SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

Deliverable Number:	D2.2
Deliverable Type:	Report
Contractual Date of Delivery:	August 31, 2009
Actual Date of Delivery:	**August 31, 2009**
Title of Deliverable:	**Report on the final assessment of the methods for the definition of the control architecture and preliminary report on extended algorithms coping with structural constraints, changes, and multi-level models**
Dissemination level:	Restricted to other programme participants
Workpackage contributing to the Deliverable:	WP2
WP Leader:	POLIMI
Partners:	TUD, EDF, POLIMI, UNC, SUPELEC, INOCSA
Author:	R. Scattolini

Table of contents

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@dcsc.tudelft.nl

Project web site: `http://www.ict-hd-mpc.eu/`

Executive Summary

This report describes the research activity in the Seventh Framework Programme, Theme 3 "Information and Communication Technologies", STREP research project **Hierarchical and Distributed Model Predictive Control of Large Scale Systems- HD-MPC**, focusing on WP2 - "Definition of the hierarchical architecture for control design" - task 2.2: "Definition of the control architecture", task 2.3: "Extension of the control architecture", task 2.4: "Multi-level models". The report is organized in five main chapters:

- Chapter 1 summarizes four main cases where hierarchical control can be used, i.e. cascade systems, multi time scale systems, or system described at different levels of abstraction (see also Deliverable D2.1), which have been considered in the report.
- Chapter 2 presents an approach to the design of hierarchical control systems with Model Predictive Control (MPC). A mathematical formulation of the problem is given by considering a three layers structure and different time scales.
- Chapter 3 describes two different communications protocol among the layers which have been defined to coordinate the control actions computed at the different levels.
- Chapter 4 shows how the proposed algorithmic solution fits with the main structures adopted for hierarchical control and described in Chapter 1.
- Chapter 5 is devoted to present a very simple worked example useful to illustrate how the proposed methodology can be adapted to consider the different cases discussed in Chapter 1.

Chapter 1

Introduction

In Deliverable D2.1 it has been extensively discussed how hierarchical structures in control can be used to cope with a number of industrial problems. Specifically, the following four relevant cases have been identified:
Case 1. When the plant under control is composed by many process units, it is advisable to design a high level regulator optimizing the overall performance and coordinating the underlying units. In practice, the controller can be organized according to a cascade structure where the high level regulator computes the reference signals for the systems at the lower level, which in turn are equipped with local controllers and provide the higher level with the required (control) actions. Additional information may be transmitted from the lower level to the upper one to guarantee that the coordinator provides the lower level with feasible references.
Case 2. The synthesis problem for systems characterized by significantly different dynamic behaviors, often called singularly perturbed [2], has been widely studied in the literature (see [1] for an industrial example). The control action is usually due to two main contributions: a regulator working at low frequency and accounting for the slow dynamics produces both the value of the control variables with a long term effect and the references (inputs and states) for another regulator working at an higher frequency. In turn, the latter regulator computes the values of the manipulated variables with a short term effect, so as to obtain a tighter control action and to reject disturbances.
Case 3. A hierarchical control scheme widely employed is the one adopted in the context of plantwide control [4],[3] where different models of the system under control are used to design regulators working at slow and high frequencies. At the higher level of the hierarchy a simplified model is used to compute the reference values for the lower level by minimizing a cost function usually based on economic considerations. At the lower level a dynamic model is used for the synthesis of a regulator (typically designed with MPC) guaranteeing the proper effective control action.
Case 4. A simplified version of the scheme described in case 3 consists of a top layer with a *static* model of the system used to fix the set-point for the lower level controllers.

Although studies on the design of hierarchical control systems can be traced back to the early ‘70s (see, e.g., [8]) and have received attention for many years [4], the problem of designing an overall control system based on MPC and guaranteeing some fundamental properties, such as stability and robustness, has not been fully solved yet and only partial results are available, see [11, 9].
In this direction, this report gives a mathematical formulation of the control design problem and presents a possible solution with MPC that provides a unifying framework for the four cases listed above. Specifically, the system under control is assumed to be described by a three layers structure, where each layer is characterized by a different dynamic behavior. For any layer, starting from the

highest one which corresponds to the slowest representation of the system, an MPC problem is formulated and its solution is passed to the lower layer until the procedure is completed. The results contained in this report have been also described in [10].

Chapter 2

Problem definition and control algorithm

The problem is formulated for the case of three layers systems. Nevertheless, the proposed approach can be modified in a natural way to include n layers systems with $n \geq 2$.
Consider a system with different dynamic behaviors and described by the following set of state equations:

$$\begin{cases} x_s^f(k_f+1) = f_s^f\Big(x_s^f(k_f), x_m^f(k_f), x_f^f(k_f), \\ \qquad u_s^f(k_f), u_m^f(k_f), u_f^f(k_f)\Big) & \text{(2.1a)} \\ x_m^f(k_f+1) = f_m^f\Big(x_s^f(k_f), x_m^f(k_f), x_f^f(k_f), \\ \qquad u_s^f(k_f), u_m^f(k_f), u_f^f(k_f)\Big) & \text{(2.1b)} \\ x_f^f(k_f+1) = f_f^f\Big(x_s^f(k_f), x_m^f(k_f), x_f^f(k_f), \\ \qquad u_s^f(k_f), u_m^f(k_f), u_f^f(k_f)\Big) & \text{(2.1c)} \end{cases}$$

where the *high level* state and input variables x_s^f and u_s^f are associated to a “slow” dynamics, the *middle level* variables x_m^f and u_m^f to a “medium” dynamics and the *low level* variables x_f^f and u_f^f to a “fast” dynamics. The time index k_f is related to a base fast time scale, where the fastest dynamic behaviors are adequately represented. Then, the superscript f in the previous symbology means that the state and input variables, as well as the transition functions, are associated to the fast time scale.
The state and control inputs are required to belong to given compact sets containing the origin as interior point, i.e., $x_s^f \in \mathscr{X}_s$, $u_s^f \in \mathscr{U}_s$, $x_m^f \in \mathscr{X}_m$, $u_m^f \in \mathscr{U}_m$, $x_f^f \in \mathscr{X}_f$, $u_f^f \in \mathscr{U}_f$.

