

For all zone subsystems, the following values represent a typical office activity [101]: Metabolic rate (sedentary activity): $M = 70\,\mathrm{W}/m^2$, effective mechanical power: $W = 0\,\mathrm{W}/m^2$, clothing insulation (light business suit): $I_{cl} = 0.1156\,m^2\mathrm{K/W}$, relative air velocity (typical for office): $v_{air} = 0.1\,m/s$, clothing surface temperature initial value: $T_{cl,0} = 20\ C^0$, mean radiant temperature initial value: $\bar{T}_{r,0} = 20\ C^0$, and for relative humidity of 50% (for typical comfort level) the following are set as $k_{pa} = 84.21, q_{pa} = -515.79$. The last four values are required by the equations presented in [101].

For the AHU optimizer, all used data is given in appendix D and for the cost of calculating the $P_{EDS}$ set-point, the following equation is derived using the damper flow characteristic for all zones to relate the total flow-rate (cfm) and $P_{EDS}$ set-point:

$$flow\ rate_{total}(\mathrm{cfm}) = 5149 \cdot P_{EDS}(\mathrm{in.\ water}) - 979.8$$

General parameters for simulation:

Sampling time: $t_s = 5\ min$, prediction horizon: $N_p = 8\ (40\ \mathrm{min.})$, convex combined scalar: $\beta = 0.1$, and number of iterations per sampling: $N_a = 5$. The last two values are required by the structure of the LC-DMPC approach.

For the zone low level PI controllers, the gains are varying between the limits given as:

The proportional gains: $0.06 \leq k_{p,i} \leq 1.1$, and the integral time constant: $600 \leq T_{I,i} \leq 3000$ for $i = 1,2,\cdots,10$, where the unit for $k_{p,i}$ is $flow\ fraction/℃$ and for $T_{I,i}$ is $flow\ fraction/(sec \cdot ℃)$, respectively.

To verify the application of Algorithm 5.1 for controlling the UBO HVAC system, four different cases are simulated. All cases have same environmental conditions and cost function variables. However, they are different in the priorities that have been assigned for the subsystems or optimizers. In the first case room 8 is represented to be the most important subsystem. Therefore, more weight is assigned for the room in the comfort cost variable. In the second case, a higher priority is assigned for room 9. For the third cases, we consider the cost of computing the $P_{EDS}$ set-point at the AHU level to have the highest priority. As the last case, less priority has been given for the zone cost functions comparing with AHU optimizer costs. In the first two cases, which are the normal operating daily cases, at the AHU optimizer the cost of running the fan for five minutes (i.e. during the sampling time) is less than the cost of producing chilled water for the same time period. And this is why we have not considered the case where the cost of computing the AHU discharge air temperature set-point has the largest priority over the $P_{EDS}$ set-point. Table 5.1 shows the weighting costs associated for each of the four simulation cases.

All simulation groups are run from July the 11$^{th}$ to the 13$^{th}$ of the same month and four of the ten zones (zones 2, 7, 8, and 9) are selected as representative behavior for all of the simulated zones.

**Table 5.1: Cost Associated with Each Simulation Case**

| Subsystem | Case 1 | Case 2 | Case 3 | Case 4 | Cost |
|---|---|---|---|---|---|
| Zone 1 | 0.1603 | 0.1603 | 0.1603 | 0.0020 | $ |
| Zone 2 | 0.2003 | 0.2003 | 0.2003 | 0.0020 | $ |
| Zone 3 | 0.2003 | 0.2003 | 0.2003 | 0.0020 | $ |
| Zone 4 | 0.2003 | 0.4006 | 0.2003 | 0.0020 | $ |
| Zone 5 | 0.2003 | 0.2003 | 0.2003 | 0.0020 | $ |
| Zone 6 | 0.2003 | 0.2003 | 0.2003 | 0.0020 | $ |
| Zone 7 | 0.2003 | 0.2003 | 0.2003 | 0.0020 | $ |
| Zone 8 | 0.4006 | 0.2003 | 0.4006 | 0.0020 | $ |
| Zone 9 | 0.3005 | 0.6010 | 0.3005 | 0.0020 | $ |
| $P_{EDS}$ optimizer | 0.087 | 0.087 | 2.000 | 0.0870 | $\$/(k\mathrm{W}.hr)$ |
| $T_{AHU}$ optimizer | 0.0655 | 0.0655 | 0.0136 | 0.0655 | $\$/(\mathrm{mmBTU})$ |

In table 5.1, the cost associated with the zones is the productivity cost of the occupancy. The desired tracking references for the rooms (specified by the user) are assumed to be the same for all the rooms. This desired reference changes between 22.22 ℃ (the cooling mode) and 29.44 ℃ (the non-cooling mode). During a business day, the cooling starts at 7 am and continues until 7 pm. We also assumed that the building occupancy starts at 8 am and stay constant until 6 pm for a normal business day. In addition to occupancy loads, there are lighting and machine scheduled heating loads as well.

### 5.3.1 Case 1: When Room 8 Has the Higher Priority

Figures 5.17 through 5.20 show the temperature and applied air flow rate responses of the pre-selected zones when higher priority is assigned for room 8. For this case, the optimum temperature that the local MPCs try to approach is 22.4 ℃ which is the

average of the desired reference 22.22 ℃ and the comfort optimum temperature 22.58 ℃. This average temperature is shown with green lines in the figures.

During the cooling mode, the local MPC regulates the damper position whenever the room temperature is around the optimum average temperature. This can be seen at the beginning of the cooling period. Otherwise it fully opens the damper when the outside temperature is high and the room temperature is above the average temperature. When the desired temperature changes to 29.44 ℃, the optimum average temperature changes to around 26 ℃ and the local MPCs attempt to adjust the room temperature around this new reference. This is why the room temperature starts to oscillate. However, as we are only considering cooling in the simulations, the only way for the local MPC to regulate the temperature around the new reference is to bring down the damper position to its minimum limit. On the other hand, at the AHU level, from Figures 5.21 and 5.22 we can see that the $P_{EDS}$ set-point starts to increase at the beginning of the day (or beginning of cooling) and stays approximately constant at 1.8 in. of water for the rest of cooling mode. At the same time, the AHU discharge air temperature set-point will be constant at its minimum (around 17.8 ℃) and starts decreasing when the outside temperature increases where the rooms demand for more cooling loads. The AHU chilled water valve response is shown in Figure 5.23. Figure 5.24 gives the instant operating dollar cost of running the HVAC system with Algorithm 5.1. In this cost we only considered the cooling portion of the zones (i.e. when the target temperature is 22.4 ℃ ).

The actual air flow rate that enters a room can be computed simply by multiplying the air flow fraction by the maximum air flow rate that is specified inside the Energy Plus

(1059.44 $ft^3/min$). On the other hand, a damper position is defined to be fully open in the resulted plots when the maximum air flow fraction is reached at that sampling time. For current simulations the maximum air flow fraction is 0.7824 when $P_{EDS_{max}} = 1.8$ in. water.