In order to develop the multiscale MPC algorithm presented below, it is also worth introducing a “medium” and a “slow” time scale by defining the time index of the medium time scale $k_m \in \mathbb{N}$ so that $k_f = \nu_m k_m$ (for some fixed positive integer ν_m), and the time index of the slow time scale $k_s \in \mathbb{N}$ so that $k_m = \nu_s' k_s$ (for some fixed positive integer ν_s'), thus $k_f = \nu_s k_s$ with $\nu_s = \nu_m \nu_s'$.

The following assumption, concerning the update of the control variables, reflects the different control objectives at different time scales:

Assumption 1 *The control variable u_m^f is allowed to vary at every medium sampling period, i.e., $u_m^f(\nu_m k_m + i) = u_m^f(\nu_m k_m)$, $i = 0, \ldots, \nu_m - 1$. Such a common value is denoted by $u_m(k_m)$, thus defining a control signal in the medium time scale.*
The control variable u_s^f is allowed to vary at every slow sampling period, i.e., $u_s^f(\nu_s k_s + i) = u_s^f(\nu_s k_s)$, $i = 0, \ldots, \nu_s - 1$: such a common value is denoted by $u_s(k_s)$, thus defining a control signal in the slow

time scale. Obviously, this means that u_s^f *is also constant over the medium sampling periods, hence* $u_s^f(\nu_m k_m + i) = u_s^f(\nu_m k_m)$, $i = 0, \ldots, \nu_m - 1$ $\big($*such a common value is denoted by* $u_s^m(k_m)\big)$ *as well as* $u_s(k_s) = u_s^m(\nu_s' k_s + i) = u_s^m(\nu_s' k_s)$, $i = 0, \ldots, \nu_s' - 1$. ◇

We let $x_*^m(k_m) = x_*^f(\nu_m k_m)$ and $x_*^s(k_s) = x_*^f(\nu_s k_s)$ be the sampling of the state variables according to the medium and to the slow time scale, respectively (where "$*$" is f, m or s, in turn). To simplify the notation, we let $x_f(k_f) = x_f^f(k_f)$, $u_f(k_f) = u_f^f(k_f)$, $x_m(k_m) = x_m^m(k_m)$ and $x_s(k_s) = x_s^s(k_s)$.

According to Assumption 1 and to the notation introduced above, the dynamics of x_s^m and x_m^m can easily (although implicitly, in the case of nonlinear systems) be described in terms of suitable functions f_s^m and f_m^m as follows:

$$\begin{cases} x_s^m(k_m+1) = f_s^m\big(x_s^m(k_m), x_m(k_m), x_f^m(k_m), \\ \qquad\qquad u_s^m(k_m), u_m(k_m), U_f^m(k_m)\big) & \text{(2.2a)} \\ x_m(k_m+1) = f_m^m\big(x_s^m(k_m), x_m(k_m), x_f^m(k_m), \\ \qquad\qquad u_s^m(k_m), u_m(k_m), U_f^m(k_m)\big), & \text{(2.2b)} \end{cases}$$

with

$$U_f^m(k_m) = \big[u_f(\nu_m k_m)\ u_f(\nu_m k_m + 1)\ \cdots\ u_f\big(\nu_m(k_m+1) - 1\big)\big].$$

In a similar way, the dynamics of x_s^s is described in terms of a suitable function f_s^s by

$$x_s(k_s+1) = f_s^s\big(x_s(k_s), x_m^s(k_s), x_f^s(k_s), u_s(k_s), U_m^s(k_s), U_f^s(k_s)\big), \tag{2.3}$$

with

$$\begin{cases} U_m^s(k_s) = \big[u_m(\nu_s' k_s)\ u_m(\nu_s' k_s + 1)\ \cdots\ u_m\big(\nu_s'(k_s+1) - 1\big)\big] \\ U_f^s(k_s) = \big[u_f(\nu_s k_s)\ u_f(\nu_s k_s + 1)\ \cdots\ u_f\big(\nu_s(k_s+1) - 1\big)\big]. \end{cases}$$

With slight abuse of notation, we write $U_f^m(k_m) = \bar{u}_f$ to mean that $u_f(\nu_m k_m + i) = \bar{u}_f\ \forall i = 0, \ldots, \nu_m - 1$. Analogous convention is set for the vectors U_m^s and U_f^s.