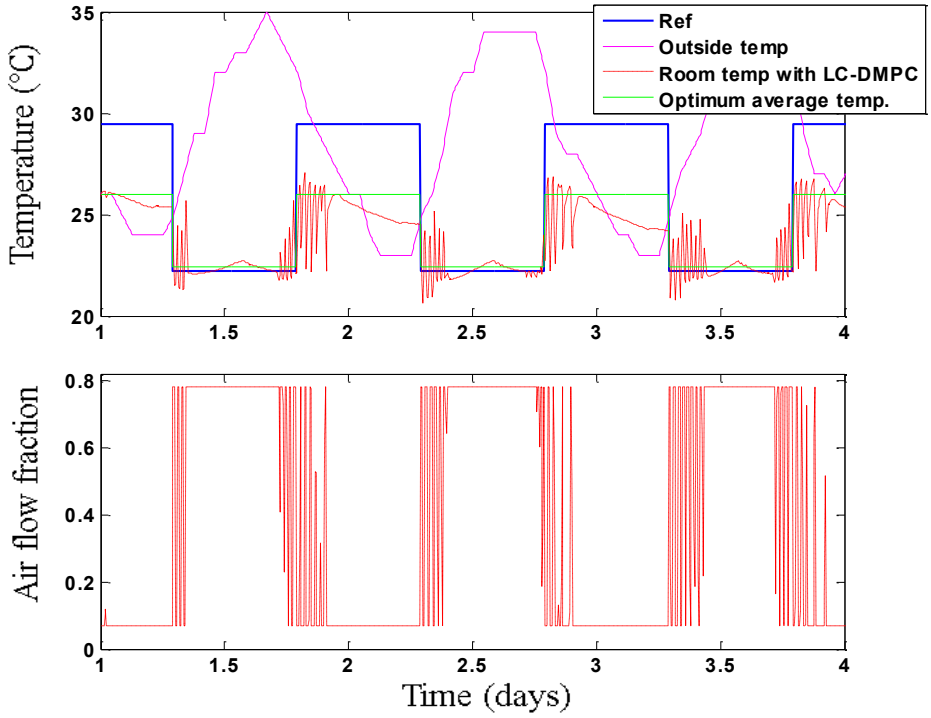



Figure 5.17: Temperature and air flow fraction for room 2 - case 1

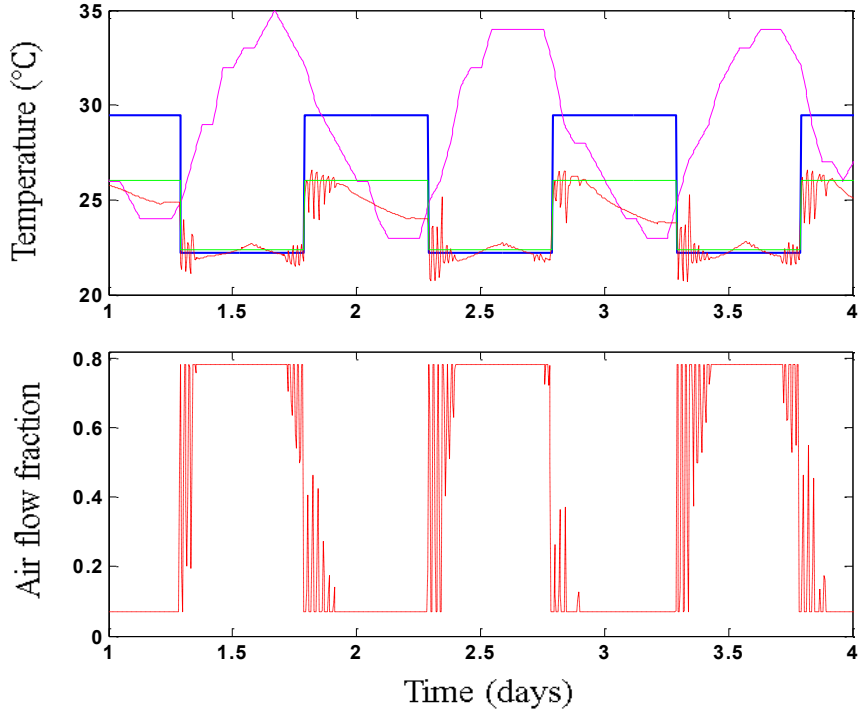



Figure 5.18: Temperature and air flow fraction for room 7 - case 1

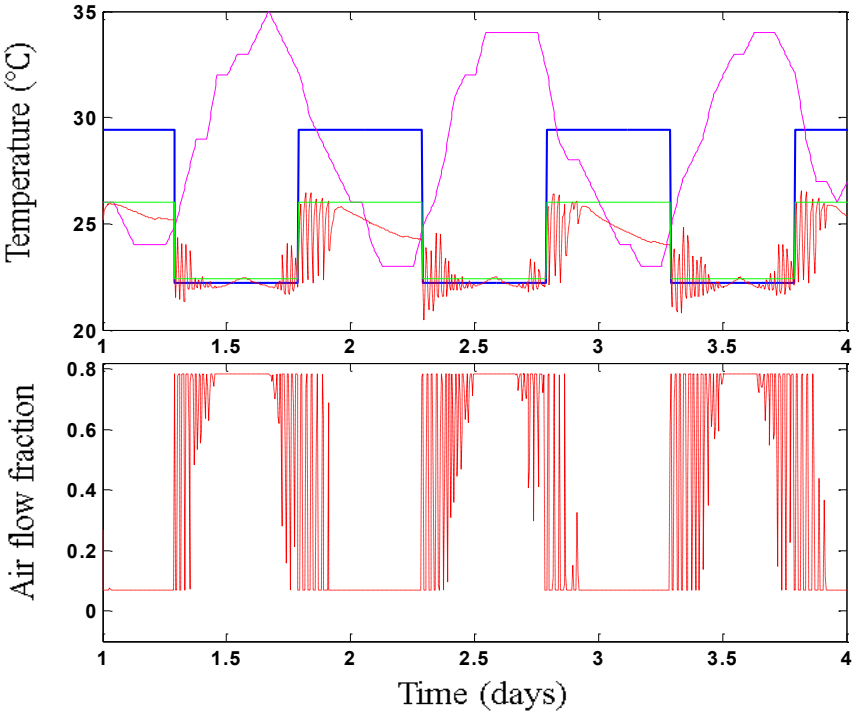



Figure 5.19: Temperature and applied air for room 8 - case 1

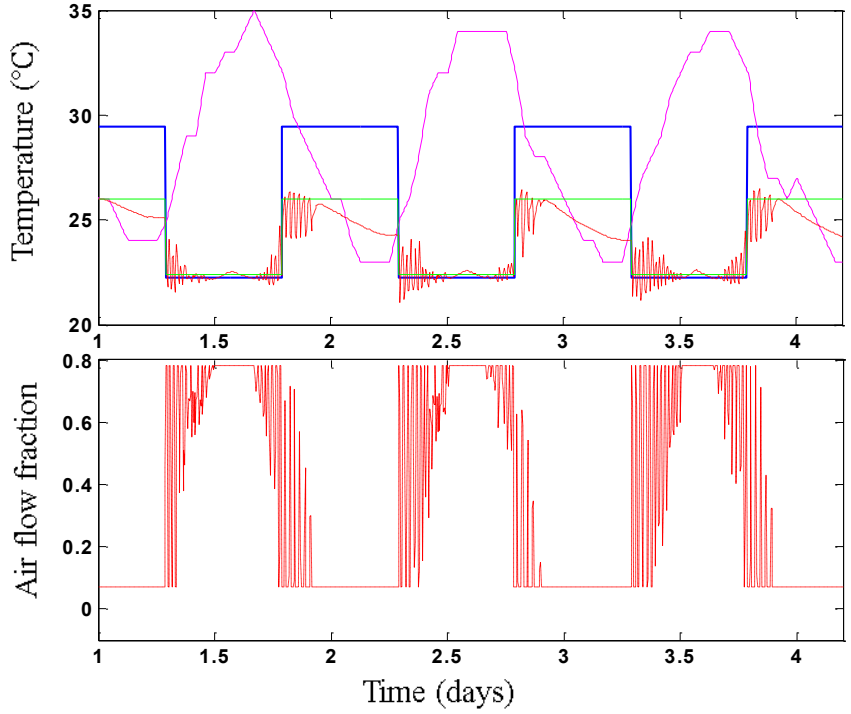



Figure 5.20: Temperature and air flow fraction for room 9 - case 1

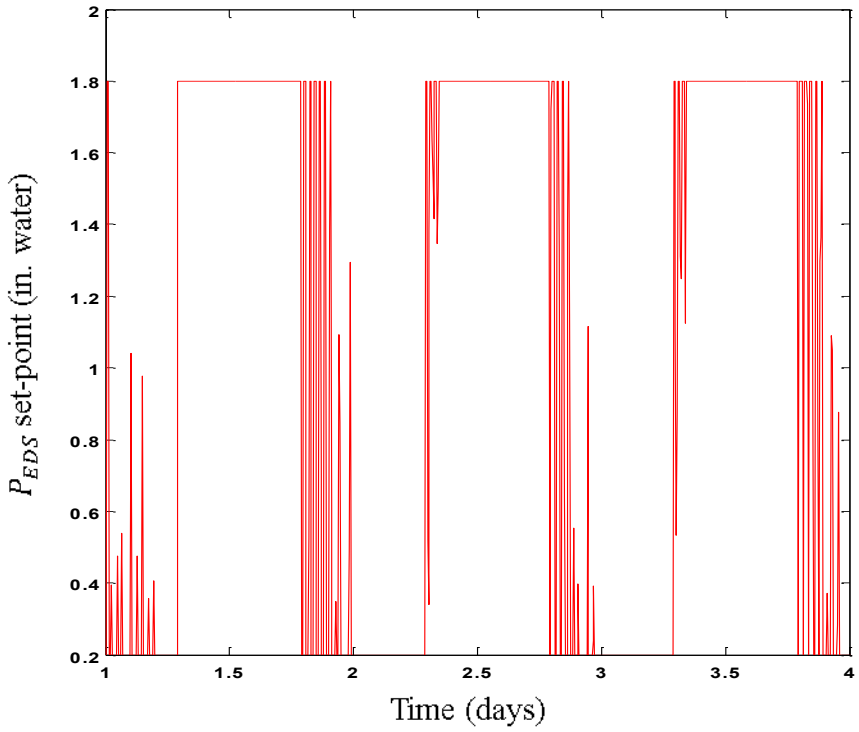



Figure 5.21: The $P_{EDS}$ set-point - case 1

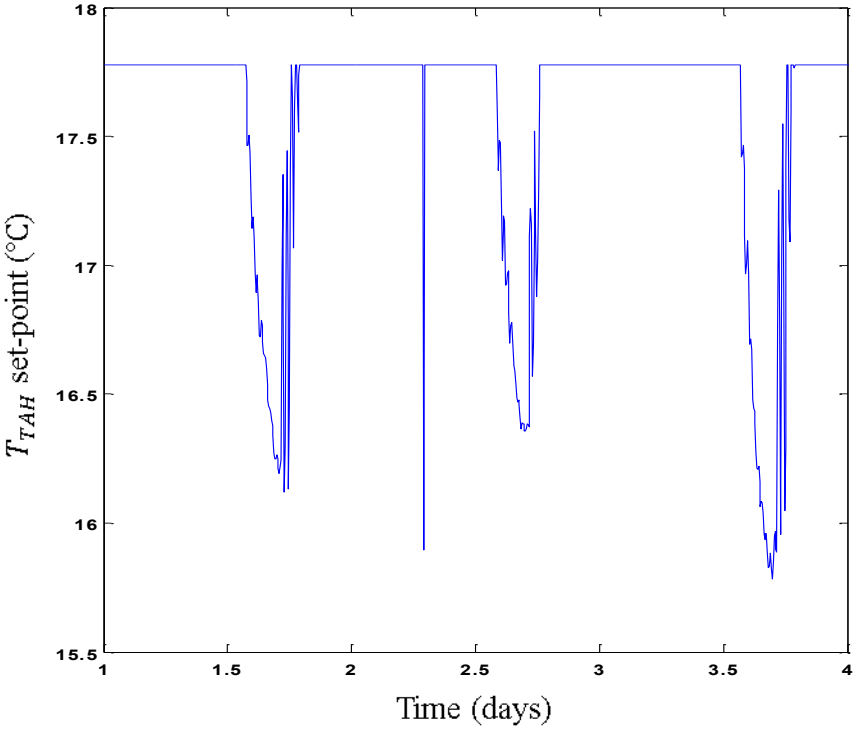



Figure 5.22: The $T_{AHU}$ set-point - case 1

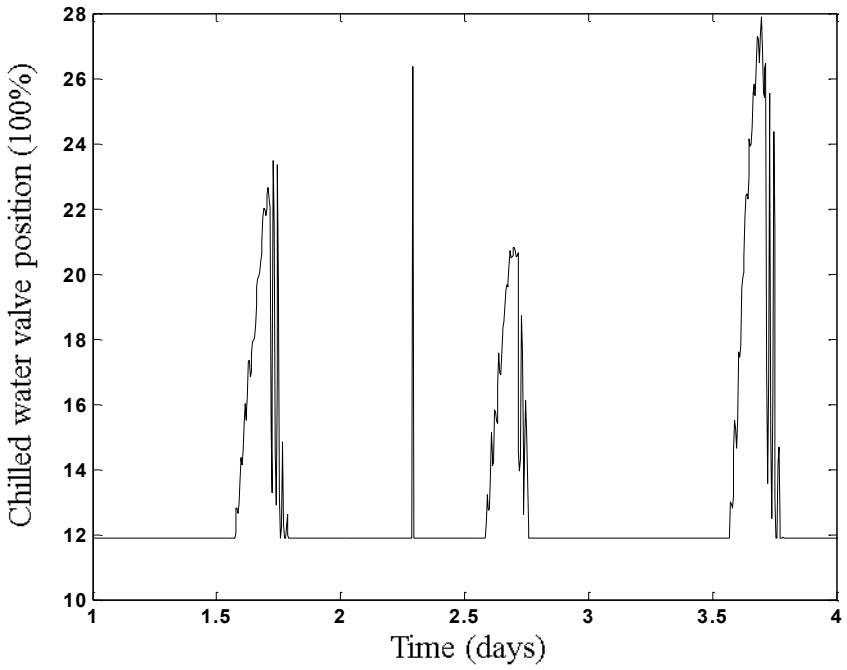



Figure 5.23: AHU chilled water valve response - case 1

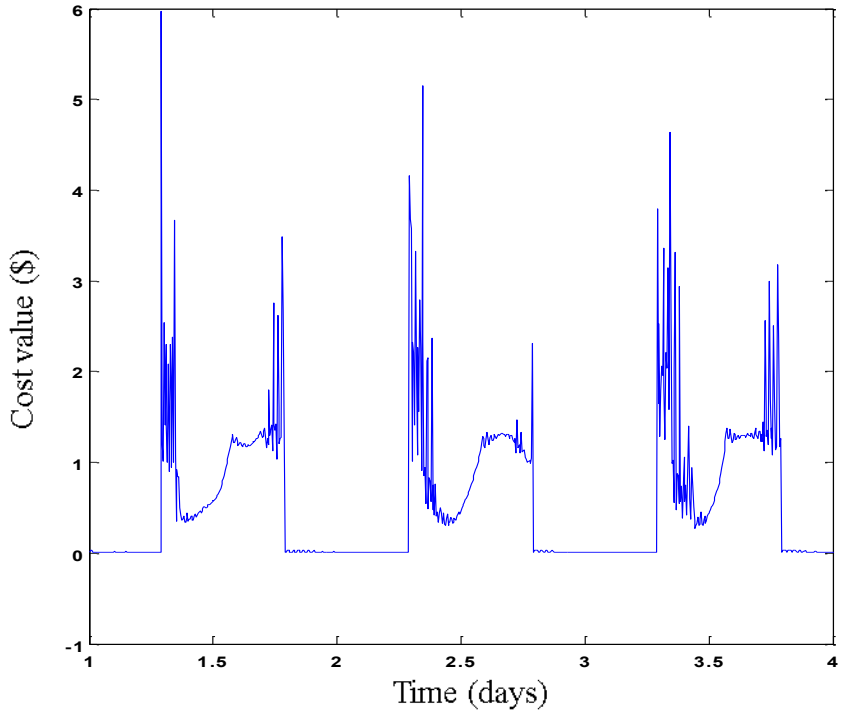



Figure 5.24: Instant operating cost of the UBO building HVAC system - case 1

### 5.3.2 Case 2: When Room 9 Has the Higher Priority

The main reasons for simulating this case are to show that Algorithm 5.1 gives the ability to assign different priorities for the subsystems and to show that the zone with high priority dominates the behavior of the AHU through the cost sensitivities. Figures 5.25 through 5.29 show room 9 temperature and air flow fraction responses, $P_{EDS}$ and $T_{AHU}$ set-points, AHU chilled water valve response, and instant total operation cost, respectively.

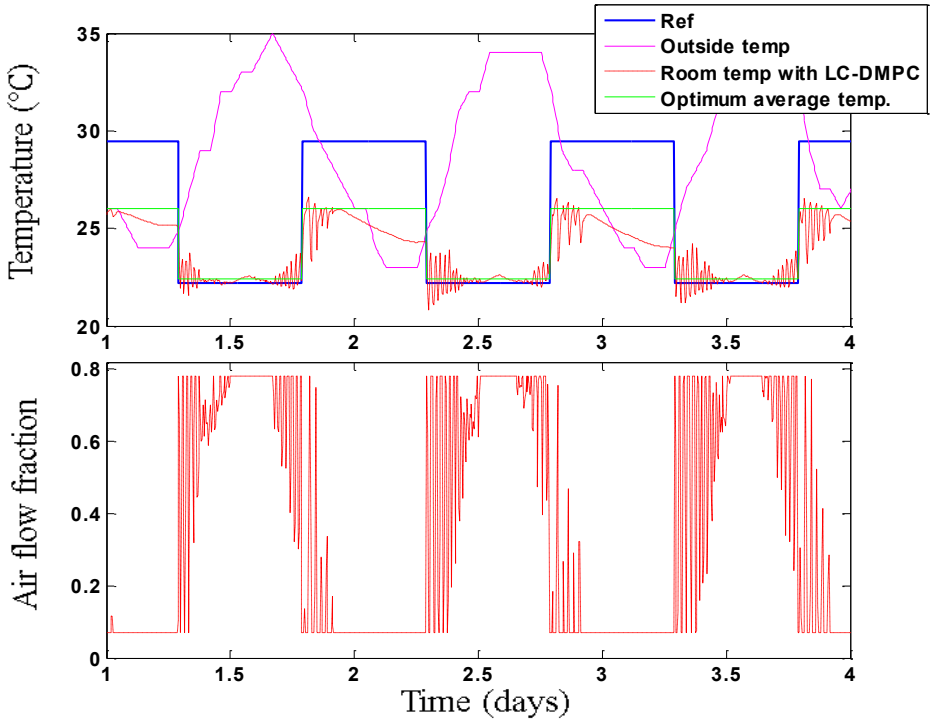



Figure 5.25: Temperature and air flow fraction for room 9 - case 2

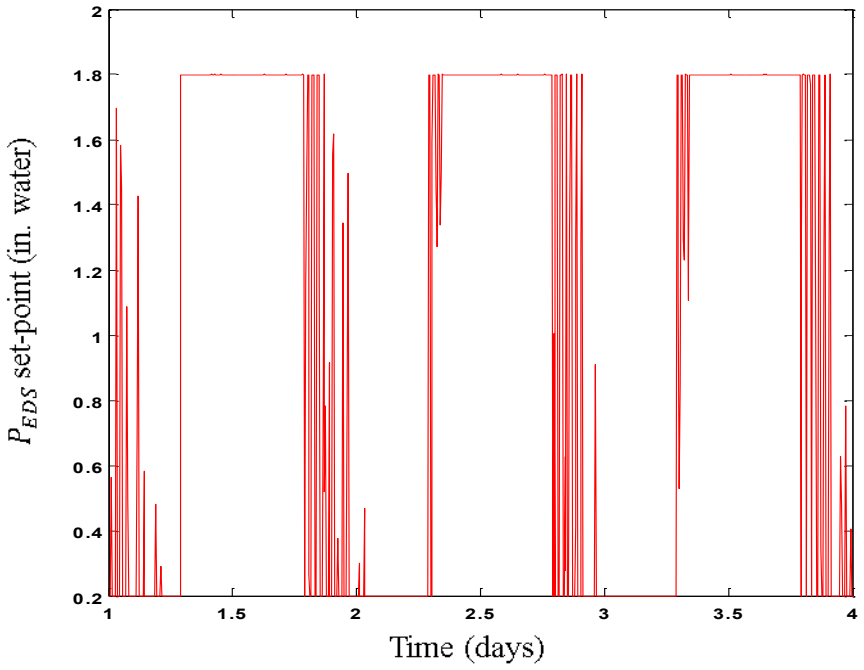



Figure 5.26: The $P_{EDS}$ set-point - case 2

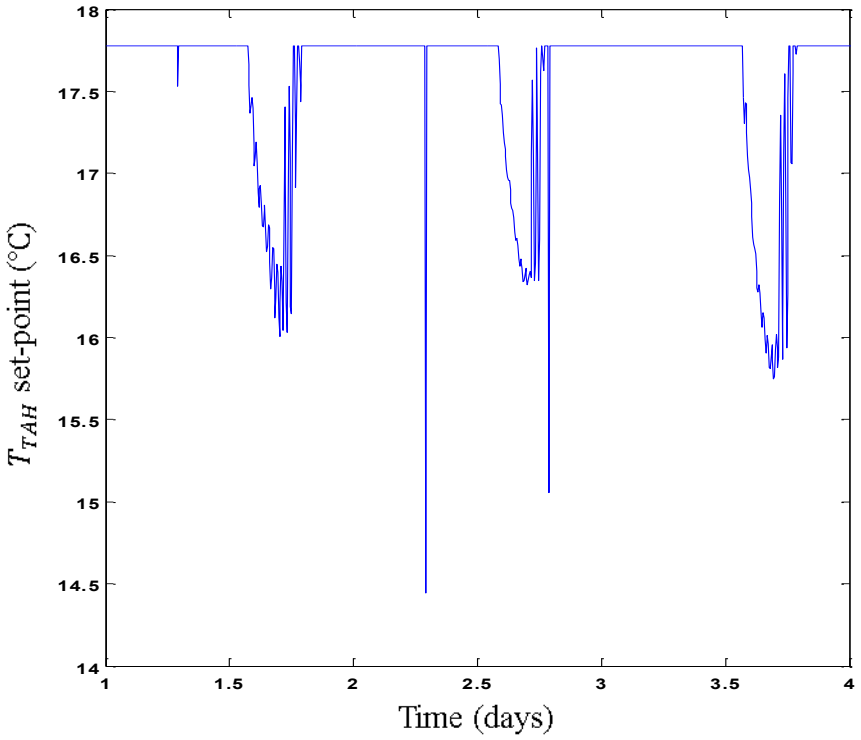



Figure 5.27: The $T_{AHU}$ set-point - case 2

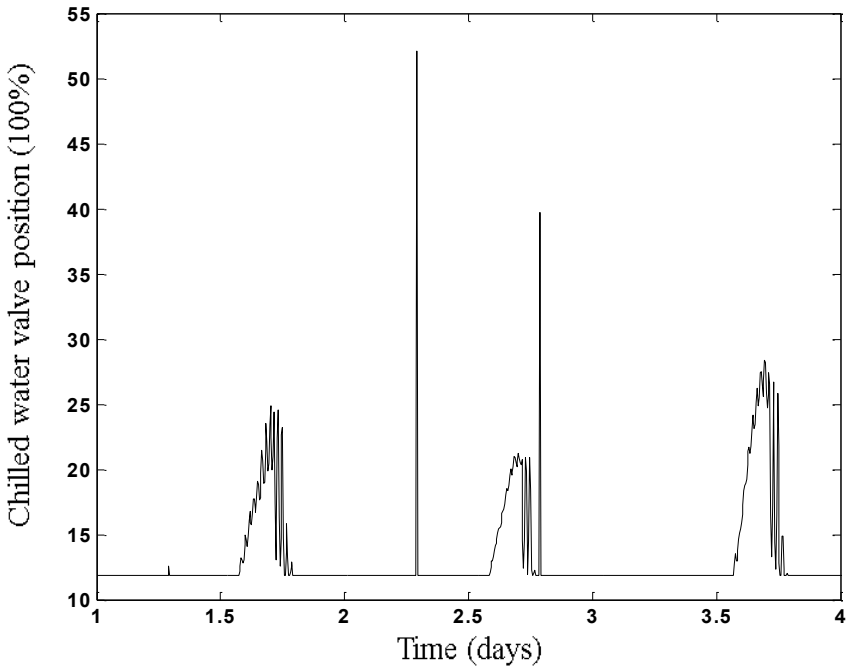



Figure 5.28: AHU chilled water valve response - case 2

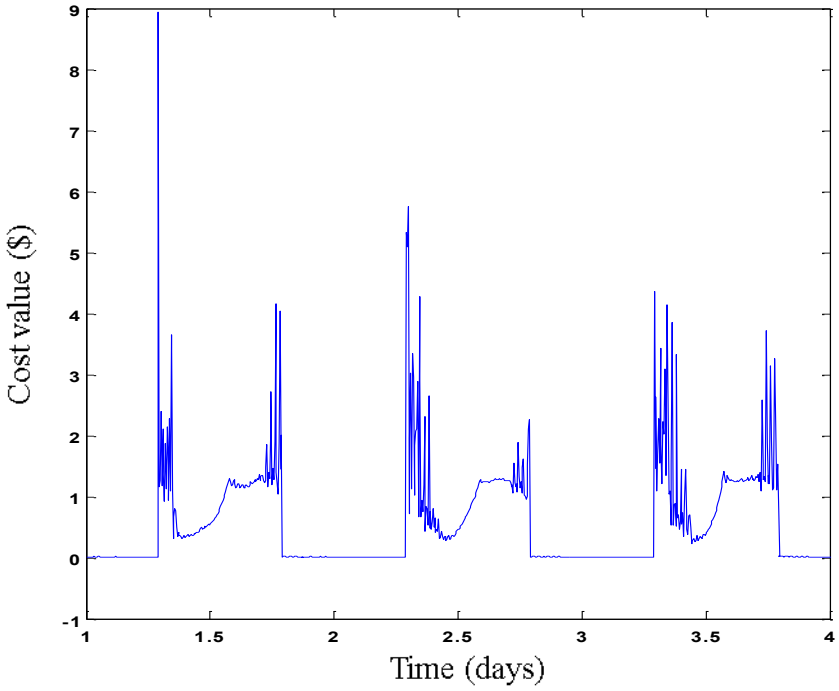


Figure 5.29: Instant operating cost of the UBO building HVAC system - case 2

### 5.3.3 Case 3: When AHU Fan Has Higher Energy Cost

For this case, when the fan energy cost is higher than the cost of producing chilled water, the AHU optimizer will try to change the discharge air temperature set-point and keep the $P_{EDS}$ set-point as close as to the minimum value (0.2 in. water). However, because the modified room temperature models approximate the actual dynamics between zones and the AHU discharge air temperature as a linear relation (the actual relation is bilinear), this causes the oscillation of the zone temperatures as the set-point of the AHU discharge temperature changes rapidly. Room temperature dynamics are sensitive for changes in AHU discharge temperature. Figures 5.30 and 5.31 give the $P_{EDS}$ and $T_{AHU}$ set-points while Figure 5.32 show the room 9 temperature and air flow fraction responses, respectively.