MPC problem at the slow time scale

Since the dynamics are supposed to be increasingly faster at the middle and low levels, at any long sampling time the control design for the higher level is carried out under the assumption that, along the time interval $[k_s, k_s+1[$, the lower levels are at the steady state (say, the state and control variables take suitable constant values $\bar{x}_m$, $\bar{u}_m$, $\bar{x}_f$ and $\bar{u}_f$). This position motivates the following

Definition 1 *A 6-tuple* $(x_s, u_s, \bar{x}_m, \bar{u}_m, \bar{x}_f, \bar{u}_f) \in \mathscr{X}_s \times \mathscr{U}_s \times \mathscr{X}_m \times \mathscr{U}_m \times \mathscr{X}_f \times \mathscr{U}_f$ *such that*

$$\begin{cases} \bar{x}_m = f_m^m(x_s, \bar{x}_m, \bar{x}_f, u_s, \bar{u}_m, \bar{u}_f) \\ \bar{x}_f = f_f^f(x_s, \bar{x}_m, \bar{x}_f, u_s, \bar{u}_m, \bar{u}_f) \end{cases}$$

is said to be s-admissible.
A 4-tuple $(x_s, u_s, \bar{x}_m, \bar{u}_m) \in \mathscr{X}_s \times \mathscr{U}_s \times \mathscr{X}_m \times \mathscr{U}_m$ *such that* $\exists(\bar{x}_f, \bar{u}_f) \in \mathscr{X}_f \times \mathscr{U}_f$ *so that the corresponding 6-tuple* $(x_s, u_s, \bar{x}_m, \bar{u}_m, \bar{x}_f, \bar{u}_f)$ *is s-admissible is said to be* feasible. ◇

Thus, the s-admissibility of a 6-tuple is related to the request that $\bar{x}_m$ and $\bar{x}_f$ be equilibrium states according to their "natural" dynamics (i.e., f_m^m and f_f^f, respectively).

Assumption 2

1. *$\forall x_s \in \mathscr{X}_s$, there exists at least one 5-tuple $(u_s, \bar{x}_m, \bar{u}_m, \bar{x}_f, \bar{u}_f) \in \mathscr{U}_s \times \mathscr{X}_m \times \mathscr{U}_m \times \mathscr{X}_f \times \mathscr{U}_f$ so that the corresponding 6-tuple $(x_s, u_s, \bar{x}_m, \bar{u}_m, \bar{x}_f, \bar{u}_f)$ is s-admissible.*

2. *If the 4-tuple $(x_s, u_s, \bar{x}_m, \bar{u}_m) \in \mathscr{X}_s \times \mathscr{U}_s \times \mathscr{X}_m \times \mathscr{U}_m$ is feasible, then there exists a unique pair $(\bar{x}_f, \bar{u}_f) \in \mathscr{X}_f \times \mathscr{U}_f$ so that the corresponding 6-tuple $(x_s, u_s, \bar{x}_m, \bar{u}_m, \bar{x}_f, \bar{u}_f)$ is s-admissible.* $\diamond$

The variable $u_s(k_s)$ is computed by solving an MPC optimization problem and by adopting the Receding Horizon paradigm. That is, letting $l_s(\cdot,\cdot,\cdot,\cdot)$ and $v_s(\cdot)$ be positive cost functions and $N_s > 0$ an integer, the optimization problem

$$\min_{\substack{u_s(k_s+i),\ i=0,\ldots,N_s-1 \\ \bar{x}_m(k_s+i),\bar{u}_m(k_s+i),\ i=0,\ldots,N_s-1}} J_s\big(x_s(k_s)\big), \tag{2.4}$$

where

$$J_s\big(x_s(k_s)\big) = \sum_{i=0}^{N_s-1} l_s\big(x_s(k_s+i), u_s(k_s+i), \bar{x}_m(k_s+i), \bar{u}_m(k_s+i)\big) + v_s\big(x_s(k_s+N_s)\big),$$

is considered subject to the following constraints:

- the system dynamics (2.3) with,

$$\forall i = 0,\ldots,N_s-1, \quad \begin{cases} x_m^s(k_s+i) = \bar{x}_m(k_s+i) \\ x_f^s(k_s+i) = \bar{x}_f(k_s+i) \\ U_m^s(k_s+i) = \bar{u}_m(k_s+i) \\ U_f^s(k_s+i) = \bar{u}_f(k_s+i), \end{cases}$$

  where the 4-tuple $\big(x_s(k_s+i), u_s(k_s+i), \bar{x}_m(k_s+i),\ \bar{u}_m(k_s+i)\big)$ is feasible and, according to Assumption 2.2, $\big(\bar{x}_f(k_s+i),\ \bar{u}_f(k_s+i)\big) \in \mathscr{X}_f \times \mathscr{U}_f$ is the unique pair such that the 6-tuple $\big(x_s(k_s+i), u_s(k_s+i), \bar{x}_m(k_s+i),\ \bar{u}_m(k_s+i), \bar{x}_f(k_s+i), \bar{u}_f(k_s+i)\big)$ is s-admissible;

- the feasibility constraint $x_s(k_s+N_s) \in \mathscr{X}_s$.

Then, according to the RH paradigm, only the first computed value $u_s(k_s)$ is applied and the overall procedure is repeated at the new slow sampling period. Indeed, besides computing the optimal control sequence at the high level, also the desired reference for the state and the input variables $\big(\bar{x}_m(k_s), \bar{u}_m(k_s)\big)$ at the middle level is returned.

Remark 1 *Notice that, thanks to Assumption 2.2, the input and state variables of the low level are not explicitly involved in the optimization* (2.4) *at the high level.* $\diamond$

Remark 2 *If the current state $x_m^s(k_s)$ of the middle level is available to the controller at the high level, in order that a sensible reference be provided to the middle level, one can add the constraint $\|x_m^s(k_s) - \bar{x}_m(k_s)\| \le \varepsilon_m$ (for suitably small $\varepsilon_m \ge 0$) to the optimization problem* (2.4). $\diamond$

MPC problem at the medium time scale
Now consider the medium time scale. In view of the solution of the optimization problem at the high level, the goal is to track given reference values $(\overline{x}_m, \overline{u}_m)$ of the state and input variables. Moreover, similarly to the problem at the slow time scale, since the dynamics at the low level is supposed to be the fastest one, the control problem at the middle level is solved under the assumption that, along the time interval $[k_m, k_m+1[$, the low level is at the steady state (say, the state and control variables take suitable constant values $\overline{x}_f$ and $\overline{u}_f$). This position motivates the following

Definition 2 *A 6-tuple* $(x_s, u_s, x_m, u_m, \overline{x}_f, \overline{u}_f) \in \mathscr{X}_s \times \mathscr{U}_s \times \mathscr{X}_m \times \mathscr{U}_m \times \mathscr{X}_f \times \mathscr{U}_f$ *such that*

$$\overline{x}_f = f_f^f(x_s, x_m, \overline{x}_f, u_s, u_m, \overline{u}_f)$$

is said to be m-admissible. ◇

Assumption 3 *For any tern* $(x_s, u_s, x_m) \in \mathscr{X}_s \times \mathscr{U}_s \times \mathscr{X}_m$*, there exists at least one tern* $(u_m, \overline{x}_f, \overline{u}_f) \in \mathscr{U}_m \times \mathscr{X}_f \times \mathscr{U}_f$ *so that the corresponding 6-tuple* $(x_s, u_s, x_m, u_m, \overline{x}_f, \overline{u}_f)$ *is m-admissible.* ◇

The variable $u_m(k_m)$ is computed by solving an MPC optimization problem and by adopting the Receding Horizon paradigm. That is, letting $l_m(\cdot,\cdot,\cdot,\cdot)$ and $v_m(\cdot)$ be positive cost functions, $N_m > 0$ be an integer and

$$k_s = \left\lfloor \frac{k_m}{\nu_s'} \right\rfloor,$$

we consider the optimization problem

$$\min_{\substack{u_m(k_m+i),\ i=0,\dots,N_m-1 \\ \overline{x}_f(k_m+i), \overline{u}_f(k_m+i),\ i=0,\dots,N_m-1}} J_m\big(x_m(k_m)\big), \tag{2.5}$$

where

$$\begin{aligned} &J_m\big(x_m(k_m)\big) = \\ &= \sum_{i=0}^{N_m-1} l_m\Big(\big(x_m(k_m+i) - \overline{x}_m(k_s)\big), \big(u_m(k_m+i) - \overline{u}_m(k_s)\big), \\ &\quad \overline{x}_f(k_m+i), \overline{u}_f(k_m+i)\Big) + v_m\big(x_m(k_m+N_m) - \overline{x}_m(k_s)\big) \end{aligned}$$

and $\big(\overline{x}_m(k_s), \overline{u}_m(k_s)\big)$ is the reference provided by the controller at the high level, subject to the following constraints:

- the system dynamics (2.2) with,

$$\forall i = 0, \dots, N_m - 1, \quad \begin{cases} x_f^m(k_m+i) = \overline{x}_f(k_m+i) \\ u_s^m(k_m+i) = u_s(k_s) \\ U_f^m(k_m+i) = \overline{u}_f(k_m+i), \end{cases}$$

  and the initial condition $\big(x_s^m(k_m), x_m(k_m)\big)$ $\big($where $x_s^m(k_m)$ is supposed to be available, see the discussion in Section 3$\big)$;

- the additional feasibility constraints
  - the 6-tuple $\big(x_s^m(k_m+i), u_s^m(k_m+i), x_m(k_m+i),\ u_m(k_m+i), \overline{x}_f(k_m+i),\ \overline{u}_f(k_m+i)\big)$ is m-admissible $\forall i = 0, \dots, N_m - 1$;

– $x_s^m(k_m+i) \in \mathscr{X}_s \quad \forall i = 1,\ldots,N_m$;
– $x_m(k_m+N_m) \in \mathscr{X}_m$.

Then, according to the RH paradigm, only the first computed value $u_m(k_m)$ is applied and the overall procedure is repeated at the new medium sampling period. Also in this case, besides computing the optimal control sequence at the middle level, also the desired values of the state and of input variables $\left(\bar{x}_f(k_m), \bar{u}_f(k_m)\right)$ at the low level are returned.

Remark 3 *Notice that in* (2.5) *the reference value* $\left(\bar{x}_m(k_s), \bar{u}_m(k_s)\right)$, *as well as the control* $u_s(k_s)$, *are considered constant over the prediction horizon even if time instants* $k_m \geq \nu_s'(k_s+1)$ *are included in the horizon.* $\diamond$

Remark 4 *If the current state* $x_f^m(k_m)$ *of the low level is available to the controller at the middle level, in order that a sensible reference be provided to the low level, one can add the constraint* $\|x_f^m(k_m) - \bar{x}_f(k_m)\| \leq \varepsilon_f$ *(for suitably small* $\varepsilon_f \geq 0$*) to the optimization problem* (2.5). $\diamond$

MPC problem at the fast time scale

Finally we consider the fast time scale. Now the goal is to track given reference values $(\bar{u}_f, \bar{x}_f)$ of the state and input variables provided by the controller at the middle level. Then, similarly to the two upper levels, the variable $u_f(k_f)$ is computed by solving an MPC optimization problem and by adopting the Receding Horizon paradigm. Then, letting $l_f(\cdot,\cdot)$ and $v_f(\cdot)$ be positive cost functions, $N_f > 0$ be an integer and

$$k_m = \left\lfloor \frac{k_f}{\nu_m} \right\rfloor,$$

we consider the optimization problem

$$\min_{u_f(k_f+i),\ i=0,\ldots,N_f-1} J_f\big(x_f(k_f)\big), \tag{2.6}$$

where

$$\begin{aligned} & J_f\big(x_f(k_f)\big) = \\ & = \sum_{i=0}^{N_f-1} l_f\Big(\big(x_f(k_f+i) - \bar{x}_f(k_m)\big), \big(u_f(k_f+i) - \bar{u}_f(k_m)\big)\Big) + \\ & \quad + v_f\big(x_f(k_f+N_f) - \bar{x}_f(k_m)\big) \end{aligned}$$

and where $(\bar{x}_f(k_m), \bar{u}_f(k_m))$ is the reference provided by the MPC controller at the middle level, subject to the following constraints:

- the system dynamics (2.1) with[1],

$$\forall i = 0,\ldots,N_f-1, \qquad \begin{cases} u_s^f(k_f+i) = u_s^m(k_m) \\ u_m^f(k_f+i) = u_m(k_m), \end{cases}$$

  and the initial condition $\left(x_s^f(k_f), x_m^f(k_f), x_f(k_f)\right)$ (where $x_s^f(k_f)$ and $x_m^f(k_f)$ are supposed to be available, see the discussion in Section 3);

[1]Notice that, with $k_s = \left\lfloor \frac{k_f}{\nu_s} \right\rfloor$, it holds that $u_s^m(k_m) = u_s(k_s)$.

- the feasibility constraints
  - $x_s^f(k_f+i) \in \mathscr{X}_s \quad \forall i = 1,\ldots,N_f$;
  - $x_m^f(k_f+i) \in \mathscr{X}_m \quad \forall i = 1,\ldots,N_f$;
  - $x_f(k_f+i) \in \mathscr{X}_f \quad \forall i = 1,\ldots,N_f$;
  - $u_f(k_f+i) \in \mathscr{U}_f \quad \forall i = 0,\ldots,N_f-1$.

Then, according to the RH paradigm, only the first computed value $u_f(k_f)$ is applied and the overall procedure is repeated at the new fast sampling period.

Remark 5 *Similarly to problem* (2.5), *in* (2.6) *the reference value* $\left(\bar{x}_f(k_m),\bar{u}_f(k_m)\right)$, *as well as the controls* $u_s^m(k_m)$ *and* $u_m(k_m)$, *are considered constant over the prediction horizon even if time instants* $k_f \geq \nu_f(k_m+1)$ *or* $k_f \geq \nu_s(k_s+1)$ *are included in the horizon.* ◇

Chapter 3

Communication protocols

In order to implement the proposed multilayer MPC algorithm, it is necessary to specify the communication protocol regulating the information exchange among the layers. First, the basic communication rules are established, then two alternative protocols are described. It is agreed that the time instants are shown in the base fast time scale.

Basic rules:

1. At every instant k_f, each level is supposed to know the current value of its state and control;
2. At time $\nu_s k_s$ the high level communicates to the middle level its current control value $u_s(k_s)$ and the references $\left(\bar{x}_m(k_s), \bar{u}_m(k_s)\right)$;
3. At time $\nu_s k_s$ the middle level communicates to the low level the control value $u_s(k_s)$ which has received by the high level;
   At time $\nu_m k_m$ the middle level communicates to the low level its current control value $u_m(k_m)$ and the references $\left(\bar{x}_f(k_m), \bar{u}_f(k_m)\right)$.

Since the availability of $x_s^m(k_m)$ is needed by the MPC controller at the middle level and the availability of both $x_s^f(k_f)$ and $x_m^f(k_f)$ is needed by the MPC controller at the low level, then more information is to be exchanged between the layers. The choice between the following two communication protocols is proposed (see also Figure 3.1).

Protocol 1:

1. At time $\nu_m k_m$ the high level communicates to the middle level its current state $x_s^m(k_m)$;
2. At time $\nu_s k_s$ the middle level communicates to the low level the value $x_s(k_s)$ which has received by the high level and its current state $x_m^s(k_s)$.

Protocol 2:

1. At time $\nu_s k_s$ the high level communicates to the middle level its current state $x_s(k_s)$;
2. At time $\nu_s k_s$ the low level communicates to the middle level its current state $x_f(\nu_s k_s)$;
   At time k_f the low level communicates to the middle level its current control $u_f(k_f)$;
3. At time $\nu_s k_s$ the middle level communicates to the low level the value $x_s(k_s)$ which has received by the high level and its current state $x_m^s(k_s)$.

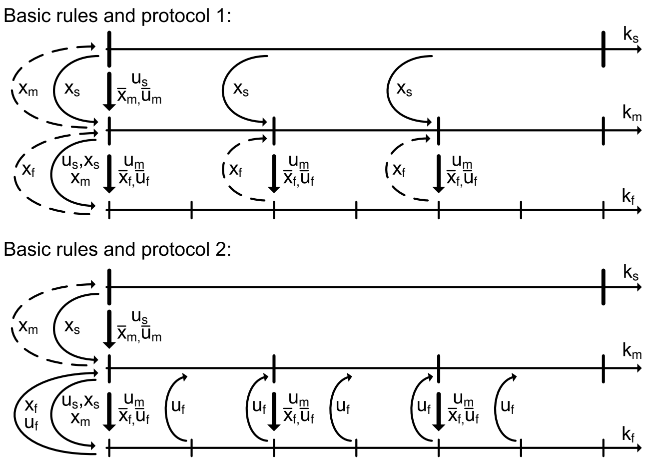

Figure 3.1: Pictorial representation of the proposed communication protocols. *Straight arrows:* communication of the result of the optimization problems; *Curved arrows:* other information to be transmitted; *Broken arrows:* the optional information transmission discussed in Remarks 2 and 4.

Both protocols are able to provide the necessary information to the controllers. In fact, in case of Protocol 1, the availability of $x_s^m(k_m)$ to the MPC controller at the middle level is guaranteed by rule 1. As for the low level, the whole information of the system is available because, at time $\nu_s k_s$, all the components of the state are known as well as all the successive control values applied at any level. In case of Protocol 2, the whole information of the system is available to both the middle and the low level because, at time $\nu_s k_s$, all the components of the state are known as well as all the successive control values applied at any level.