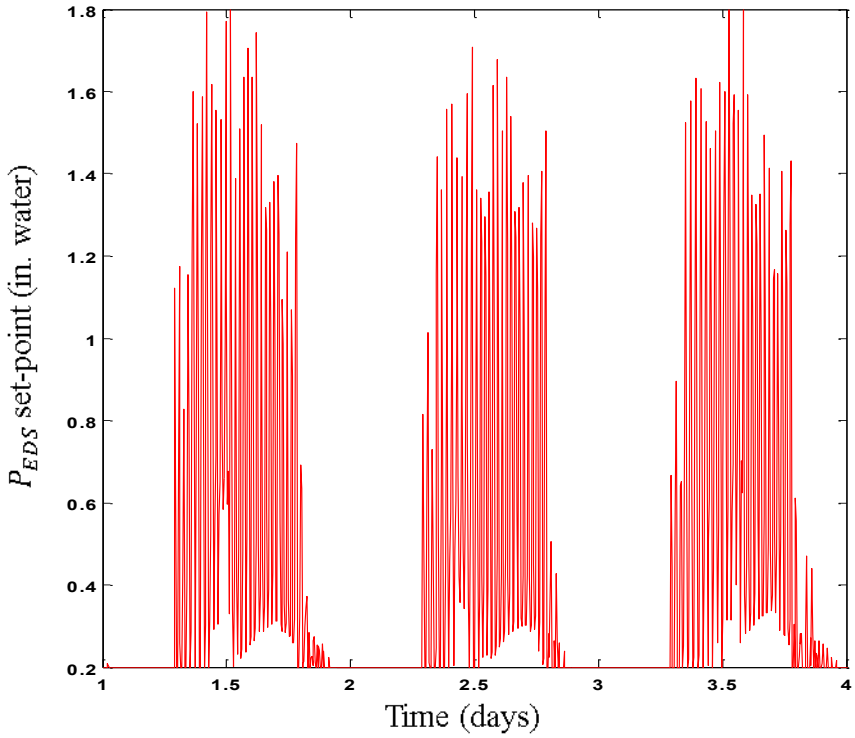



Figure 5.30: The $P_{EDS}$ set-point - case 3

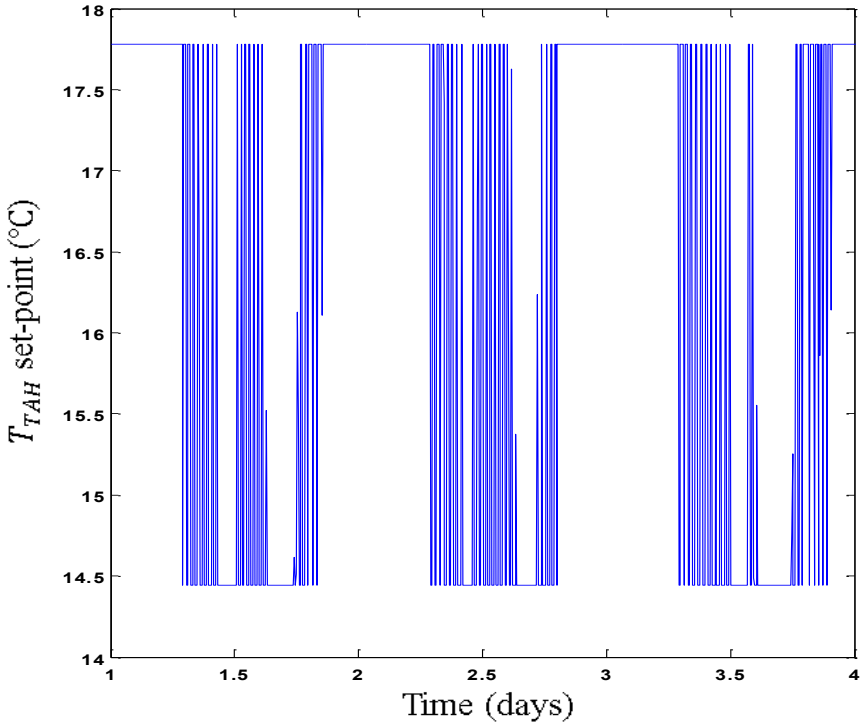



Figure 5.31: The $T_{AHU}$ set-point - case 3

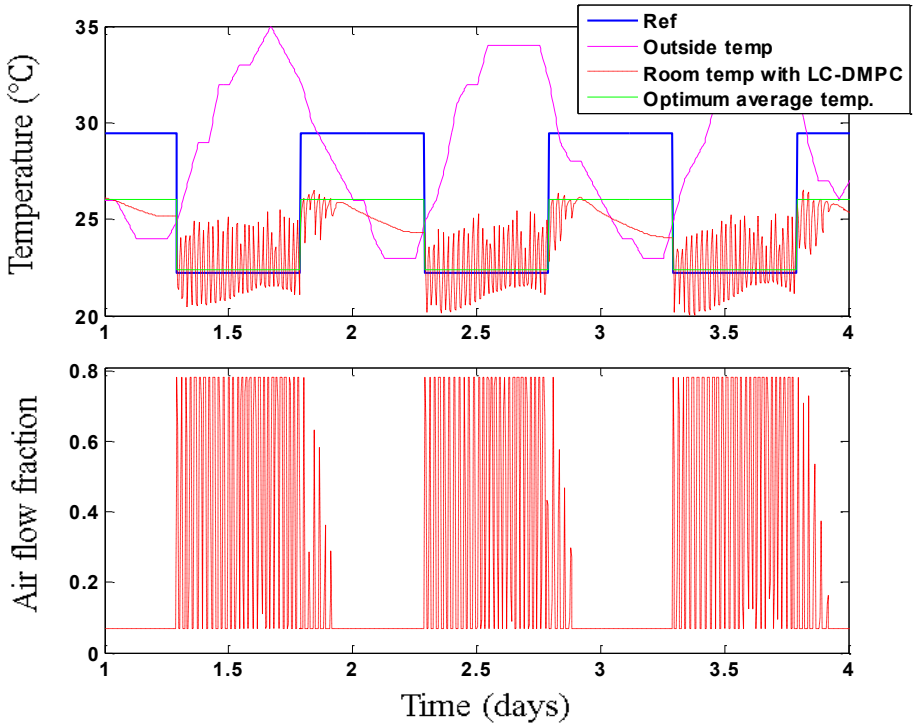



Figure 5.32: Temperature and air flow fraction for room 9 - case 3

### 5.3.4 Case 4: When the Zones Have the Less Priority

In this case the productivity costs for the zones are set to be much lower than the costs of running the fan and producing chilled water at the AHU level. We can see that AHU optimizer is trying to keep the discharge air temperature at its minimum value (around 17.7 ℃) and changing the $P_{EDS}$ between the minimum and a value of 0.53 in. of water. This is because the cost of producing chilled water for five minutes is greater than running the fan for same time period (the normal daily operations as in cases 1 and 2). Also, the zone demands are small such that they can be regulated by the inlet flow rate only with the minimum AHU discharge air temperature. Once again Figures 5.33

through 5.35 show the $P_{EDS}$ and $T_{AHU}$ set-points and room 9 responses for this case, respectively.

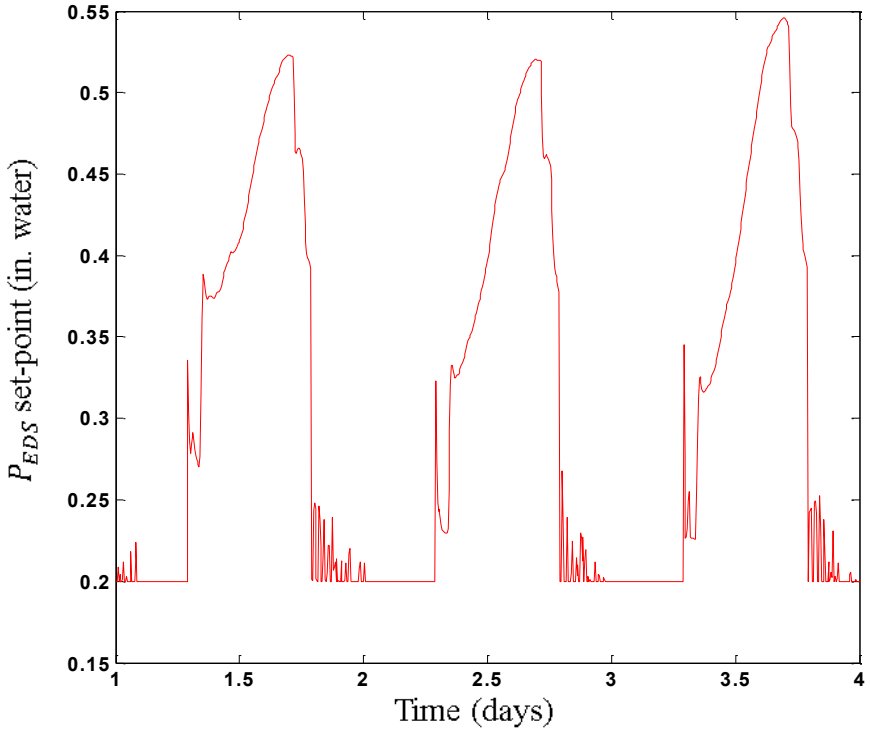



Figure 5.33: The $P_{EDS}$ set-point - case 4

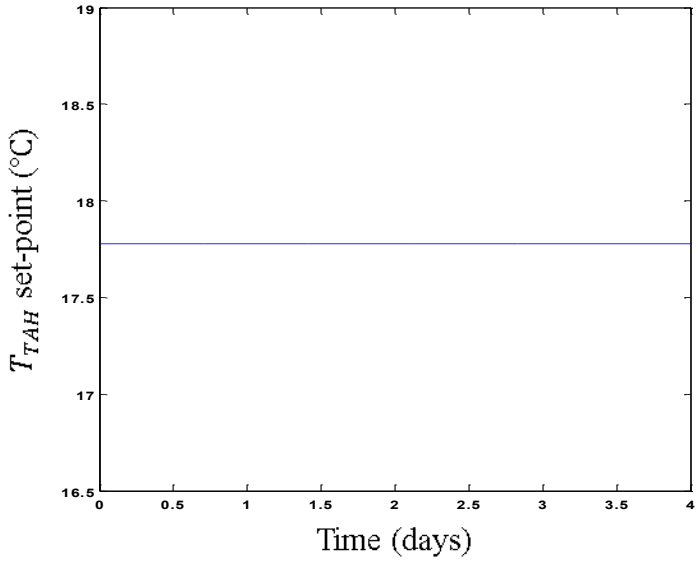



Figure 5.34: The $T_{AHU}$ set-point - case 4 ($T_{AHU}$ is at the warmest (minimum) point)

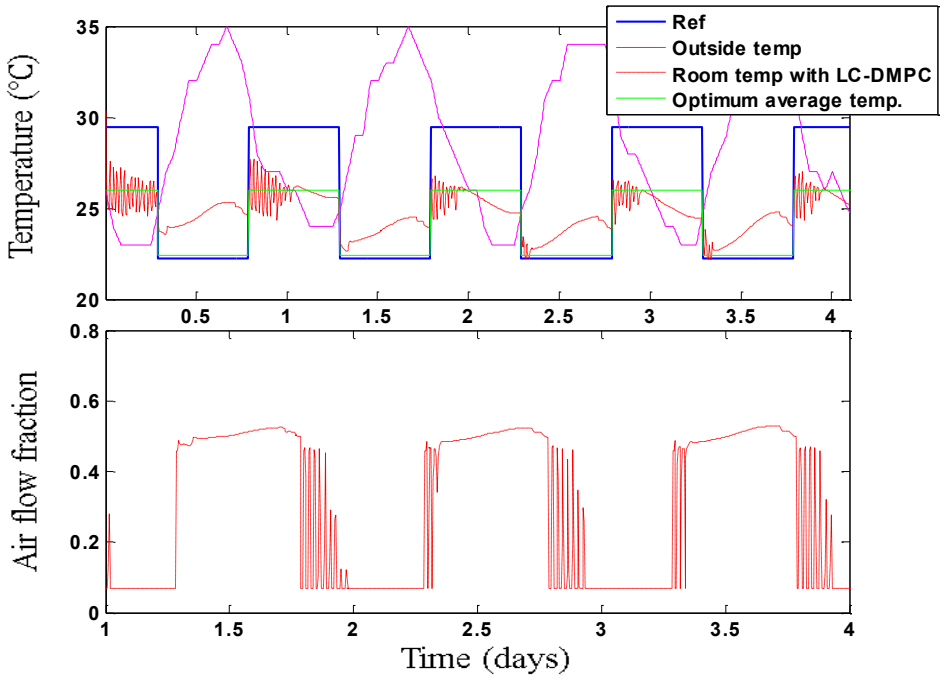



Figure 5.35: Temperature and air flow fraction for room 9 - case 4

### 5.3.5 LC-DMPC Approach and Centralized MPC

A centralized MPC was also implemented using the same building states in case 1. Figures 5.36 through 5.41 show that the solutions with Algorithm 5.1 can converge to a place that is very close to the centralized MPC solution. Moreover, Figure 5.42 gives the eigenvalues of the convergence matrix (2.27a) for the UBO building HVAC system with Algorithm 5.1. The figure shows the convergence (or stability) of the Algorithm.

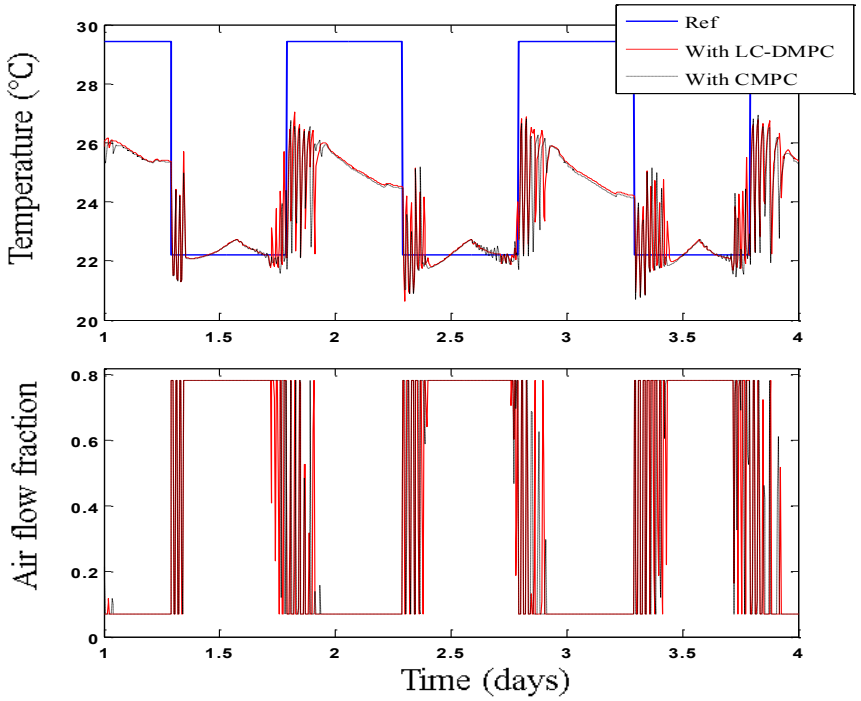



Figure 5.36: Temperature and air flow fraction for room 2 with Algorithm 5.1 and the centralized MPC

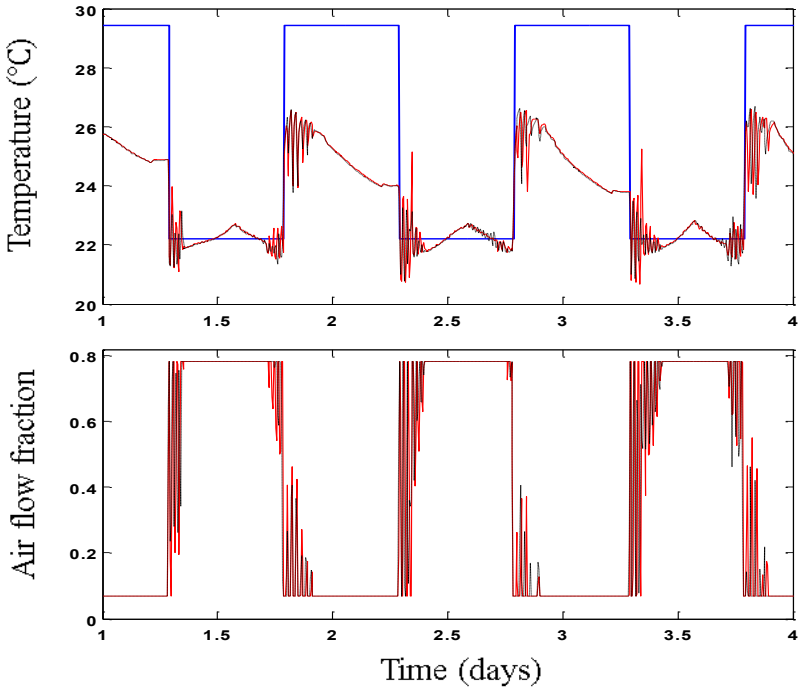



Figure 5.37: Temperature and air flow fraction for room 7 with Algorithm 5.1 and the centralized MPC

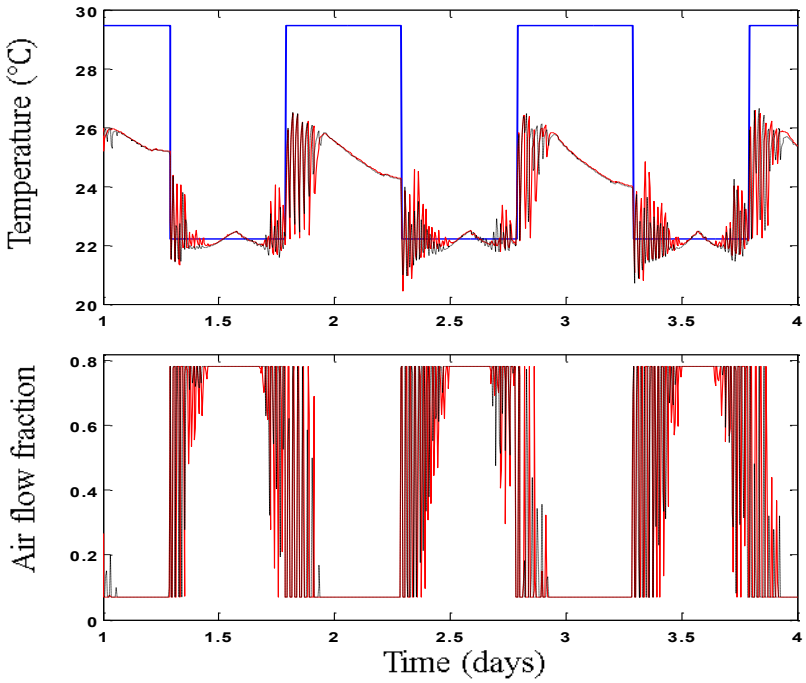



Figure 5.38: Temperature and air flow fraction for room 8 with Algorithm 5.1 and the centralized MPC

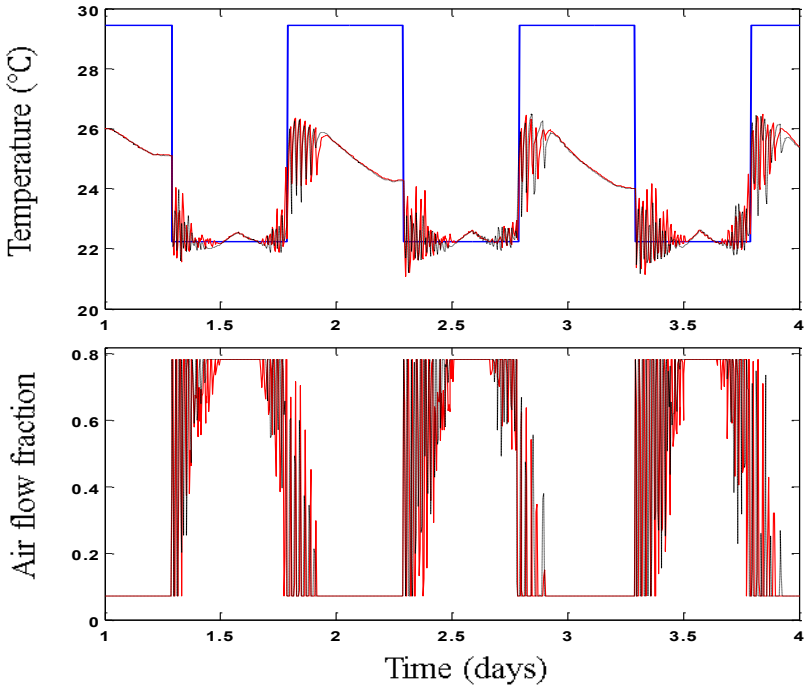



Figure 5.39: Temperature and air flow fraction for room 9 with Algorithm 5.1 and the centralized MPC