Remark 6 *In Protocol 1 there is an exact pyramidal structure of the information on the system: any level has no information concerned with the lower levels and the lowest level knows everything on the system.*
According to Protocol 2, the amount of information exchanged is much larger than that in Protocol 1. On the other hand, Protocol 2 allows one to reduce the information transmitted by the high level (i.e., the top level of the hierarchy is less pressed with information requests). However, such a reduction is to be counterbalanced by a large amount of information to be transmitted by the low level (see Figure 3.1). As a consequence, much more information becomes available to the middle level than that really needed $\big($*in fact, the whole information on the system is available rather than* $x_s^m(k_m)$ *only*$\big)$.
◇

In both Protocols 1 and 2, tolerance to unmodelled disturbances affecting the system can be gained if the information transmission from the middle level to the low level is more frequent: e.g., if at time $\nu_m k_m$ the middle level communicates to the low level the value $x_s^m(k_m)$ (which has received by the high level, in case of Protocol 1, or which has reconstructed, in case of Protocol 2) and its current state $x_m(k_m)$.
Finally, notice that to implement the MPC controllers at the high and at the middle level according to the modification suggested in Remarks 2 and 4, a further flow of information from lower to higher levels is needed.

Chapter 4

Hierarchical structures

Let us show how the proposed approach fits well with the different cases, mentioned in the Introduction, which can arise when designing a hierarchical control structure.

Case 1. When the system is divided into functional layers, the control structure can be seen as a cascade one, where the states of an inner loop are the control variables for the outer one, as shown in Figure 4.1. This control structure can be described in the layers framework adopted in this paper as follows: the cascade interconnection of three systems

$$x_s^f(k_f+1) \ = \ f_s^f\left(x_s^f(k_f), u_s^f(k_f)\right) \tag{4.1a}$$

$$x_m^f(k_f+1) \ = \ f_m^f\left(x_m^f(k_f), u_m^f(k_f)\right) \tag{4.1b}$$

$$x_f(k_f+1) \ = \ f_f^f\left(x_f(k_f), u_f(k_f)\right), \tag{4.1c}$$

in which $u_s^f = x_m^f$ and $u_m^f = x_f$, is equivalent to the system

$$\left\{ \begin{array}{l} x_s^f(k_f+1) = f_s^f\left(x_s^f(k_f), x_m^f(k_f)\right) \\ x_m^f(k_f+1) = f_m^f\left(x_m^f(k_f), x_s^f(k_f)\right) \\ x_f(k_f+1) = f_f^f\left(x_f(k_f), u_f(k_f)\right). \end{array} \right. \tag{4.2}$$

The latter is a special case of system (2.1) and can be controlled through the proposed algorithm. Specifically, the MPC controller at the high level (working in the long time scale) provides the reference $\bar{x}_m$ to the controller at the medium level; this reference is tracked by the MPC controller at the middle level (working in the medium time scale) which returns the state and control reference $(\bar{x}_f, \bar{u}_f)$ for the low level controller (working in the fast time scale) which in turn computes the real input $u_f(k_f)$ for system (4.2), see Figure 4.1.

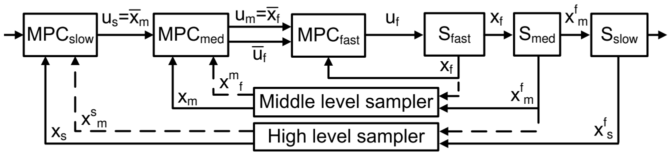

Figure 4.1: Cascade of three interconnected systems: the proposed control structure (broken lines represent the optional connections discussed in Remarks 2 and 4).

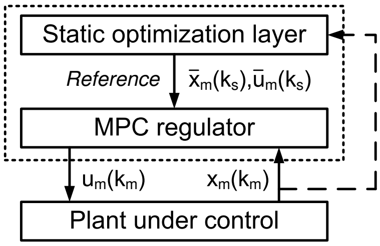

Figure 4.2: The typical structure of a hierarchical controller adapted to the framework considered in this paper.

Thanks to the special form of the system, the "basic rules" of the communication protocols described in Chapter 3 suffices for the implementation of the control algorithm.

Case 2. It is apparent that singularly perturbed systems, i.e., systems with interlaced slow and fast dynamics, perfectly fits with the adopted model description: it is sufficient to consider a two layers version of the proposed algorithm.

Case 3. Hierarchical control structures find another helpful employment to handle optimal control problems over a long time horizon $N_s \gg 1$ for complex systems. A sub-optimal solution to this otherwise untractable problem is provided by a two layers hierarchical controller organized as follows: at the top level the long horizon problem is solved for a simplified model of the system; at the bottom level an optimal control problem, to track the top level solution, is solved for the overall model but with a shorter horizon $N_m \ll N_s$. This approach has been proposed in [4]-Section 1.2. Also the control algorithm presented in this paper is suitable to deal with such a problem, in fact: consider a two layers version of the algorithm for a system of the type

$$\begin{cases} x_s^m(k_m+1) = f_s^m\big(x_s^m(k_m), u_s^m(k_m), u_m(k_m)\big) & (4.3a) \\ x_m(k_m+1) = f_m^m\big(x_m(k_m), u_s^m(k_m), u_m(k_m)\big), & (4.3b) \end{cases}$$