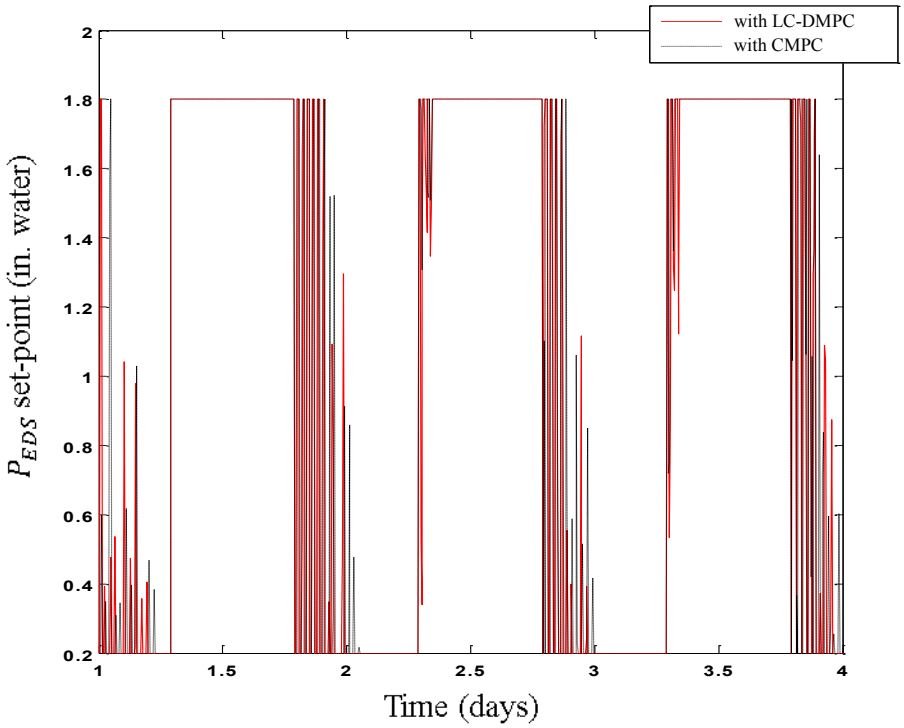



Figure 5.40: The $P_{EDS}$ set-point with Algorithm 5.1 and the centralized MPC

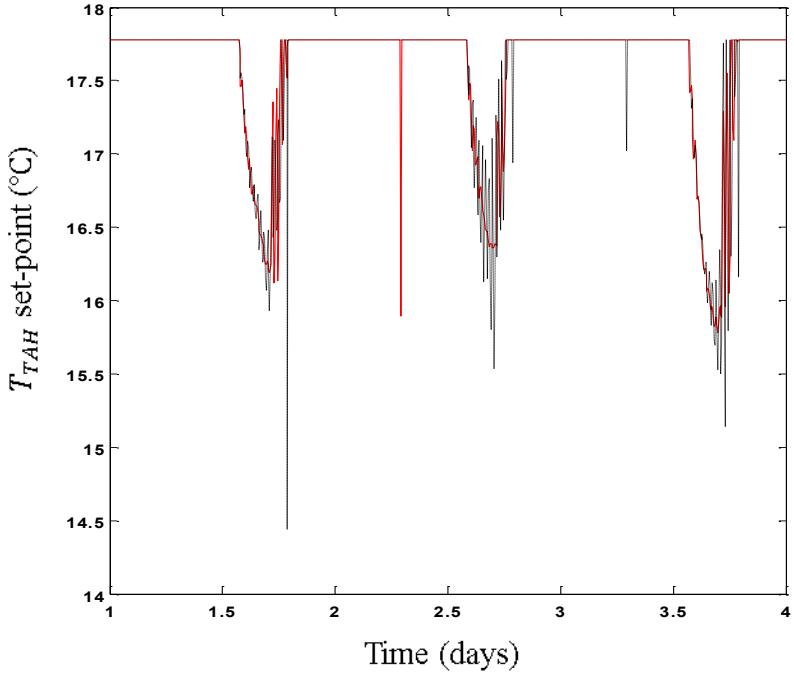



Figure 5.41: The $T_{AHU}$ set-point with Algorithm 5.1 and the centralized MPC

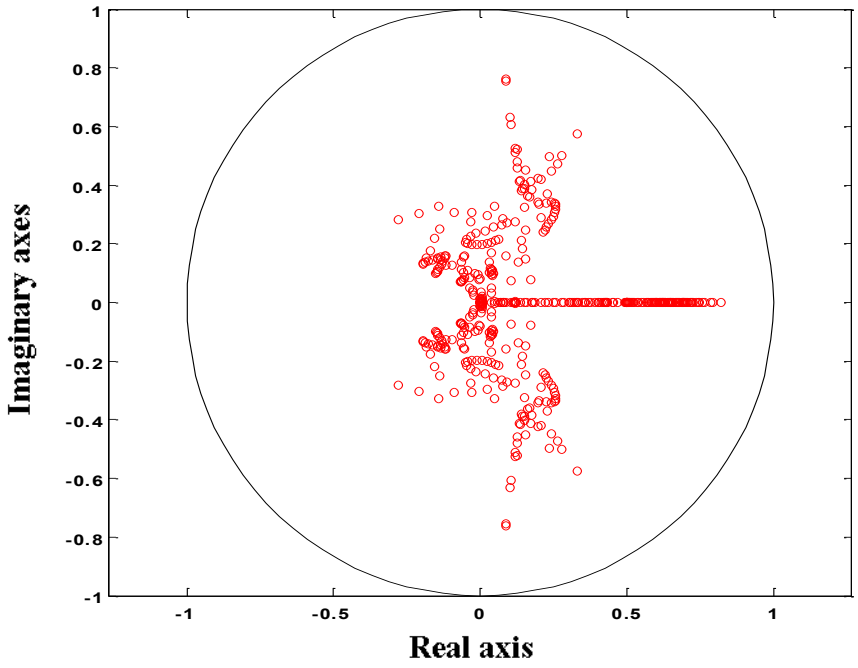


Figure 5.42: Eigenvalues of the convergence matrix (2.27a) for Algorithm 5.1 with the UBO Building HVAC system application

# 6. CONCLUSION AND FUTURE WORKS

## 6.1 Conclusion

The presented research in this dissertation has the main goal of developing an iterative and predictive control architecture for networks with coupled subsystems that can produce stabilizing control actions and at same time optimize overall performance. The work introduced here explores developing such algorithm in two different methods.

In the first method the introduced algorithm can converge to the global optimum solution. In this framework, the global control problem is divided into a number of coupled subsystems based on a neighbor upstream and downstream structure. According to this structure, a subsystem views the coupling signals from upstream neighbors as measured disturbances and at the same time has outputs to downstream neighbors. In contrast with most DMPC schemes, the individual subsystems solve a different cost than the centralized problem. At each iteration, two bi-directional signals are communicated: Predicted disturbances for downstream neighbors and local cost sensitivity for upstream neighbors. Here, the convergence of the algorithm is tested at a centralized monitor that has access to the all local information. The distributed controller with local knowledge can converge to the systemwide solution by sharing information with neighbor agents only. The closed-stability of the distributed controllers relays on the convergence of the algorithm as the main condition. It also requires the local controllers to have long horizons. Long horizons not only reduce the number of iterations as the local problems need more time to be solved but also increase the communication burden. To reduce the

effects of long horizons, Laguerre functions are proposed to parametrize the local decision variables and the exchanged signals. This parametrization allows having more iterations between the local agents and reduces the communication bandwidth.

In the second method, the suboptimal structure of the introduced algorithm is developed. The same subsystem partition structure and communication architecture are used with the new proposed algorithm. A systemwide monitor is no longer need and the global convergence is now ensured by the local controllers. For local closed-stability, the coupled agents need to share the coupling dynamics. Thus, the global convergence and local closed-loop stability are ensured locally. However, this enforces the local controllers to operate in a suboptimal level with respect to the systemwide performance. The conservative design of the local controllers is the reason of the suboptimality for the new algorithm. The distributed controllers are conservative because of the dissipativity of the local exchanged signal dynamics in the iteration domain.

The last port of this thesis is the application of the LC-DMPC algorithm (presented in the first method) using a Heating, Ventilation, and Air Conditioning (HVAC) system in a typical office building. The HVAC system is divided into two subsystems. A zone subsystem that consists of a room and the corresponding controlling VAV box and the Air handling Unit (AHU) subsystem where fan speed and discharge air temperature are controlled. At the zone level, the local distributed MPC regulates the room temperature by computing optimum set-points for the local controllers (usually PI). On the other hand, at the AHU, there is an optimizer that produces the optimum set-points of the end static pressure and discharge air temperature for the local controllers. Economic cost

functions are used for the local MPCs in addition to the AHU optimizer. Therefore, the local zone MPCs and AHU optimizer are attempting to minimize the dollar costs of the zone comfort levels and fan power as well as the chilled water production.

For different priority cases, the algorithm gives the expected optimum solutions. For instance, when the zones and the chilled water operating costs are more important than the electricity cost for running the fan, the algorithm attends to use less chilled water with more fan speed to meet the cooling demands required by the zones. The LC-DMPC algorithm can track the centralized solutions at the zone levels as well as the AHU level.

Linear models are used to represent the dynamics of the rooms at the zone MPCs. Where the control input is the inlet air flow while the AHU discharge air temperature is considered as unmolded disturbance. These linear models, however, do not reveal the actual behavior of the rooms as the control inputs are the products of the inlet air flows and AHU discharge temperature (the cooling loads). To model the effect of the AHU discharge temperature as a linear disturbing signal, a modification of the existed room linear models is proposed. This modification is valid for small variations in the AHU discharge air temperature.

In summary, the main contributions of this dissertation compared to existing work in the areas of DMPC algorithms and building MPC control are:

- Developing an optimal DMPC algorithm that can approach to the systemwide optimum performance through local agent communications only.
- This algorithm has a high modularity and requires a centralized monitor for testing the convergence only.

- Developing a suboptimal DMPC algorithm that does not require (optional) a systemwide supervisor where all decisions can be taken by local agents.
- In case if a supervisor is used by the suboptimal DMPC algorithm, the global convergence can be tested using less information from the network.
- Developing a distributed MPC algorithm for buildings that is able to coordinate the optimality of HVAC system in a distributed approach with less exchanging information requirements between the local defined agents.

## 6.2 Future Works

In this work, the optimum and suboptimum LC-DMPC approaches are introduced for reference tracking. The local agents are solving constrained optimization problems with cost functions that count for the effects for downstream systems. Therefore, it is important to show the recursive feasibility of the distributed problems where the existing feasibility MPC methods do not apply. This issue becomes more important when hard constraints on state or output are imposed or coupled constraints between the neighbor subsystems are proposed. The local performance index has conflicting terms where the first tracking term tries to minimize the differences between the output and a tracking reference which is not same as minimizing the state. The last term, on the other hand, aims at minimizing the outputs for downstream by minimizing the state. Because of this confliction, it is not possible to use the local cost as Lyapunov function directly. However, according to a lemma by Aderson and Moore 1971, this confliction can be eliminated by a certain arrangement of the weighting matrices. This may give a new method for proofing the local LC-DMPC agent stability.

Through a proper scaling of the exchanged signals, one may reduce the conservative problem with the suboptimal LC-DMPC approach. This point needs more developing work. Also, local subsystem information may be used to develop an optimal method for selecting the local free design variables for the global convergence. This new method may, instead of using the $H_\infty$ norm, use the closeness to the systemwide performance as the base idea. It is also would be welcome development efforts to work on the continuous-time version of the proposed algorithms and to extend the work for fault-

detections in networks. As a final suggestion for future applications, a real system (such as a multi-stage compression cycle) could be used to demonstrate the aspects of the suboptimal LCD-DPMC approach.

As future works for the HVAC application of the LC-DMPC algorithm, it would be interesting to include the dynamics of the Air Handling Unit (AHU) and to use a real building as a physical application environment. This can give a closer picture of the algorithm aspects. As simulations, the HVAC application level of the LC-DMPC algorithm introduced in Chapter V can be extend to include the chilled water pumps, multiple AHU systems, and coordinating more than one building. On the other hand, it is better to use bilinear models for the rooms which consequently require developing a bilinear version of the LC-DMPC algorithm to handle such model structures.

## REFERENCES


[1] Energy yearly statistics, 2008. Technical report, EUROSTAT, 2010. Available: http://ec.europa.eu/eurostat/web/energy/data/energy-balances.

[2] U.S. Energy Information Administration, 2012. Annual energy review 2011. Technical report, U.S. Department of Energy. Available: https://www.eia.gov/totalenergy/data/annual/pdf/aer.pdf

[3] R. E. Brown and J. G. Koomey, "Electricity use in California: Past trends and present usage patterns," Energy Policy, vol. 31, pp. 849-864, 2003.

[4] A. Boyano, P. Hernandez, and O. Wolf, "Energy demands and potential savings in European office buildings: Case studies based on EnergyPlus simulations," Energy and Buildings, vol. 65, pp. 19-28, 2013.

[5] K. J. Chua, S. K. Chou, W. M. Yang, and J. Yan, "Achieving better energy-efficient air conditioning - A review of technologies and strategies," Applied Energy, vol. 104, pp. 87-104, 2013.

[6] A. Costa, M. M. Keane, J. I. Torrens, and E. Corry, “Building operation and energy performance: Monitoring, analysis and optimisation toolkit,” Applied Energy, vol. 101, pp. 310-316, 2013.

[7] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," Control Engineering Practice, vol. 11, pp. 733-764, 2003.

[8] M. Morari, and J. H. Lee, “Model predictive control: Past, present and future,” Computers and Chemical Engineering, vol. 23, pp. 667-682, 1999.

[9] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," Automatica, vol. 32, pp. 1361-1379, 1996.

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

[11] E.C. Kerrigan, and J. M. Maciejowski, "Robust feasibility in model predictive control - Necessary and sufficient conditions," In Proceedings of the 40th IEEE Conference on Decision and Control, pp. 728-733, 2001

[12] B. Kim, G. N. Washington, and H. S. Yoon, "Hysteresis-reduced dynamic displacement control of piezoceramic stack actuators using model predictive sliding mode control," Smart Materials and Structures, vol. 21, pp. 1-13, 2012.

[13] Z. Langwen and W. Jingcheng, "Distributed model predictive control with a novel partition method," Proceedings of the 31st Chinese Control Conference, pp. 4108-4113, 2012.

[14] E. Zafiriou and A. L. Marchal, "Stability of SISO quadratic dynamic matrix control with hard output constraints," American Institute of Chemical Engineers Journal, vol. 37, pp. 1550-1560, 1991.

[15] H. Cui and E. W. Jacobsen, "Performance limitations in decentralized control," Journal of Process Control, vol. 12, pp. 485-494, 2002.

[16] D. D. Sourlas and V. Manousiousthakis, "Best achievable decentralized performance," IEEE Transactions on Automatic Control, vol. 40, pp. 1858-1871, 1995.

[17] L. Bakule, "Decentralized control: An overview," Annual Reviews in Control, vol. 32, pp. 87-98, 2008.

[18] G. C. Goodwin, M. E. Salgado, and E. I. Silva, "Time-domain performance limitations arising from decentralized architectures and their relationship to the RGA," International Journal of Control, vol. 78, pp. 1045-1062, 2005.

[19] L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems," Automatica, vol. 42, pp. 1231-1236, 2006.

[20] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - A review," Journal of Process Control, vol. 19, pp. 723-731, 2009.

[21] P. D. Christofides, R. Scattolini, D. M. Pena., and J. Liu, "Distributed model predictive control: A tutorial review and future research directions", Computers and Chemical Engineering, vol. 51, pp. 21-41, 2013.

[22] Y. Zheng, S. Li, and H. Qiu, "Networked coordination-based distributed model predictive control for large-scale system," IEEE Transactions on Control Systems Technology, vol., 21, pp., 991-998, 2013.

[23] P. Giselsson, and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," 49th IEEE Conference on Decision and Control, pp. 7272-7277, 2010.

[24] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi, "Decentralized model predictive control via dual decomposition," Proceedings of the 47th IEEE Conference on Decision and Control, pp. 381–386, 2008.

[25] C. Conte, T. Summers, M.N. Zeilinger, M. Morari, and C.N. Jones, “Computational aspects of distributed optimization in model predictive control,” Proceedings of the 51th IEEE Conference on Decision and Control, pp 6819-6824, 2012.

[26] T. Keviczky and K. H. Johansson, “A study on distributed model predictive consensus,” Proceedings of the 17th World Congress of the International Federation of Automatic Control, vol. 41, pp 1516-1521, 2008.

[27] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, “On decentralized negotiation of optimal consensus,” Automatica, vol. 44, pp. 1175-1179, 2008.

[28] P-D. Morosan, R. Bourdais, D. Dumur, and J. Buisson, “A distributed MPC strategy based on Benders' decomposition applied to multi-source multi-zone temperature regulation,” Journal of Process Control, vol. 21, pp. 729-737, 2011.

[29] D. Jia and B. Krogh, “Min-max feedback model predictive control for distributed control with communication,” in Proceedings of the American Control Conference, pp. 507-4512, 2002.

[30] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, “Distributed MPC strategies with application to power system automatic generation control,” IEEE Transactions on Control Systems Technology, vol. 16, pp. 1192-1206, 2008.

[31] M. Farina, and R. Scattolini "Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," Automatica, vol. 48, pp. 1088-1096, 2012.

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

[33] J. B. Rawlings, and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," Journal of Process Control, vol. 18, pp. 839-845, 2008.

[34] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Distributed model predictive control of large-scale systems," Assessment and Future Directions of Nonlinear Model Predictive Control, vol. 358, pp. 591-605, 2007.

[35] I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R Negenborn, B. De Schutter, F. Valencia and J. Espinosa., "A comparative analysis of distributed MPC techniques applied to the HD- MPC four-tank benchmark," Journal of Process Control, vol. 21, pp. 800-815, 2011.

[36] V. Putta, G. Zhu, D. Kim, J. Hu, and J. E. Braun, "A Distributed approach to efficient model predictive control of building HVAC systems," International High Performance Buildings Conference, pp. 3516-3525, 2012.

[37] P. Giselsson, M. D, Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," Automatica, vol. 49, pp. 829-833, 2013.

[38] H. Chen, and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," Automatica, vol. 34, pp. 1205-1217, 1998.

[39] L. Magni, and R. Sepulchre, "Stability margins of nonlinear receding horizon control via inverse optimality," System and Control Letters, vol. 32, pp., 241-245, 1997.

[40] S. S. Keerthi, and E. G. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations," Journal of Optimization Theory and Applications, vol. 57, pp. 265-293, 1988.

[41] J. A. Primbs, V. Nevistic, and J. C. Doyle, "A receding horizon generalization of pointwise min-norm controllers," IEEE Transactions on Automatic Control, vol. 45, pp. 898-909, 2000.

[42] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," Proceeding of the 40th IEEE Conference on Decision and Control, pp. 4826-4831, 2001.

[43] F. Xie, and R. Fierro, "First-state contractive model predictive control of nonholonomic mobile robots," Proceeding of the American Control Conference, pp. 3494-3499, 2008.

[44] X. Cheng, and D. Jia, "Robust stability constrained model predictive control," Proceedings of American Control Conference, pp. 1580-1585, 2004.

[45] X. Cheng and D. Jia, "Robust stability constrained model predictive control with state estimation," Proceedings of American Control Conference, pp. 1581-1586, 2006.

[46] W. Chen, T. Zhenc, M. Chen, and X. Li, “Improved nonlinear model predictive control based on genetic algorithm,” In Advanced Model Predictive Control, published by In Tech, Croatia, pp. 49-64, 2011.

[47] C. Conte, N. R. Voellmy, M. N. Zeilinger, M. Morari, and C. N. Jones, "Distributed synthesis and control of constrained linear systems," Proceedings of American Control Conference, pp. 6017-6022, 2012.

[48] A.N. Venkat, "Distributed model predictive control: Theory and applications, " Ph.D. thesis, Department of Chemical Engineering, University of Wisconsin-Madison, 2006.

[49] R. Hermans, M. Lazar and A. Jokic, "Distributed Lyapunov-based MPC, " Chapter in Distributed Model Predictive Control Made Easy, vol. 69 of the series Intelligent Systems, Control and Automation: Science and Engineering, pp. 225-241, 2014.

[50] D. Jia, and B.H. Krogh, "Distributed model predictive control," Proceedings of the American Control Conference, pp. 2767-2772, 2001.

[51] J. A. Primbs, and V. Nevistic, “Feasibility and stability of constrained finite receding horizon control,” Automatica, vol. 36, pp. 965-971, 2000.

[52] J. A. Primbs, and V. Nevistic, “Constrained finite receding horizon linear quadratic control,” Proceedings of the 36$^{th}$ IEEE Conference on Decision and Control, vol. 4 pp. 3196 -3201, 1997.

[53] A. Boccia, L. Grune, and K. Worthmann, “Stability and feasibility of state constrained MPC without stabilizing terminal constraints,” Systems and Control Letters, vol. 72, pp. 14-21, 2014.

[54] K. Worthmann, "Estimates on the prediction horizon length in model predictive control," Proceedings of the 20th International Symposium on Mathematical Theory of Networks and Systems, 2012.

[55] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," Automatica, vol. 38, pp. 3-20, 2002.

[56] S. Hovland, K. Willcox, and J. T. Gravdahl, "MPC for large-scale systems via model reduction and multiparametric quadratic programming," Proceedings of the 45th IEEE Conference on Decision and Control, pp., 3418-3423, 2006.

[57] P. Tondel, T. A. Johansen, and A. Bemporad, "An algorithm for multiparametric quadratic programming and explicit MPC solutions," Automatica, vol. 39, pp. 489-497, 2003.

[58] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," IEEE Transaction on Control System Technology, vol. 18, pp. 267-278, 2010.

[59] C-J. Ong, and Z. Wang, "Reducing variables in Model Predictive Control of linear system with disturbances using singular value decomposition," Systems and Control Letters, vol. 71, pp. 62-68, 2014.

[60] X. Cai, P. Sun, J. Chen, and L. Xie, "Rapid distributed model predictive control design using singular value decomposition for linear systems," Journal of Process Control, vol. 24, pp. 1135-1148, 2014.

[61] L. Wang, "Discrete model predictive controller design using Laguerre functions," Journal of Process Control, vol. 14, pp. 131-142, 2004.

[62] G. V-Palomo, and J. Rossiter, "Using Laguerre functions to improve efficiency of multi-parametric predictive control," Proceedings of American Control Conference, pp. 4731-4736, 2010.

[63] M. J. Tippett and J. Bao, "Control of plant-wide systems using dynamic supply rates," Automatica, vol. 50, pp. 44-52, 2014.

[64] C. L. Byrnes, A. Isidori A., and J. C. Willems, "Passivity feedback equivalence and the global stabilization of minimum phase nonlinear systems," IEEE Transactions on Automatic Control, vol. 36, pp. 1228-1240, 1991.

[65] A. A. Alonso, and B. E. Ydstie, "Stabilization of distributed systems using irreversible thermodynamics," Automatica, vol. 37, pp. 1739-1755, 2001.

[66] O. J. Rojas, R. Setiawan, and J. Bao, "Dynamic operability analysis of nonlinear process networks based on dissipativity," American Institute of Chemical Engineers Journal, vol. 55, pp. 963-982, 2009.

[67] D. Hioe, J. Bao, and B. E. Ydstie, "Dissipativity analysis for networks of process systems," Computers & Chemical Engineering, vol. 50, pp. 207-219, 2013.

[68] C. Zheng, M. J. Tippett, and J. Bao, "Dissipativity-based distributed model predictive control with low rate communication," American Institute of Chemical Engineers Journal, vol. 61, pp. 3288-3303, 2015.

[69] P. Varutti, B. Kern, and R. Findeisen, "Dissipativity-based distributed nonlinear predictive control for cascaded systems," Preprints of the 8th International Federation of Automatic Control Symposium on Advanced Control of Chemical Processes, pp. 439-444, 2012.

[70] M. A. Muller, D. Angeli, and F. Allgower, “On necessity and robustness of dissipativity in economic model predictive control,” IEEE Transactions on Automatic Control, vol. 60, pp. 16714-1676, 2015.

[71] A. Tahirovic, and G. Magnani, “Passivity-based model predictive control for mobile robot navigation planning in rough terrains,” The IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 307-312, 2010.

[72] T. Tran, K-V Ling, and J. M. Maciejowski, “Model predictive control via quadratic dissipativity constraint,” The 53rd IEEE Conference on Decision and Control, pp. 6689-6694, 2014.

[73] C. Wang, A. Ohsumi, and I. Djurovic , “Model predictive control of noisy plants using Kalman predictor and filter,” Proceedings of IEEE TENCON, pp. 1404-1407, 2002.

[74] D. Q. Mayne, S. V. Rakovic, R. Findeisen, and F. Allgower, “Robust output feedback model predictive control of constrained linear systems,” Automatica, vol. 42, pp. 1217-1222, 2006.

[75] J. Yan and R. R. Bitmead, “Incorporating state estimation into model predictive control and its application to network traffic control,” Automatica, vol. 41, pp. 595-604, 2005.

[76] M. Geetha J. Jovitha and V. Devatha, "Design of state estimation based Model predictive controller for a two tank interacting system," Procedia Engineering, vol. 64, pp. 244-253, 2013.

[77] M. Hong and S. Cheng, "Model predictive control based on Kalman filter for constrained Hammerstein-wiener systems," Hindawi Publishing Corporation Mathematical Problems in Engineering, vol. 2013, 2013.

[78] D. D. Ruscio, "Model predictive control with integral action: A simple MPC algorithm," Modeling, Identification and Control, vol. 34, pp. 119-129, 2013.

[79] R. Vadigepalli and F. J. Doyle, "A distributed state estimation and control algorithm for plantwide processes," IEEE Transaction on Control System Technology, vol. 11, pp. 119-127, 2003.

[80] M. Farina, and R. Scattolini, "An output feedback distributed predictive control algorithm," Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, pp. 8139-8144, 2011.

[81] Y. Zheng, S. Li, X. Wang, "Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process," Journal of Process Control, vol. 19, pp. 1427-1437, 2009.

[82] S. R-Yamchi, M. Cychowski, R. R. Negenborn, B. D. Schutter, K. Delaney. and J. Connell, "Kalman filter-based distributed predictive control of large-scale multi-rate systems: Application to power networks," IEEE Transaction on Control Systems Technology, vol. 21, pp. 27-39, 2013.

[83] K.Menighed C. Aubrun and J-J. Yamé, "Distributed state estimation and model predictive control: Application to fault tolerant control," Proceedings of IEEE International Conference on Control and Automation, pp. 936-941, 2009.

[84] J. M. Maciejowski, "Predictive control with constraints", Prentice Hall, 2001.

[85] Z. Tan, Y. C. Soh, and L. Xie, "Dissipative control for linear discrete-time systems," Automatica, vol. 35, pp. 1557-1564, 1999.

[86] P. J. Moylan and D. J. Hill, "Stability criteria for large-scale systems," IEEE Transactions on Automatic Control, vol. AC-23, pp., 143-149, 1978.

[87] W. Lin and C. I. Byrness, "Passivity and absolute stabilization of a class of discrete-time Nonlinear Systems," Automatica, vol. 31, pp. 263-267, 1995.

[88] D. Hill and P. Moylan, "The Stability of nonlinear dissipative systems," IEEE Transactions on Automatic Control, vol. 21 , pp. 708-711, 1976.

[89] M. Gree and D. J. N. Limebeer, "Linear robust control," Prentice-Hall, Inc. 1995.

[90] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," American Institute of Chemical Engineers Journal, vol. 39, pp. 262-287, 1993.

[91] D. Simon, J. Lofberg, and T. Glad, "Reference tracking MPC using terminal set scaling," 51st IEEE Conference on Decision and Control, pp.4543-4548, 2012.

[92] D. Limon, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained MPC without terminal constraint," IEEE Transactions on Automatic Control, vol. 51, pp. 832-836, 2006.

[93] D. M. Pena, T. Alamo, M. Lazar, and W.P.M.H. Heemes, "Design of ISS-lyapunov functions for discrete-time linear uncertain systems," Proceedings of the 17th World Congress, The International Federation of Automatic Control, pp. 1135-1140, 2008.

[94] U. S. department of Energy. (10/11/2016). EnergyPlus Energy Simulation Software. Available: http://apps1.eere.energy.gov/buildings/energyplus/

[95] T. Buildings. (SketchUp Home Page, 10/11/2016). Available: http://www.sketchup.com/

[96] U. S. department of Energy, Office of Energy Efficiency & Renewable Energy (10/11/2016). Available: http://energy.gov/eere/buildings/downloads/openstudio-0

[97] The Building Controls Virtual Test Bed (10/11/2016). Available: http://simulationresearch.lbl.gov/bcvtb

[98] MLE+ Manual (10/11/2016). Available: http://txn.name/mleplus/mlep_manual.html

[99] R. Chintala and B. P. Rasmussen, "Automated multi-zone linear parametric black box modelling approach for building HVAC systems," Proceedings of the ASME Dynamic Systems and Control Conference, pp. 1-10, 2015.

[100] C. Bay, "Efficiency and control of energy systems in buildings," PhD thesis, Department of Mechanical Engineering, Texas A&M University (under preparations).

[101] J. Cigler, S. Privara, Z. Vana, E. Zacekova, and L. Ferkl, "Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution," Energy and Buildings, vol. 52, pp. 39-49, 2012.

[102] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," IEEE Transection on Control System Technology, vol. 8, pp. 456-465, May 2000.

[103] R. Jalal and B. Rasmussen, "Limited-communication distributed model predictive control for coupled and constrained linear systems," IEEE Transaction on Control System Technology, accepted, 2016.

# APPENDIX A

## EXTENDING THE INTERCONNECTING MATRIX Γ ALONG ANY PREDICTION HORIZON $N_p$

Extending the interconnecting matrix Γ along the prediction horizon is not trivial. In this appendix, we will explain how to such extension using the local information about the input disturbance $v_i$ only for all subsystems in the given network.

To be more specific, we have used the network example given in chapter 5. There are 9 coupled rooms with room temperature as output disturbance variables. Therefore:

input_distrubance = {'T_room1', 'T_room2', 'T_room3', 'T_room4', 'T_room5', 'T_room6', 'T_room7', 'T_room8', 'T_room9'}

Each subsystem has local information by which it can specify the local input disturbance $v_i$ as a subset of the set of input_distrubance. For a specific identification, we had the following couplings:

$$v_1 = (\text{'T\_room5'} \quad \text{'T\_room6'})^T$$

$$v_2 = (\text{'T\_room6'} \quad \text{'T\_room4'} \quad \text{'T\_room3'})^T$$

$$v_3 = (\text{'T\_room5'} \quad \text{'T\_room2'} \quad \text{'T\_room4'})^T$$

$$v_4 = (\text{'T\_room7'} \quad \text{'T\_room2'} \quad \text{'T\_room3'})^T$$

$$v_5 = (\text{'T\_room8'} \quad \text{'T\_room2'} \quad \text{'T\_room4'})^T$$

$$v_6 = (\text{'T\_room7'} \quad \text{'T\_room3'} \quad \text{'T\_room2'})^T$$

$$v_7 = (\text{'T\_room2'} \quad \text{'T\_room9'})^T$$

$$v_8 = (\text{'T\_room2'} \quad \text{'T\_room9'})^T$$

$$v_9 = (\text{'T\_room2'} \quad \text{'T\_room7'} \quad \text{'T\_room1'})^T$$

Or in term of input_distrubance we can have numerical values of the local network disturbances as following:

$$v_1 = [5 \quad 6]^T,\ v_2 = [6 \quad 4 \quad 3]^T,\ v_3 = [5 \quad 2 \quad 4]^T$$

$$v_4 = [7 \quad 2 \quad 3]^T,\ v_5 = [8 \quad 2 \quad 4]^T,\ v_6 = [7 \quad 3 \quad 2]^T$$

$$v_7 = [2 \quad 9]^T,\ v_8 = [2 \quad 7]^T,\ v_9 = [2 \quad 7 \quad 1]^T$$

By stacking all local input disturbances into a new variable $V$ as:

$$V_{24\times1} = [v_1^T \quad v_2^T \quad v_3^T \quad v_4^T \quad v_5^T \quad v_6^T \quad v_7^T \quad v_8^T \quad v_9^T]^T$$

gives the number of variables in each local output disturbance $z_i$ the sort of these variables with respect to $V$, as well as the output network disturbance $V_{24\times1}$. The following simple Matlab code can extrapolate these information:

```
  Lz = []; % length of local zi
 orz = []; % order of zi w.r.t. V
for j = 1:length(input_distrubance)
  orzi = find(V == j); Lzi = length(orzi);
    Lz = [Lz;Lzi];
   orz = [orz;orzi];
end

Z = [];
for i = 1:length(Lz)
    for k = 1:Lz(i)
        Z = [Z;input_distrubance(i)];
    end
end
```

The results of above are:

```
 Lz = [1   7   3   3   2   2   4   1   1]^T
orz = [24   7   10   13   17   18   20   22   5   11   16   4   8   14   1   6   2
                                                    3   9   15   21   23   12   19]^T
```

and

```
Z = ['T_room1'  'T_room2'  'T_room2'  'T_room2'  'T_room2'  'T_room2'
     'T_room2'  'T_room2'  'T_room3'  'T_room3'  'T_room3'  'T_room4'
     'T_room4'  'T_room4'  'T_room5'  'T_room5'  'T_room6'  'T_room6'
     'T_room7'  'T_room7'  'T_room7'  'T_room7'  'T_room8'  'T_room9']^T
```

The vector $V$ also gives the length of each individual $v_i$ as:

```
lvi =[length(v1);length(v2);length(v3);length(v4);
      length(v5);length(v6); length(v7);length(v8);
      length(v9)];
```

or

```
Lzi = [2   3   3   3   4   4   3   3   3]^T .
```

Now using these information, we can calculate the value of $\Gamma$ when $N_p = 1$. This is done using the following code:

```
orv = []; % order of v w.r.t. z or (Γ at Np = 1)
for j = 1:length(Z)
    q   = find(orz == j);
    orv = [orv;q];
end
orv = orv';
```

or

`orz` = [15 17 18 12 9 16 2 13 19 3 10 23 4 14 20
11 5 6 24 7 21 8 22 1]$^T$

To extend Γ for any $N_p$, we need two more information which can be determined from the already computed vectors. First, we need to know which element in $Z$, the corresponding element in $V$ is referring to. This is computed through:

```
erdv = []; % reffered element in z
T1 = []; T2 = []; T3 = [];
for t = 1:length(gz)
    for k = 1:gz(t)
        T1 = [T1;k];
        T2 = [T2;gz(t)];
    end
end
for i = 1:length(orv)
    f = orv(i);
    erdv = [erdv;T1(f)];
      T3 = [T3;T2(f)];
end
erdv = erdv';
```

or

`erdv` = [1 1 2 1 1 2 1 2 1 2 2 1 3 3 2
3 4 5 1 6 3 7 4 1]$^T$

Second, we need to know the length of the $z_i$ in which an element in $v_i$ in `erdv` is referring to. This is already being computed in the above code and given in the variable `T3`, therefore:

```
% ngz is the no. of elements in each group w.r.t. erdv
```

```
ngz = T3';
```

where

$$ngz = [2 \quad 2 \quad 2 \quad 3 \quad 3 \quad 2 \quad 7 \quad 3 \quad 4 \quad 7 \quad 3 \quad 1 \quad 7 \quad 3 \quad 4 \quad 3 \quad 7 \quad 7 \quad 1 \quad 7 \quad 4 \quad 7 \quad 4 \quad 1]^T$$

Now all information are available to write the linear mapping matrix Γ for any given $N_p$ as follows:

1. Writing Γ as a zero matrix with dimension of `sum(Lz)*Np`:

```
cap_gam = zeros(sum(Lz)*Np,sum(Lz)*Np);
```

2. Shifting the elements in `ordv` (Γ at Np = 1) by the elements in `Lz`:

```
for i = 1:sum(Lz)
    if erdv(i) == 1
        mogam(i) = (ordv(i)-1)*Np + 1;
    else
        mogam(i) = (ordv(i)-erdv(i))*Np + erdv(i);
    end
end
```

3. The elements in `mogam` are now shifted by the elements in `ngz`:

```
nt = [];
for j =  1:Np-1
    nt1 = mogam + j*ngz;
    nt = [nt; nt1'];
end
nt = [mogam';nt];
```

4. For the final step, all shifted information are now stored in the variable `p`:

```
 p = []; h = 0; g = 0;
L1 = length(ordv); L2 = length(Lv);
for j = 1:L2
    for k = 1:Np
        for i = 1:Lv(j)
            p = [p;nt(i+h)];
        end
       h = h + L1;
    end
    h = Lv(j) + g; g = h;
end
```

5. Finally the matrix Γ along $N_p$ is computed by changing the elements given by `p` in `cap_gam` to one:

```
for i = 1:Np*L1
    cap_gam(i,p(i)) = 1;
end
```

Recall that all above steps are constant for any network and the only need information are prediction horizon $N_p$ and the first vector input_distrubance.