where equation (4.3b) is a detailed model of the plant under control and equation (4.3a) is a reduced order model of the same plant with f_s^m accounting for the slow dynamics of the system. In this case, the optimization problem (2.4) returns the "slow" component $u_s(k_s)$ of the control variable (in the slow time scale) and a reference $\bar{u}_m(k_s)$ for the "medium" component. In correspondence with $\big(u_s(k_s), \bar{u}_m(k_s)\big)$, an equilibrium state $\bar{x}_m(k_s)$ is computed for system (4.3b) $\big($assume either that $\bar{x}_m(k_s)$ is univocally determined by $\big(u_s(k_s), \bar{u}_m(k_s)\big)$ or that some rule to select $\bar{x}_m(k_s)$ is given$\big)$. Hence, the MPC controller at the middle level solves a tracking problem, with reference $\big(\bar{x}_m(k_s), \bar{u}_m(k_s)\big)$, for system (4.3b) and returns the medium component $u_m(k_m)$ of the control signal.
In this formulation, a less detailed (but dynamical) model of the plant is employed to fix the reference (this layer is called *dynamic-RTO* [5]), then the overall model is considered to determine the control action. This is different from case 2, where the control synthesis is carried out by solving various optimal control problems defined on simpler sub-systems.

Case 4. The easiest version of the hierarchical controller described in case 3 has a two layers structure where the top level performs a static optimization (*static-RTO*) and returns the set-point for the bottom level dynamic regulator, see Figure 4.2. Also this control structure results as a special case of the control algorithm described in this paper, in fact: if equation (2.1a) is static, then the high level controller resulting from the proposed algorithm is nothing but the static optimization layer. In a formal way, let $x(k+1) = f\big(x(k), u(k)\big)$ be a dynamical system modelling the plant under control, add a fictitious equation $z(k+1) = z(k)$ and consider a two layers version of the algorithm where $x_m = x$, $x_s = z$ and $k_m = k$ is the base time scale. In this way, the optimization problem (2.4) turns out to be a static optimization over the equilibrium pairs $(\bar{x}_m, \bar{u}_m)$ of the plant providing the reference for the bottom level MPC controller. Notice that, if the current state of the system is available to the static optimizer and a constraint $\|x_m^s(k_s) - \bar{x}_m(k_s)\| \leq \varepsilon_m$ is considered (see Remark 2), then a new set-point is computed at any time instant k_s so that the reference is time-varying.

Chapter 5

An illustrative example

Consider a system composed by three tanks as in Figure 5.1. Each tank has an input flow: tanks 2 and 3 are fed by $\tilde{q}_{i2}$ and $\tilde{q}_{i3}$; tank 1, which is supposed to be significantly larger than the other tanks, is fed by tanks 2 and 3 with flows $\tilde{q}_{o2}$ and $\tilde{q}_{o3}$ through pipes with fixed opening valves. There is also an output flow $\tilde{q}_o$ from tank 1. For $i = 1,2,3$, let the state variable $\tilde{x}_i$ denote the level of the i-th tank, while the control variables are the flows $\tilde{q}_{i2}$, $\tilde{q}_{i3}$ and $\tilde{q}_o$. A discrete time and linearized model for this system is the following:

$$\begin{cases} x_1(k+1) = x_1(k) + \beta_3 x_3(k) + \beta_2 x_2(k) - \beta_1 q_o(k) & (5.1a) \\ x_2(k+1) = (1-\beta_2)x_2(k) + \alpha_2 q_{i2}(k) & (5.1b) \\ x_3(k+1) = (1-\beta_3)x_3(k) + \alpha_3 q_{i3}(k), & (5.1c) \end{cases}$$

where $x_i = \tilde{x}_i - x_i^*$ $(i = 1,2,3)$, $q_{ij} = \tilde{q}_{ij} - q_{ij}^*$ $(j = 2,3)$ and $q_o = \tilde{q}_o - q_o^*$, with $(x_1^*, x_2^*, x_3^*, q_{i2}^*, q_{i3}^*, q_o^*)$ being a given equilibrium for the nonlinear model of the system. Moreover, β_1, β_2, β_3, α_2 and α_3 are positive constants (which depend on the areas of the tanks and on the characteristics of the valves) and β_2, β_3 are smaller than 1.
For this system, four different hierarchical controllers can be constructed according to the four applications of the proposed algorithm discussed above:
Case 1. System (5.1) can be viewed as the cascade of two subsystems characterized by two different dynamics: one for tanks 2 and 3 and a slower one for tank 1 $\big($thus, $x_m = (x_2, x_3)$, $u_m = (q_{i2}, q_{i3})$, $x_s = x_1$ and $u_s = q_o\big)$. Indeed, system (5.1) is in the form of system (4.2) except for the presence of a slow control variable q_o directly acting on the slow part of the system. According to the proposed algorithm, the optimization problem for (5.1a) returns the input q_o and the reference $(\overline{x}_2, \overline{x}_3, \overline{q}_{i2}, \overline{q}_{i3})$ for (5.1b) and (5.1c). Then the low level controller computes q_{i2} and q_{i3}.

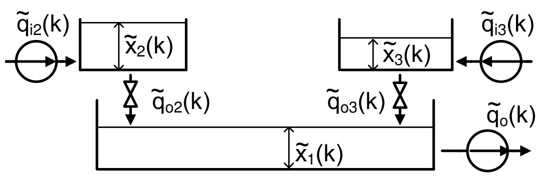

Figure 5.1: System composed by three tanks.

Case 2. Letting $x_{\rm t} = x_1 + x_2 + x_3$, system (5.1) becomes

$$\begin{cases} x_{\rm t}(k+1) = x_{\rm t}(k) + \alpha_3 q_{\rm i3}(k) + \alpha_2 q_{\rm i2}(k) - \beta_1 q_{\rm o} & (5.2a) \\ x_2(k+1) = (1-\beta_2)x_2(k) + \alpha_2 q_{\rm i2}(k) & (5.2b) \\ x_3(k+1) = (1-\beta_3)x_3(k) + \alpha_3 q_{\rm i3}(k) & (5.2c) \end{cases}$$

where the slow dynamics $x_s = x_{\rm t}$ is decoupled by the faster one $x_m = (x_2, x_3)$. The control algorithm for this case works in the same way as in case 1 $\big($but the model (5.2a) used to produce $q_{\rm o}$ and the reference for the low level is different$\big)$.

Case 3. Consider system (5.2) and let $x_s = x_{\rm t}$, $x_m = (x_{\rm t}, x_2, x_3)$, $u_s = q_{\rm o}$ and $u_m = (q_{\rm i2}, q_{\rm i3})$. In this way, equation (5.2a) is a reduced order model of the system and plays the role of (4.3a), while in place of (4.3b) we take the overall system (5.2). According to the algorithm, the optimization problem for (5.2a) computes the input $q_{\rm o}$ and the reference $(\overline{q}_{\rm i2}, \overline{q}_{\rm i3})$ for the "medium" component. The MPC controller at the middle level solves a tracking problem for the overall system (5.2), with suitable reference $\big(\overline{x}_{\rm t}, \overline{x}_2, \overline{x}_3, \overline{q}_{\rm i2}, \overline{q}_{\rm i3}\big)$, and returns the control values $q_{\rm i2}$ and $q_{\rm i3}$. Notice that, since equation (5.2a) is an integrator, the relation $\alpha_3 \overline{q}_{\rm i3} + \alpha_2 \overline{q}_{\rm i2} - \beta_1 q_{\rm o} = 0$ must be satisfied and, in correspondence with such a $(q_{\rm o}, \overline{q}_{\rm i2}, \overline{q}_{\rm i3})$, there is not an unique equilibrium state $(\overline{x}_{\rm t}, \overline{x}_2, \overline{x}_3)$ (specifically, $\overline{x}_{\rm t}$ is not univocally determined).

Case 4. The reference $\big(\overline{x}_{\rm t}, \overline{x}_2, \overline{x}_3, \overline{q}_{\rm i2}, \overline{q}_{\rm i3}, \overline{q}_{\rm o}\big)$ for the overall system (5.2) is provided by the static optimization layer. $\diamond$

Chapter 6

Conclusions

A general formulation to the problem of designing hierarchical control structures with MPC has been described in this report. It has also been shown how the adopted approach and the corresponding sequence of MPC synthesis methods can deal with a number of significant industrial control problems.
The proposed approach is coherent with the goals of Task 2.2 of Work Package 2 (WP2), since it defines a hierarchical control architecture that integrates sequential decisions in the global MPC scheme, and allows for the use at each level of various optimisation criteria (quadratic, linear, etc.) and control schemes (MPC, classical PID, etc.). Moreover, it fits with the activity planned in the framework of Task 2.3 (WP2), as different communication protocols and constraints have been considered between adjacent layers of the hierarchical control structures. Finally, it allows for the use of multi-level, multi-resolution models, i.e., models with various levels of spatial and temporal aggregation, as required in Task 2.4 (WP2). Concerning this last point, it is worth recalling that existing reduction and aggregation methods to obtain such models have been extensively reviewed in Deliverable D2.1.
The present work can be seen as partially preliminary to the activities planned in Work Package 3, which will concern the modification of the basic MPC algorithms here proposed to achieve guaranteed stability and robustness properties, see the papers [7] and [6] on these aspects.
Secondly, while only the regulation problem has been considered here, the output feedback, tracking and disturbance rejection problems are of paramount importance and require further developments.

Bibliography

[1] M.A. Brdys, M. Grochowski, T. Gminski, K. Konarczk, M. Drewa, Hierarchical predictive control of integrated wastewater treatment systems, *Control Engineering Practice*, vol. 16, 2008, pp 751-767.

[2] J.H. Chow, P.V. Kokotovic, A decomposition of near-optimum regulators for systems with slow and fast modes, *IEEE Transactions on Automatic Control*, vol. 21, 1976, pp 701-705.

[3] S. Engell, Feedback control for optimal process operation, *Journal of Process Control*, vol. 17, 2007, pp 203-219.

[4] W. Findeisen, F.N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, A. Wozniak, *Control and Coordination in Hierarchical Systems*, Wiley, 1980.

[5] J.V. Kadam, V. Marquardt, M. Schlegel, T. Backx, O.H. Bosgra, P.-J. Brouwer, G. Dünnebier, D. van Hessem, A. Tiagounov, S. de Wolf, Towards integrated dynamic real-time optimization and control of industrial processes, *FOCAPO 2003*, pp 593-596.

[6] L. Magni, D.M. Raimondo, R. Scattolini, Regional Input-to-State Stability for Nonlinear Model Predictive Control, *IEEE Transactions on Automatic Control*, vol. 51, 2006, pp 1548-1553.

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

[8] M.D. Mesarovic, D. Macko, Y. Takhare, *Theory of Hierarchical, Multilevel* , Systems, Mathematics in Science and Engineering, vol. 68, 1970.

[9] B. Picasso, C. Romani and R. Scattolini, Hierarchical model predictive control of Wiener models, *Internation Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Pavia*, 2008.

[10] B. Picasso, C. Romani and R. Scattolini, On the design of hierarchical control systems with MPC, *Proc. of the European Control Conference, Budapest*, 2009.

[11] R. Scattolini, P. Colaneri, Hierarchical model predictive control, *46th IEEE Conference on Decision and Control*, 2007, pp 4803-4808.