## APPENDIX B

## THE INTERCONNECTED SUBSYSTEMS EXAMPLE USED IN CHAPTER IV: ALL RELATED INFORMATION

This appendix provides all information that has been used for the example of the chapter IV and details some results that have not be been stated in the chapter.

First, the dynamic of each subsystem as well as variables and matrices that have been used in simulation for the local cost functions are given in table B.1 as follows:

**Table B.1: Local Subsystem Information**

| **For $i = 1, 2, 3$** | **Subsystems** | | |
|---|---|---|---|
| | **1** | **2** | **3** |
| State matrix $A_i$ | $\begin{bmatrix} 0.3525 & -0.5946 & 0.3841 \\ -0.5946 & 0.0381 & 0.4962 \\ 0.3841 & 0.4962 & 0.3897 \end{bmatrix}$ | $\begin{bmatrix} -0.0470 & 0.6663 \\ 0.6663 & -0.0412 \end{bmatrix}$ | $\begin{bmatrix} -0.3654 & -0.5016 \\ -0.5016 & -0.0921 \end{bmatrix}$ |
| Control input matrix $B_{u,i}$ | $\begin{bmatrix} 0 & 0.8400 \\ -1.1448 & -0.6138 \\ -2.2324 & 1.1722 \end{bmatrix}$ | $\begin{bmatrix} -0.8727 \\ -0.9308 \end{bmatrix}$ | $\begin{bmatrix} -2.1246 \\ 0 \end{bmatrix}$ |
| Disturbance input matrix $B_{v,i}$ | $\begin{bmatrix} -0.9582 & 0.5599 \\ 0.9531 & -1.0380 \\ 0.7034 & -0.1783 \end{bmatrix}$ | $\begin{bmatrix} 0.0762 & 0 \\ 0 & 2.1315 \end{bmatrix}$ | $\begin{bmatrix} -0.2500 & 0.8200 \\ 0.6800 & -1.0824 \end{bmatrix}$ |
| Regulated output matrix $C_{y,i}$ | $\begin{bmatrix} -0.4699 & 0 & -0.8314 \\ 0 & -0.1975 & -1.0486 \end{bmatrix}$ | $[-0.3258 \quad 0]$ | $[0 \quad 0.7412]$ |
| Disturbance output matrix $C_{z,i}$ | $\begin{bmatrix} -0.9599 & -0.1574 & 0 \\ 0 & 0 & 0 \\ 0 & -0.1975 & -1.0486 \end{bmatrix}$ | $\begin{bmatrix} -0.3258 & 0 \\ -0.3258 & 0 \end{bmatrix}$ | $[0 \quad 0]$ |
| Disturbance output matrix $D_{z,i}$ | $\begin{bmatrix} 0 & 0 \\ 0.5249 & -0.7724 \\ 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ | $[0.1224]$ |
| Initial condition | $[0.00 \quad 0.20 \quad -0.98]^T$ | $[-0.10 \quad 2.00]^T$ | $[3.20 \quad 0.00]^T$ |
| Reference $r_i$ | $[0.80 \quad 0.50]^T$ | 1.20 | 2.00 |
| State weighting matrix $q_i$ | $2I_3$ | $2I_2$ | $6I_2$ |
| control weighting matrix $s_i$ | $\begin{bmatrix} 0.10 & 0.00 \\ 0.00 & 0.30 \end{bmatrix}$ | 0.20 | 0.50 |

Second the LC-DMPC input and output vectors, interconnecting matrix Γ, and the mapping matrix $H$ for network dissipativity are detailed.

The network input disturbance vector with local related information is given by:

$$V = \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} v_{1,2} \\ v_{1,3} \\ v_2 \\ v_{3,1} \\ v_{3,2} \end{Bmatrix} = \begin{Bmatrix} y_{2,1} \\ u_3 \\ y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ y_{2,3} \end{Bmatrix}$$

While the network output disturbance vector with local related information is given by:

$$Z = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} z_{1,2} \\ z_{1,3} \\ z_{2,1} \\ z_{2,3} \\ z_3 \end{Bmatrix} = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ y_{2,1} \\ y_{2,3} \\ u_3 \end{Bmatrix}$$

For prediction horizon equal to one ($N_p = 1$), these two vectors can be related as:

$$\underbrace{\begin{Bmatrix} y_{2,1} \\ u_3 \\ y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ y_{2,3} \end{Bmatrix}}_{V} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\Gamma} \underbrace{\begin{Bmatrix} y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ y_{2,1} \\ y_{2,3} \\ u_3 \end{Bmatrix}}_{Z}$$

The mapping matrix $H$ for network dissipativity is determined as follows:

For the local information dynamics, let the input be denoted as $u_i$ and the output as $y_i$ where: $u_i = \{\mathrm{V}_i^T \quad \Psi_i^T\}^T$ and $y_i = \{\mathrm{Z}_i^T \quad \gamma_i^T\}^T$.

and let the total output be given by:

$$\mathrm{y} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = \begin{Bmatrix} \mathrm{Z}_1 \\ \gamma_1 \\ \mathrm{Z}_2 \\ \gamma_2 \\ \mathrm{Z}_3 \\ \gamma_3 \end{Bmatrix} = \begin{Bmatrix} z_{1,2} \\ z_{1,3} \\ \gamma_{1,2} \\ \gamma_{1,3} \\ z_{2,1} \\ z_{2,3} \\ \gamma_2 \\ z_3 \\ \gamma_{3,1} \\ \gamma_{3,2} \end{Bmatrix} = \begin{Bmatrix} y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ \gamma_{1,2_{y_2}} \\ \gamma_{1,3_{u_3}} \\ y_{2,1} \\ y_{2,3} \\ \gamma_{2_{y_{1,1}}} \\ \gamma_{2_{u_{1,2}}} \\ u_3 \\ \gamma_{3,1_{y_{1,2}}} \\ \gamma_{3,2_{y_2}} \end{Bmatrix}$$

Then for subsystem 1, we can write the mapping between local input $u_1$ and total output y as:

$$u_1 = \begin{Bmatrix} \mathrm{V}_1 \\ \Psi_1 \end{Bmatrix} = \begin{Bmatrix} v_{1,2} \\ v_{1,3} \\ \psi_{1,2} \\ \psi_{1,3} \end{Bmatrix} = \begin{Bmatrix} y_{2,1} \\ u_3 \\ \gamma_{2_{y_{1,1}}} \\ \gamma_{2_{u_{1,2}}} \\ \gamma_{3,1_{y_{1,2}}} \end{Bmatrix} = H_1 \mathrm{y} = H_1 \begin{Bmatrix} y_{1,1} \\ u_{1,2} \\ y_{1,2} \\ \gamma_{1,2_{y_2}} \\ \gamma_{1,3_{u_3}} \\ y_{2,1} \\ y_{2,3} \\ \gamma_{2_{y_{1,1}}} \\ \gamma_{2_{u_{1,2}}} \\ u_3 \\ \gamma_{3,1_{y_{1,2}}} \\ \gamma_{3,2_{y_2}} \end{Bmatrix}$$

where

$$H_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Similarly for subsystem 2:

$$u_2 = \begin{Bmatrix} \mathrm{V}_2 \\ \Psi_2 \end{Bmatrix} = \begin{Bmatrix} v_2 \\ \psi_{2,1} \\ \psi_{2,3} \end{Bmatrix} = \begin{Bmatrix} y_{1,1} \\ u_{1,2} \\ \gamma_{1,2_{y_2}} \\ \gamma_{3,2_{y_2}} \end{Bmatrix} = H_2 \mathrm{y}$$

where

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and for subsystem 3:

$$u_3 = \begin{Bmatrix} \mathrm{V}_3 \\ \Psi_3 \end{Bmatrix} = \begin{Bmatrix} v_{3,1} \\ v_{3,2} \\ \psi_3 \end{Bmatrix} = \begin{Bmatrix} y_{1,2} \\ y_{2,3} \\ \gamma_{1,3_{u_3}} \end{Bmatrix} = H_3 \mathrm{y}$$

where $H_3$ is given as:

$$H_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now let us define the total input vector as:

$$\mathrm{u} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} \mathrm{V}_1 \\ \Psi_1 \\ \mathrm{V}_2 \\ \Psi_2 \\ \mathrm{V}_3 \\ \Psi_3 \end{Bmatrix} = \begin{Bmatrix} v_{1,2} \\ v_{1,3} \\ \psi_{1,2} \\ \psi_{1,3} \\ v_2 \\ \psi_{2,1} \\ \psi_{2,3} \\ v_{3,1} \\ v_{3,2} \\ \psi_3 \end{Bmatrix} = \begin{Bmatrix} y_{2,1} \\ u_3 \\ \gamma_{2_{y_{1,1}}} \\ \gamma_{2_{u_{1,2}}} \\ \gamma_{3,1_{y_{1,2}}} \\ y_{1,1} \\ u_{1,2} \\ \gamma_{1,2_{y_2}} \\ \gamma_{3,2_{y_2}} \\ y_{1,2} \\ y_{2,3} \\ \gamma_{1,3_{u_3}} \end{Bmatrix}$$

which can be related to the total output vector by simply stacking $H_1, H_2$, and $H_3$ in a column vector and writing:

$$\mathrm{u} = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} \mathrm{y} \;\text{ or }\; \mathrm{u} = H\mathrm{y}$$

where

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Finally the dynamics of the possible three combinations are given.

Combining subsystems 1 & 2:

The dynamics of subsystem 1 is:

$$\begin{aligned} x_1^+ &= A_1 x_1 + B_{u,1} u_1 + B_{v,1} \begin{bmatrix} v_{1,2} \\ v_{1,3} \end{bmatrix} \\ y_1 &= C_{y,1} x_1 \\ z_1 &= \begin{bmatrix} z_{1,2} \\ z_{1,3} \end{bmatrix} = C_{z,1} x_1 + D_{z,1} u_1 \end{aligned}$$

and for subsystem 2:

$$\begin{aligned} x_2^+ &= A_2 x_2 + B_{u,2} u_2 + B_{v,2} v_2 \\ y_2 &= C_{y,2} x_2 \\ z_2 &= C_{z,2} x_2 + D_{z,2} u_2 \end{aligned}$$

Then the combined dynamics would be:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{u,1} & 0 \\ 0 & B_{u,2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} B_{v,1} & 0 \\ 0 & B_{v,2} \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{1,3} \\ v_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} z_{1,3} \\ z_{2,3} \end{bmatrix} = \begin{bmatrix} C_{z,1,3} & 0 \\ 0 & C_{z_{,,}3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_{z,1,3} & 0 \\ 0 & D_{z_{,,}3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

or

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{u,1} & 0 \\ 0 & B_{u,2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} [B_{v,1,2} & B_{v,1,3}] & 0 \\ 0 & & B_{v,2} \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{1,3} \\ v_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} z_{1,3} \\ z_{2,3} \end{bmatrix} = \begin{bmatrix} C_{z,1,3} & 0 \\ 0 & C_{z_{,,}3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_{z,1,3} & 0 \\ 0 & D_{z_{,,}3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The interaction between both subsystems is given by:

$$v_{1,2} = z_{2,1} = C_{z,2,1}x_2 + D_{z,2,1}u_2, \quad v_2 = z_{1,2} = C_{z,1,2}x_1 + D_{z,1,2}u_1$$

Using these couplings, the final dynamics become:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & B_{v,1,2}C_{z,2,1} \\ B_{v,2}C_{z,1,2} & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{u,1} & B_{v,1,2}D_{z,2,1} \\ B_{v,2}D_{z,1,2} & B_{u,3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} + \begin{bmatrix} B_{v,1,3} \\ 0 \end{bmatrix} v_{1,3}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} z_{1,3} \\ z_{2,3} \end{bmatrix} = \begin{bmatrix} C_{z,1,3} & 0 \\ 0 & C_{z_{,,}3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_{z,1,3} & 0 \\ 0 & D_{z_{,,}3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Figure B.1 shows the network with subsystems 1 & 2 combined.

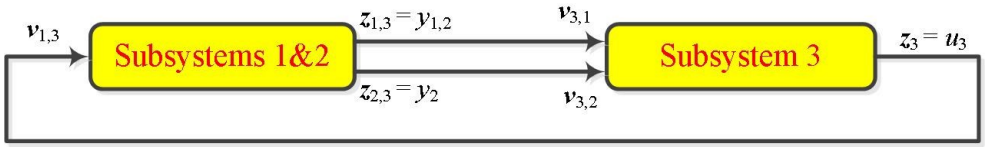



Figure B.1: Subsystems 1 & 2 are combined

Combining subsystems 1 & 3:

The dynamics of subsystem 1 is:

$$\begin{aligned} x_1^+ &= A_1 x_1 + B_{u,1} u_1 + B_{v,1} \begin{bmatrix} v_{1,2} \\ v_{1,3} \end{bmatrix} \\ y_1 &= C_{y,1} x_1 \\ z_1 &= C_{z,1} x_1 + D_{z,1} u_1 = \begin{bmatrix} z_{1,2} \\ z_{1,3} \end{bmatrix} \end{aligned}$$

and for subsystem 3:

$$\begin{aligned} x_3^+ &= A_3 x_3 + B_{u,3} u_3 + B_{v,3} \begin{bmatrix} v_{3,1} \\ v_{3,2} \end{bmatrix} \\ y_3 &= C_{y,3} x_3 \\ z_3 &= C_{z,3} x_3 + D_{z,3} u_3 \end{aligned}$$

Then the combined dynamics would be:

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,1} & 0 \\ 0 & B_{u,3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} + \begin{bmatrix} B_{v,1} & 0 \\ 0 & B_{v,3} \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{1,3} \\ v_{3,1} \\ v_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

$$z_{1,2} = \begin{bmatrix} C_{z,1,2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} D_{z,1,2} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix}$$

Or

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,1} & 0 \\ 0 & B_{u,3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} B_{v,1,2} & B_{v,1,3} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} B_{v,3,1} & B_{v,3,2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{1,3} \\ v_{3,1} \\ v_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

$$z_{1,2} = \begin{bmatrix} C_{z,1,2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} D_{z,1,2} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix}$$

The interaction between both subsystems is given by:

$$v_{1,3} = z_3 = C_{z,3}x_3 + D_{z,3}u_3, \quad v_{3,1} = z_{1,3} = C_{z,1,3}x_1 + D_{z,1,3}u_1$$

Through these couplings, the final dynamics become:

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_1 & B_{v,1,3}C_{z,3} \\ B_{v,3,1}C_{z,1,3} & A_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,1} & B_{v,1,3}D_{z,3} \\ B_{v,3,1}D_{z,1,3} & B_{u,3} \end{bmatrix}\begin{bmatrix} u_1 \\ u_3 \end{bmatrix} +$$

$$\begin{bmatrix} B_{v,1,2} & 0 \\ 0 & B_{v,3,2} \end{bmatrix}\begin{bmatrix} v_{1,2} \\ v_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,1} & 0 \\ 0 & C_{y,3} \end{bmatrix}\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

$$z_{1,2} = \begin{bmatrix} C_{z,1,2} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} + \begin{bmatrix} D_{z,1,2} & 0 \end{bmatrix}\begin{bmatrix} u_1 \\ u_3 \end{bmatrix}$$

The network with subsystems 1 & 3 combined is shown Figure B.2 shows.

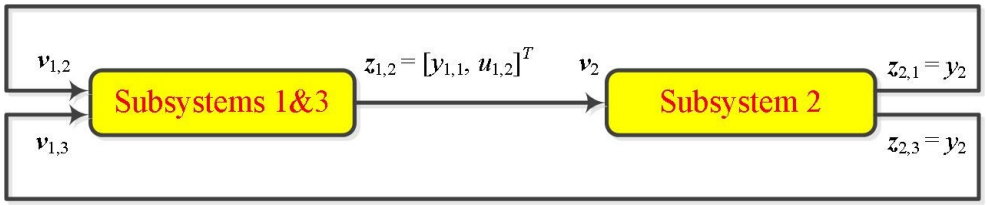



Figure B.2: Subsystems 1 & 3 are combined

Finally combining subsystems 2 & 3:

We have the following dynamics for subsystem 1:

$$\begin{aligned} x_2^+ &= A_2x_2 + B_{u,2}u_2 + B_{v,2}v_2 \\ y_2 &= C_{y,2}x_2 \\ z_2 &= C_{z,2}x_2 + D_{z,2}u_2 \end{aligned}$$

and for subsystem 2:

$$\begin{aligned} x_3^+ &= A_3x_3 + B_{u,3}u_3 + B_{v,3}\begin{bmatrix} v_{3,1} \\ v_{3,2} \end{bmatrix} \\ y_3 &= C_{y,3}x_3 \\ z_3 &= C_{z,3}x_3 + D_{z,3}u_3 \end{aligned}$$

The combined dynamics are given as:

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_2 & 0 \\ 0 & A_3 \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,2} & 0 \\ 0 & B_{u,3} \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} B_{v,2} & 0 \\ 0 & B_{v,3} \end{bmatrix}\begin{bmatrix} v_2 \\ v_{3,1} \\ v_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,2} & 0 \\ 0 & C_{y,3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} z_{2,1} \\ z_3 \end{bmatrix} = \begin{bmatrix} C_{z,2,1} & 0 \\ 0 & C_{z,3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} D_{z,2,3} & 0 \\ 0 & D_{z,,3} \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \end{bmatrix}$$

Or

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_2 & 0 \\ 0 & A_3 \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,2} & 0 \\ 0 & B_{u,3} \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} B_{v,2} & 0 \\ 0 & [B_{v,3,2} \quad B_{v,3,1}] \end{bmatrix}\begin{bmatrix} v_2 \\ v_{3,1} \\ v_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,2} & 0 \\ 0 & C_{y,3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

For subsystems 2 & 3, the coupling is given by:

$$v_{3,2} = z_{2,3} = C_{z,2,3}x_2 + D_{z,2,3}u_2$$

Therefore:

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_2 & 0 \\ B_{v,3,2}C_{z,2,3} & A_3 \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{u,2} & 0 \\ B_{v,3,2}D_{z,2,3} & B_{u,3} \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} B_{v,2} & 0 \\ 0 & B_{v,3,1} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} v_2 \\ v_{3,1} \end{bmatrix}$$

$$\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} C_{y,2} & 0 \\ 0 & C_{y,3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} z_{2,1} \\ z_3 \end{bmatrix} = \begin{bmatrix} C_{z,2,1} & 0 \\ 0 & C_{z,3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} D_{z,2,3} & 0 \\ 0 & D_{z,,3} \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \end{bmatrix}$$

and Figure B.3 shows the combined subsystem in the network.

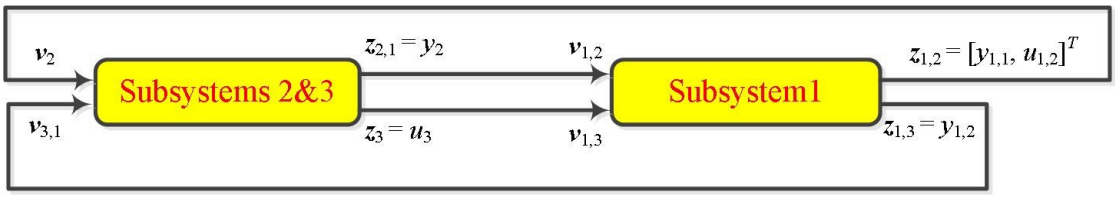



Figure B.3: Subsystems 2 & 3 are combined

## APPENDIX C

## DERIVATIVE OF THE CENTRALIZED COST FUNCTION WITH RESPECT TO THE VARIABLE α

Recall that the local control actions as a function of α at the network level is given by (assuming $\boldsymbol{\mu} = I$):

$$\left[\mathbf{S} + \left(\mathbf{M}_y + \left(\boldsymbol{\alpha}\mathbf{M}_z^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_z^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_y^T\right)^T\right)^T \mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)\right]\mathbf{U}^{\mathbf{QP}}$$

$$= \left[\mathbf{M}_y + \left(\boldsymbol{\alpha}\mathbf{M}_z^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_z^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_y^T\right)^T\right]^T \mathbf{Q}\mathbf{r} -$$

$$\left[\left(\mathbf{M}_y + \left(\boldsymbol{\alpha}\mathbf{M}_z^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_z^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_y^T\right)^T\right)^T \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\right]\mathbf{X_0}$$

Now let:

$$\mathbf{X}_{\boldsymbol{\alpha}} = \left[\mathbf{S} + G_1^T\mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)\right]^{-1}$$

and

$$\mathbf{G}_{\boldsymbol{\alpha}} = G_1^T\mathbf{Q}\mathbf{r} - G_1^T\mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\mathbf{X_0} = G_1^T\left(\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\mathbf{X}_0\right) = G_1^T\mathbf{H}_1$$

where

$$G_1 = \mathbf{M}_y + \left(\boldsymbol{\alpha}\mathbf{M}_z^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_z^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_y^T\right)^T$$

$$\mathbf{H_1} = \left[\mathbf{Q}\mathbf{r} - \mathbf{Q}\left(\mathbf{F}_y + \mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\mathbf{X_0}\right]$$

Then we can write the local control actions at the network level as following:

$$\mathbf{U}^{\mathbf{QP}} = \mathbf{X}_{\boldsymbol{\alpha}}\mathbf{G}_{\boldsymbol{\alpha}}$$

Recall that the centralized cost function is given by:

$$J_{\mathrm{CMPC}} = \mathbf{U}^T\left[\mathbf{S} + \left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)^T\mathbf{Q}\left(\mathbf{M}_y + \mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)\right]\mathbf{U} +$$

$$2\left[\mathbf{X}_0^T\left(\mathbf{F}_y+\mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)^T\mathbf{Q}\left(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)-\mathbf{r}^T\mathbf{Q}\left(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_z\right)\right]\mathbf{U}+C_0$$

where the constant part is given by:

$$C_0=\mathbf{X}_0^T\left(\mathbf{F}_y+\mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)^T\mathbf{Q}\left(\mathbf{F}_y+\mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\mathbf{X}_0-2\mathbf{X}_0^T\left(\mathbf{F}_y+\mathbf{N}_y\mathbf{W}\mathbf{F}_z\right)\mathbf{Q}\mathbf{r}+\mathbf{r}^T\mathbf{Q}\mathbf{r}$$

By introducing $\mathbf{U}^{\mathbf{QP}}$ **(**the suboptimal unconstrained control action at network level**)** into the centralized cost, we can write:

$$J_{\mathrm{CMPC}}=\ \mathbf{L}^T\mathbf{H}_2\mathbf{L}+2\mathbf{H}_3\mathbf{L}+C_0$$

where $\mathbf{L}, \mathbf{H_1}$ and $\mathbf{H_2}$ are given by:

$$\mathbf{L}=\mathbf{X}_{\boldsymbol{\alpha}}\mathbf{G}_{\boldsymbol{\alpha}}$$

$$\mathbf{H}_2=\mathbf{S}+\left(\mathbf{M}_\mathbf{y}+\mathbf{N}_\mathbf{y}\mathbf{W}\mathbf{M}_\mathbf{z}\right)^T\mathbf{Q}\left(\mathbf{M}_\mathbf{y}+\mathbf{N}_\mathbf{y}\mathbf{W}\mathbf{M}_\mathbf{z}\right)$$

$$\mathbf{H}_3=\mathbf{X}_0^T\left(\mathbf{F}_\mathbf{y}+\mathbf{N}_\mathbf{y}\mathbf{W}\mathbf{F}_\mathbf{z}\right)^T\mathbf{Q}\left(\mathbf{M}_\mathbf{y}+\mathbf{N}_\mathbf{y}\mathbf{W}\mathbf{M}_\mathbf{z}\right)-\mathbf{r}^T\mathbf{Q}\left(\mathbf{M}_\mathbf{y}+\mathbf{N}_\mathbf{y}\mathbf{W}\mathbf{M}_\mathbf{z}\right)$$

For derivatives of matrices, we have the following mathematical rules:

i. Let A be a symmetric matrix and $\alpha=\mathrm{x}^T\mathrm{Ax}$ and $\mathrm{x}=\mathrm{f(z)}$ where z is a vector then:

$$\frac{\partial\alpha}{\partial z}=2\mathrm{x}^T\mathrm{A}\frac{\partial\mathrm{x}}{\partial z}$$

ii. Derivative of inverse:

$$\frac{\partial\mathrm{A}^{-1}}{\partial z}=-\mathrm{A}^{-1}\frac{\partial\mathrm{A}}{\partial z}\mathrm{A}^{-1}$$

iii. $\alpha=\mathrm{y}^T\mathrm{x}\ ,\ \ \mathrm{x},\mathrm{y}=\mathrm{f(z)}$ then:

$$\frac{\partial\alpha}{\partial z}=\mathrm{y}^T\frac{\partial\mathrm{x}}{\partial z}+\mathrm{x}^T\frac{\partial\mathrm{y}}{\partial z}$$

Since the matrix $\mathbf{H}_2$ symmetric, we can use rule (i) to write the derivative of $J_{\mathrm{CMPC}}$ with respect to $\boldsymbol{\alpha}$ as following:

$$\frac{\partial J_{\text{CMPC}}}{\partial \boldsymbol{\alpha}} = 2\mathbf{L}^T\mathbf{H}_2\frac{\partial \mathbf{L}}{\partial \boldsymbol{\alpha}} + 2\mathbf{H}_3\frac{\partial \mathbf{L}}{\partial \boldsymbol{\alpha}}$$

Since $\mathbf{L} = \mathbf{X}_{\boldsymbol{\alpha}}\mathbf{G}_{\boldsymbol{\alpha}}$, then the derivative of $\mathbf{L}$ with respect to $\boldsymbol{\alpha}$ can also be driven using rule (iii) as following:

$$\frac{\partial \mathbf{L}}{\partial \boldsymbol{\alpha}} = \mathbf{X}_{\alpha}\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} + \left(\mathbf{G}_{\boldsymbol{\alpha}}^T\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right)^T$$

Using last equations, we have:

$$\frac{\partial J_{\text{CMPC}}}{\partial \boldsymbol{\alpha}} = 2\mathbf{L}^T\mathbf{H}_2\left[\mathbf{X}_{\alpha}\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} + \left(\mathbf{G}_{\boldsymbol{\alpha}}^T\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right)^T\right] + 2\mathbf{H}_3\left[\mathbf{X}_{\alpha}\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} + \left(\mathbf{G}_{\boldsymbol{\alpha}}^T\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right)^T\right]$$

or

$$\frac{\partial J_{\text{CMPC}}}{\partial \boldsymbol{\alpha}} = 2[\mathbf{L}^T\mathbf{H}_2 + \mathbf{H}_3]\left[\mathbf{X}_{\alpha}\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} + \left(\mathbf{G}_{\boldsymbol{\alpha}}^T\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right)^T\right]$$

For $\partial\mathbf{G}_{\boldsymbol{\alpha}}/\partial\boldsymbol{\alpha}$ we can write:

$$\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} = \frac{\partial}{\partial \boldsymbol{\alpha}}\mathbf{G}_{\mathbf{1}}^T\mathbf{H}_1 = \frac{\partial}{\partial \boldsymbol{\alpha}}\left[\mathbf{M}_{\mathbf{y}} + \left(\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}\boldsymbol{\Gamma}^{\boldsymbol{T}}\mathbf{N}_{\mathbf{y}}^T\right)^T\right]^T\mathbf{H}_1$$

$$= \left[\left(\frac{\partial}{\partial \boldsymbol{\alpha}}\left(\underbrace{\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T}_{f_1(\boldsymbol{\alpha})}\underbrace{[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}\boldsymbol{\Gamma}^{\boldsymbol{T}}\mathbf{N}_{\mathbf{y}}^T}_{f_2(\boldsymbol{\alpha})}\right)\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N}_{\mathbf{y}}^T\right)^T\right]^T\mathbf{H}_1$$

$$= \left[\left(\left(\underbrace{\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T}_{f_1(\boldsymbol{\alpha})}\frac{\partial}{\partial \boldsymbol{\alpha}}\underbrace{[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}}_{f_2(\boldsymbol{\alpha})} + \underbrace{[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}}_{f_2(\boldsymbol{\alpha})}\frac{\partial}{\partial \boldsymbol{\alpha}}\underbrace{\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T}_{f_1(\boldsymbol{\alpha})}\right)\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N}_{\mathbf{y}}^T\right)^T\right]^T\mathbf{H}_1$$

$$\Rightarrow \left[\left(\left(\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T\frac{\partial}{\partial \boldsymbol{\alpha}}[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1} + [I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}\mathbf{M}_{\mathbf{z}}^T\right)\boldsymbol{\Gamma}^T\boldsymbol{\mu}\mathbf{N}_{\mathbf{y}}^T\right)^T\right]^T\mathbf{H}_1$$

Now by using the derivative of inverse given by rule (ii), we can differentiate $[I - \boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}$ with respect to $\boldsymbol{\alpha}$ as following:

$$\frac{\partial}{\partial \boldsymbol{\alpha}}[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}=-[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}\frac{\boldsymbol{\partial}}{\boldsymbol{\partial\alpha}}(I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T)[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}$$

$$\Rightarrow \frac{\partial}{\partial \boldsymbol{\alpha}}[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}=-[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}(-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T)[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}$$

$$=[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}(\boldsymbol{\Gamma}^{\mathbf{T}}\boldsymbol{\mu}\mathbf{N}_{\mathbf{z}}^T)[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}$$

Therefore:

$$\frac{\partial \mathbf{G}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}=\Big[\big((\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^T([I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}(\boldsymbol{\Gamma}^T\mathbf{N}_z^T)[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1})+$$

$$[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^T]^{-1}\mathbf{M}_{\mathbf{z}}^T)\boldsymbol{\Gamma}^T\mathbf{N}_{\mathbf{y}}^T\big)^T\Big]^T\mathbf{H}_1$$

Also, using the derivative of inverse for differentiation of $\mathbf{X}_{\boldsymbol{\alpha}}$ with respect to $\boldsymbol{\alpha}$, we would have the following:

$$\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}=\frac{\partial}{\partial \boldsymbol{\alpha}}\big[\mathbf{S}+G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)\big]^{-1}$$

$$=-\big[\mathbf{S}+G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)\big]^{-1}\frac{\partial}{\partial \boldsymbol{\alpha}}\big[\mathbf{S}+G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)\big]\big[\mathbf{S}+G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)\big]^{-1}$$

Note that:

$$\frac{\partial}{\partial \boldsymbol{\alpha}}\big[\mathbf{S}+G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)\big]=\frac{\partial}{\partial \boldsymbol{\alpha}}G_1^T\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)=\frac{\partial}{\partial \boldsymbol{\alpha}}G_1^T\mathbf{H}_4$$

where $\mathbf{H}_4=\mathbf{Q}\big(\mathbf{M}_y+\mathbf{N}_y\mathbf{W}\mathbf{M}_{\mathbf{z}}\big)$

Now:

$$\frac{\partial}{\partial \boldsymbol{\alpha}}G_1^T\mathbf{H}_4=\frac{\partial}{\partial \boldsymbol{\alpha}}\Big[\mathbf{M}_y+\big(\boldsymbol{\alpha}\mathbf{M}_{\boldsymbol{z}}^T[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\boldsymbol{z}}^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_{\mathbf{y}}^T\big)^T\Big]^T\mathbf{H}_4$$

$$=\Big[\frac{\partial}{\partial \boldsymbol{\alpha}}\big(\boldsymbol{\alpha}\mathbf{M}_{\boldsymbol{z}}^T[I-\boldsymbol{\Gamma}^T\boldsymbol{\alpha}\mathbf{N}_{\boldsymbol{z}}^T]^{-1}\boldsymbol{\Gamma}^T\mathbf{N}_{\mathbf{y}}^T\big)^T\Big]^T\mathbf{H}_4$$

However,

$$\frac{\partial}{\partial \boldsymbol{\alpha}}\left(\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^{T}[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}\boldsymbol{\Gamma}^{T}\mathbf{N}_{\mathbf{y}}^{T}\right)^{T}=\Big[\big((\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^{T}([I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}(\boldsymbol{\Gamma}^{T}\mathbf{N}_{z}^{T})[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1})+$$

$$[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}\mathbf{M}_{\mathbf{z}}^{T})\boldsymbol{\Gamma}^{T}\mathbf{N}_{\mathbf{y}}^{T}\big)^{T}\Big]^{T}$$

Thus:

$$\frac{\partial}{\partial \boldsymbol{\alpha}}\left[\mathbf{S}+G_{1}^{T}\mathbf{Q}\left(\mathbf{M}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]=\Big[\big((\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^{T}([I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}(\boldsymbol{\Gamma}^{T}\mathbf{N}_{z}^{T})[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1})+$$

$$[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}\mathbf{M}_{\mathbf{z}}^{T})\boldsymbol{\Gamma}^{T}\mathbf{N}_{\mathbf{y}}^{T}\big)^{T}\Big]^{T}\mathbf{H}_{4}$$

Therefore, we eventually would have:

$$\frac{\partial \mathbf{X}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}=-\left[\mathbf{S}+G_{1}^{T}\mathbf{Q}\left(\mathbf{M}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]^{-1}\Big[\big((\boldsymbol{\alpha}\mathbf{M}_{\mathbf{z}}^{T}([I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}(\boldsymbol{\Gamma}^{T}\mathbf{N}_{z}^{T})[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1})$$

$$+[I-\boldsymbol{\Gamma}^{T}\boldsymbol{\alpha}\mathbf{N}_{\mathbf{z}}^{T}]^{-1}\mathbf{M}_{\mathbf{z}}^{T})\boldsymbol{\Gamma}^{T}\mathbf{N}_{\mathbf{y}}^{T}\big)^{T}\Big]^{T}\mathbf{H}_{4}\left[\mathbf{S}+G_{1}^{T}\mathbf{Q}\left(\mathbf{M}_{\boldsymbol{y}}+\mathbf{N}_{\boldsymbol{y}}\mathbf{W}\mathbf{M}_{\mathbf{z}}\right)\right]^{-1}$$

This ends the derivation of the derivative of $J_{\mathrm{CMPC}}$ with respect to $\boldsymbol{\alpha}$.

## APPENDIX D

## MORE DETAILS FOR THE COST FUCNTIONS USED IN CHAPTER 5

In this appendix, more details are given for the cost functions that have been used in the control of the UBO building HVAC system. The economic cost functions given here are developed by Bay C. [100]. However, we modified the costs for MPC applications.

### Zone Cost Functions:

Recall that the cost function used for designing the local LC-DMPC controllers at the zone subsystem levels has the following form (without the cost for downstream outputs):

$$J_i = \sum_{j=1}^{N_p} (e_i^T(k+j)\mathbb{Z}_i(k+j))$$

where $e_i = T_{room,i} - T_{desired,i}$.

The variable $\mathbb{Z}_i$ is given as:

$$\mathbb{Z}_i(T_{room,i}, \$) = \frac{\partial PMV_i(T_{room,i})}{\partial T_{room,i}} \cdot \frac{\partial LOP(PMV_i(T_{room,i}))}{\partial PMV_i(T_{room,i})} \cdot \frac{\partial RO_{cost}(\$)}{\partial LOP(PMV_i(T_{room,i}))} \qquad \text{(D.1)}$$

where

$PMV_i(T_{room,i})$ is the Predicted Mean Vote (PMV) equation,

$LOP\left(PMV_i\left(T_{room,i}\right)\right)$ (Loss Of Productivity) is a function of the PMV given as:

$$LOP\left(PMV_i\left(T_{room,i}\right)\right) = b_1 \cdot PMV_i\left(T_{room,i}\right)^2 + b_2 \cdot PMV_i\left(T_{room,i}\right) + b_3 \qquad \text{(D.2)}$$

$RO_{cost}(\$)$ is the Room Operating cost that is given by:

$$RO_{cost}(\$) = LOP(PMV_i(T_{room,i})) \cdot \left[\frac{p_{year} \cdot t_s}{52wks \cdot 40hrs}\right] \qquad \text{(D.3)}$$

$p_{year}$ is the yearly payment of the people who occupy room *i* in dollar, and

$t_s$ is the sampling time (in hour).

In this work, the term $\left[\frac{p_{year} \cdot t_s}{52wks \cdot 40hrs}\right]$ is referred to as the productivity cost.

The productivity of the occupancies in the zones due to the HVAC system operation is measured through the quantity of Loss Of Productivity (LOP). Through LOP, we related the productivity level of the occupancies (or employees) with the comfort level inside the zones which consequently is related to performance of the HVAC system.

The equation of the PMV is nonlinear in term of the room temperature $T_{room,i}$ which givens a nonlinear and non-convex MPC problem. However, a computationally traceable linear version of the PMV for MPC applications is proposed by Cigler at al. [101] as shown below:

$$PMV_i\left(T_{room,i}(k)\right) = (0.303 \cdot e^{(-0.036 \cdot M)} + 0.028) \cdot L_i(k),$$

$$L_i(k) = (M_i - W_i) - 3.05 \cdot 10^{-3} \cdot \left(5733 - 6.99(M_i - W_i) - p_{a,i}(k)\right) -$$

$$-0.24 \cdot \left((M_i - W_i) - 58.15\right) - 17.10^{-5} \cdot M_i \cdot \left(5867 - p_{a,i}(k)\right) -$$

$$-0.0014 \cdot M_i \cdot \left(34 - T_{room,i}(k)\right) - T_{x,i}(k),$$

$$p_{a,i}(k) = k_{pa,i} \cdot T_{room,i}(k) + q_{pa,i},$$

$$T_{x,i}(k) = 3.96.10^{-8} \cdot f_{cl,i} \cdot \left(T'_{cl,i}(k) - \overline{T}'_{cl,i}(k)\right) + f_{cl,i} \cdot h_{c,i} \cdot \left(T_{cl,i}(k) - T_{room,i}(k)\right),$$

$$T'_{cl,i}(k) = T_{cl,i,0} + 273.16 + 4 \cdot \left(T_{cl,i,0} + 273.16\right)^3 \cdot \left(T_{cl,i}(k) - T_{cl,i,0}\right),$$

$$\overline{T}'_{r,i}(k) = \overline{T}_{r,i,0} + 273.16 + 4 \cdot \left(\overline{T}_{r,i,0} + 273.16\right)^3 \cdot \left(\overline{T}_{r,i}(k) - \overline{T}_{r,i,0}\right),$$

$$T_{cl,i}(k) = 35.7 - 0.028(M_i - W_i) - I_{cl,i} \cdot T_{x,i}(k),$$

$h_{c,i} = 12.1\sqrt{v_{ar,i}}$, and

$$f_{cl,i} = \begin{cases} 1.00 + 1.290 \cdot I_{cl,i} & \text{if } I_{cl,i} \leq 0.078 \\ 1.05 + 0.645 \cdot I_{cl,i} & \text{if } I_{cl,i} > 0.078 \end{cases}.$$

and the reader is referred to [101] for the definitions and values of all quantities.

To write the final linear equation of the PMV, let us define the following:

$$a = 3.96 \cdot 10^{-8} \cdot f_{cl,i},\ b = f_{cl,i} \cdot h_{c,i},\ c = 35.7 - 0.028 \cdot (M_i - W_i),$$

$$d = T_{cl,i,0} + 273.16,\ e = 4 \cdot (T_{cl,i,0} + 273.16)^3,\ g = 3.05 \cdot 10^{-3}$$

$$r = 5733 - 6.99 \cdot (M_i - W_i),\ s = 0.42 \cdot (M_i - W_i - 58.15),\ v = 1.7 \cdot 10^{-5} \cdot M_i$$

$$w = 0.0014 \cdot M_i,\ z = 4 \cdot \left(\bar{T}_{r,i,0} + 273.16\right)^3.$$

Then the $PMV_i\left(T_{room,i}(k)\right)$ can be written as:

$$PMV_i\left(T_{room,i}(k)\right) = (0.303e^{-0.036M} + 0.028) \cdot$$

$$\left[T_{room,i}(k)\left((g+v) \cdot k_{pa,i} + w + \frac{a \cdot z + b}{1 + a \cdot e \cdot I_{cl,i} + b \cdot I_{cl,i}}\right) + \right.$$

$$\left((M_i - W_i) + (g+v) \cdot q_{pa,i} - g \cdot r - s - 5867 \cdot v - 34 \cdot w - \right.$$

$$\left.\left.\frac{a \cdot d + a \cdot e \cdot c - a \cdot e \cdot \bar{T}_{r,i,0} - a \cdot \left(\bar{T}_{r,i,0} + 273.16 - z \cdot \bar{T}_{r,i,0}\right) + b \cdot c}{1 + a \cdot e \cdot I_{cl,i} + b \cdot I_{cl,i}}\right)\right]$$

where it is assumed that $\bar{T}_{r,i}(k) = T_{room,i}(k)$.

Also let:

$$n_1 = (0.303e^{-0.036M} + 0.028),$$

$$n_2 = \left((g+v) \cdot k_{pa,i} + w + \frac{a \cdot z + b}{1 + a \cdot e \cdot I_{cl,i} + b \cdot I_{cl,i}}\right),$$

$$n_3 = \Big((M_i - W_i) + (g+v)\cdot q_{pa,i} - g\cdot r - s - 5867\cdot v - 34\cdot w - \frac{a\cdot d + a\cdot e\cdot c - a\cdot e\cdot \bar{T}_{r,i,0} - a\cdot\left(\bar{T}_{r,i,0} + 273.16 - z\cdot \bar{T}_{r,i,0}\right) + b\cdot c}{1 + a\cdot e\cdot I_{cl,i} + b\cdot I_{cl,i}}\Big)$$

Then the linear approximated PMV equation can be written as:

$$PMV_i\left(T_{room,i}(k)\right) = n_1[n_2\cdot t_a + n_3] = \underbrace{n_1 n_2}_{m_1} T_{room,i}(k) + \underbrace{n_1 n_3}_{m_2}$$

or

$$PMV_i\left(T_{room,i}(k)\right) = m_1\cdot T_{room,i}(k) + m_2 \tag{D.4}$$

Now from the equations: D.2, D.3, and D.4, the following can be calculated:

$$\frac{\partial PMV_i(T_{room,i})}{\partial T_{room,i}} = m_1 \tag{D.5}$$

$$\frac{\partial LOP(PMV_i(T_{room,i}))}{\partial PMV_i(T_{room,i})} = 2\cdot b_1\cdot PMV_i\left(T_{room,i}\right) + b_2 \tag{D.6}$$

and

$$\frac{\partial RO_{cost}(\$)}{\partial LOP(PMV_i(T_{room,i}))} = \frac{p_{year}\cdot t_s}{52wks\cdot 40hrs} \tag{D.7}$$

Therefore, through (D.5) – (D.7), equation (D.1) becomes:

$$\mathbb{Z}_i\left(T_{room,i},\$\right) = m_1\cdot\left(2\cdot b_1\cdot PMV_i\left(T_{room,i}\right) + b_2\right)\cdot\frac{p_{year}\cdot t_s}{52wks\cdot 40hrs} \tag{D.8}$$

where

$$b_1 = 6.958, \qquad b_2 = 0.9628$$

To further simplify (D.8) and write down everything in term of $T_{room,i}$, let:

$$n_4 = \left[\frac{p_{year}\cdot t_s}{52\text{wKs}\cdot 40\text{hrs}}\right]$$

Then:

$$\mathbb{Z}_i\left(T_{room,i}, \$\right) = \underbrace{2 \cdot m_1 \cdot b_1 \cdot n_4}_{m_3} \cdot PMV_i\left(T_{room,i}\right) + \underbrace{m_1 \cdot b_2 \cdot n_4}_{m_4}$$

$$= m_3 \cdot PMV_i\left(T_{room,i}\right) + m_4$$

However,

$$PMV_i\left(T_{room,i}(k)\right) = m_1 \cdot T_{room,i}(k) + m_2$$

Then:

$$\mathbb{Z}_i\left(T_{room,i}, \$\right) = m_3 \cdot \left[m_1 \cdot T_{room,i}(k) + m_2\right] + m_4$$

$$= \underbrace{m_3 \cdot m_1}_{\phi_i} \cdot T_{room,i}(k) + \underbrace{m_3 \cdot m_2 + m_4}_{\varphi_i}$$

Finally we have:

$$\mathbb{Z}_i\left(T_{room,i}, \$\right) = \phi_i \cdot T_{room,i}(k) + \varphi_i \qquad \text{(D.9)}$$

Equation (D.9) is given as (5.5) in chapter 5.

To validate the linear approximation of the PMV equation with the fully nonlinear expression, Figure D.1 was drawn for some temperatures using the typical office activity data used for simulations in chapter V.

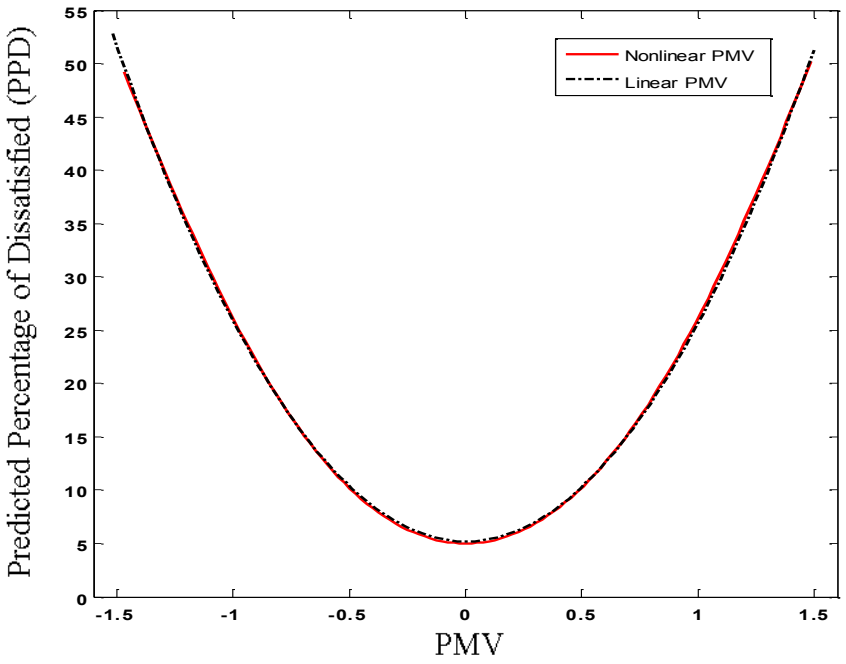



Figure D.1: Linear and nonlinear PMV for temperatures 16.9 to 28.9 ℃

**AHU Optimizer Cost Functions:**

Recall that the cost functions used in the AHU optimizer are summarized as following (without the cost of downstream outputs) and for more details for the used values the reader is referred to [100]:

$$J_{fan} = \sum_{j=0}^{N_p-1}\left[u_{fan}^{T}(k+j)\cdot\delta_{fan}(k+j)\right] \tag{D.10}$$

and

$$J_{T_{AHU}} = \sum_{j=0}^{N_p-1}\left[u_{T_{AHU}}^{T}(k+j)\cdot\delta_{T_{AHU}}(k+j)\right] \tag{D.11}$$

where

$u_{fan} = P_{EDS}$ and $u_{T_{AHU}} = T_{AHU}$, and

$\delta_{fan}$ & $\delta_{T_{AHU}}$ are given as below:

$$\delta_{fan} = \frac{0.1175 \cdot q_{total} \cdot C_{elec} \cdot t_s}{1000 \cdot \mu_m \cdot \mu_b \cdot \mu_f}$$

$$u_{T_{AHU}}^{T} \delta_{T_{AHU}} = (3.021 \cdot T_{AHU}^2 - 109.4 \cdot T_{AHU} + 1002) \cdot q_{max} \cdot \rho . c . \Delta T_{water} . t_s \cdot \frac{\$241}{3682 mmBtu}$$

where

$P_{EDS}$: is the end static pressure (in. water),

$T_{AHU}$: AHU discharge air temperature ($C^0$),

$q_{total}$: is the total air flow rate flows through the AHU (cfm),

$C_{elec} = 0.087\ \$/(k\text{W} \cdot hr)$: Cost of electricity,

$t_s$: Sampling time (in hour),

$\mu_m = 0.902,\ \mu_b = 0.86, \& \mu_f = 0.70$: are the efficiency of motor, belt, and fan blade, respectively,

$q_{max} = 0.0034[m^3/sec]$: Maximum chilled water flow provided at 100% chilled water valve position,

$\rho = 1000[kg/m^3]$: ensity of water,

$c = 4.187\ [kJ/kg/C^0]$: Specific heat of water,

$\Delta T_{water}[C^0] = T_{CWR} - T_{CWS}$: Difference in chilled water temperature,

$T_{CWR}[C^0] = (47 - 32)/1.8$: Return chilled water temperature,

$T_{CWS}[C^0] = (42 - 32)/1.8$: Supply chilled water temperature.

Figure D.2 shows the approximation of the relationship between the AHU discharge air temperature and the position of the chilled water control valve at the UBO building where the approximated relation is given as:

$$\mathrm{CWVO}(T_{AHU}) = (3.021 \cdot T_{AHU}^2 - 109.4 \cdot T_{AHU} + 1002)$$

where CWVO is the chilled water valve position.

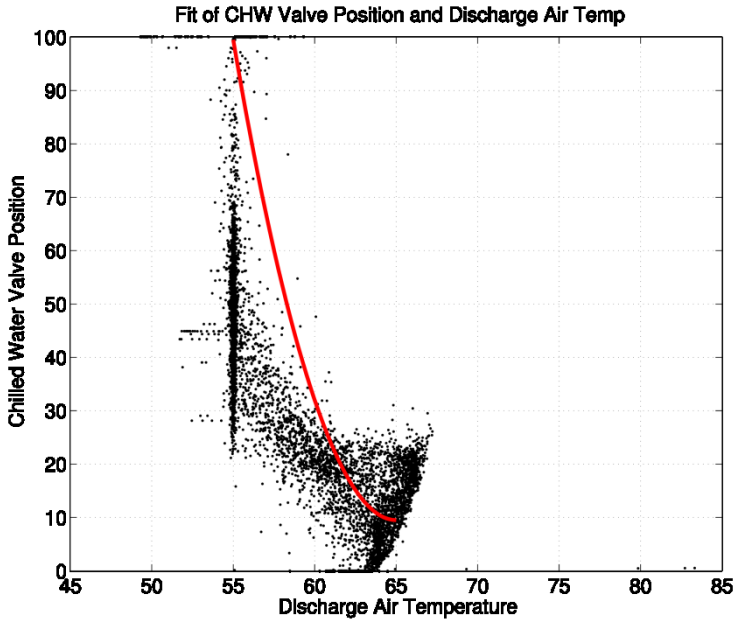



Figure D.2: AHU discharge air temptation and control valve position in the UBO building - reprinted with permission from [100]

With the introduced definitions for the AHU discharge air temperature cost function, the cost functions in (5.9) and (5.10) for $N_p = 1$ can be reformulated as quadratic cost function as followings:

For (5.9), if the following equation is used to approximate the relation between the total flow rate and the $P_{EDS}$:

$$flow\ rate_{total}(k) = 5149 \cdot P_{EDS}(k) - 979.8$$

Then $\delta_{fan}$ can be written as:

$$u_{fan}^T(k) \cdot \delta_{fan}(k) = a_{fan} \cdot P_{EDS}^2(k) + b_{fan} \cdot P_{EDS}(k)$$

where

$$a_{fan} = 5149\,\frac{0.1175 \cdot C_{elec} \cdot t_s}{1000 \cdot \mu_m \cdot \mu_b \cdot \mu_f}$$

$$b_{fan} = -979.8\,\frac{0.1175 \cdot C_{elec} \cdot t_s}{1000 \cdot \mu_m \cdot \mu_b \cdot \mu_f}$$

And by considering the zone cost sensitivities w.r.t $P_{EDS}$, then we can have the following:

$$J_{fan}(k) = H_{fan} \cdot P_{EDS}^2(k) + f_{T_{AHU}}(k) \cdot P_{EDS}(k)$$

where

$$H_{fan} = a_{fan}, \text{ and } \; f_{fan}(k) = b_{fan} + \psi_{fan}(k)$$

And for (5.10), let:

$$a_{T_{AHU}} = 3.02 \cdot q_{max} \cdot \rho . c . \Delta T_{water} . t_s \cdot \frac{4019}{263 mmBtu} \cdot 0.0034121416$$

$$b_{T_{AHU}} = -109.4 \cdot q_{max} \cdot \rho . c . \Delta T_{water} . t_s \cdot \frac{4019}{263 mmBtu} \cdot 0.0034121416$$

$$c_{T_{AHU}} = 1002 \cdot T_{AHU} \cdot q_{max} \cdot \rho . c . \Delta T_{water} . t_s \cdot \frac{4019}{263 mmBtu} \cdot 0.0034121416$$

Thus:

$$u_{T_{AHU}}^T \delta_{T_{AHU}} = a_{T_{AHU}} T_{AHU}^2 + b_{T_{AHU}} T_{AHU} + c_{T_{AHU}}$$

Therefore, if we consider the zone cost sensitivities w.r.t $T_{AHU}$, then one can write $J_{T_{AHU}}(k)$ as:

$$J_{T_{AHU}}(k) = H_{T_{AHU}} \cdot T_{AHU}^2(k) + f_{T_{AHU}}(k) \cdot T_{AHU}(k) + c_{T_{AHU}}$$

where

$$H_{T_{AHU}} = a_{T_{AHU}}, \text{ and } \; f_{T_{AHU}}(k) = b_{T_{AHU}} + \psi_{T_{AHU}}(k)$$