UNIVERSIDAD POLITÉCNICA DE VALENCIA

Departamento de Ingeniería de Sistemas y Automática.

Doctoral Thesis

MPC: Relevant Identification, and Control in the Latent Variable Space.

Author: David Laurí Pla

Director: Miguel A. Martínez Iranzo
Javier Sanchís Saez

Valencia, January 2012

ABSTRACT

Model-based Predictive Control (MPC) is widely used in Industry due to its ability to handle multivariate systems subject to input and output constraints. Two phases can be distinguished in an MPC implementation: identification and control. The purpose of this thesis is twofold: make contributions in identification for MPC, and propose a new MPC control methodology.

Closed-loop performance in an MPC implementation relies heavily on predictive performance of the model, then model identification is a crucial point in MPC and the part that often demands most of the time of the project. This thesis deals first with identification for MPC. Model identification aims at approximating a process, and models are often fit for purpose. Provided the purpose of the model in MPC is to perform multi-step ahead predictions, the identification strategy needs take multi-step ahead prediction errors into account. This identification strategy is often denoted MRI (Model Predictive Control Relevant Identification). In this thesis, identification for MPC covers three main topics. First, MRI and different approaches to it are defined. Second, fitting a multiple input multiple output model is compared to fitting

several multiple input single output models in terms of MRI concluding the approach to obtain a single multiple input multiple output model is preferable in MRI for a sufficiently large prediction horizon. Finally a PLS-based (Partial Least Squares) line search numerical optimization approach denoted PLS-PH that deals with parametric MRI in the case of collinearity in the identification data set is proposed. An example shows PLS-PH can outperform conventional MRI parametric approaches if there is collinearity in the identification data set.

Once the model to perform multi-step ahead predictions is ready, the controller can be formulated. A model-based predictive control methodology in the space of the latent variables for continuous processes is proposed in this thesis and denoted LV-MPC. LV-MPC takes the dynamic matrix approach for MRI and uses PLS to obtain the latent variable space in which the decisions of the controller are made. Implementing identification and control in the latent variable space: eases identification in the case of correlation in the identification data set, acts as a prefilter reducing the effect of noisy data, reduces computational complexity, and provides tools that ensure the predictor is used in the region in which it has been identified, hence improving closed-loop performance. Several examples show how LV-MPC can outperform traditional MPC in terms of computational complexity and closed-loop performance whilst it is easy to tune. LV-MPC for continuous processes is a novel approach that combines the best of two methodologies widely used in Industry: MPC and latent variable methods. This thesis describes LV-MPC and tackles with many of the challenges in MPC, however, there is room to add more functionalities to the methodology, then LV-MPC starts what can become a new trend in MPC.

RESUMEN

Control predictivo basado en modelos (MPC) es una metodología de control ampliamente utilizada en la industria por su habilidad para controlar procesos multivariable con restricciones en sus entradas y sus salidas. Se distinguen dos fases en la implementación de MPC: identificación y control. El propósito de esta tesis es doble: realizar contribuciones en la identificación para MPC y proponer una nueva metodología de control MPC.

La respuesta en bucle cerrado de una implementación de MPC depende, en gran medida, de la capacidad de predicción del modelo; luego la identificación del modelo es un punto crucial en MPC y la parte que a menudo exige la mayor parte del tiempo del proyecto. El primer objetivo que cubre la tesis es la identificación para MPC. Puesto que un modelo es una aproximación del comportamiento de un proceso, dicha aproximación se puede hacer teniendo en cuenta el fin que se le va a dar al modelo. En MPC, el modelo se utiliza para realizar predicciones dentro de una ventana futura, luego la identificación para MPC (MRI) tiene en cuenta dicho uso del modelo y considera los errores de predicción dentro de dicha ventana para el ajuste de los parámetros del

modelo. En esta tesis, se cubren tres temas dentro de MRI. Primero se define MRI y las distintas formas de abordarlo. Luego se compara en términos de MRI el ajuste de un modelo con múltiples entradas y múltiples salidas con el ajuste de varios modelos con múltiples entradas y una salida concluyendo que el ajuste de un único modelo con múltiples entradas y múltiples salidas proporciona mejores resultados en términos de MRI para horizontes de predicción lo suficientemente grandes. Por último, se propone el algoritmo PLS-PH para implementar MRI con modelos paramétricos en el caso de correlación en los datos de identificación. PLS-PH es un método de optimización numérica por búsqueda lineal basado en PLS (mínimos cuadrados parciales). Se muestra en un ejemplo como PLS-PH es capaz de proporcionar mejores modelos que las técnicas convencionales de MRI en modelos paramétricos en el caso de correlación en los datos de identificación.

Una vez obtenido el modelo se puede formular el controlador predictivo. En esta tesis se propone LV-MPC, un controlador predictivo para procesos continuos que implementa la optimización en el espacio de las componentes principales. En LV-MPC se obtienen las matrices dinámicas del modelo directamente desde los datos de identificación mediante PLS. Las matrices de transformación del espacio obtenidas en PLS se utilizan posteriormente en el controlador para implementar la optimización en dicho espacio reducido. Implementar identificación y control en el espacio de las componentes principales: facilita la identificación en el caso de existencia de correlación en el conjunto de datos de identificación, actúa como filtro reduciendo los efectos de datos con ruido, reduce la carga computacional del algoritmo de control y asegura la utilización del modelo en la región en la que ha sido identificado mejorando la respuesta en bucle cerrado. Varios ejemplos muestran como LV-MPC puede mejorar la implementación tradicional de MPC en términos de carga computacional, respuesta en bucle cerrado y siendo una metodología fácil de ajustar. LV-MPC es una nueva metodología que combina lo mejor de dos herramientas ampliamente utilizadas en la industria: control predictivo y técnicas de reducción de variables. En esta tesis se describe LV-MPC y se abordan varios de los retos en la implementación de control predictivo, no obstante quedan líneas abiertas para futuros trabajos.

RESUM

Control predictiu basat en models (MPC) és una metodologia de control àmpliament utilitzada en la indústria per la seva habilitat per controlar processos multivariable amb restriccions en les seves entrades i les seves sortides. Es distingeixen dues fases en la implementació de MPC: identificació i control. El propòsit d'aquesta tesi és doble: realitzar contribucions en la identificació per a MPC i proposar una nova metodologia de control MPC.

La resposta en bucle tancat d'una implementació de MPC depèn en gran mesura de la capacitat de predicció del model, doncs la identificació del model és un punt crucial en MPC i la part que sovint exigeix la major part del temps del projecte. El primer objectiu que cobreix la tesi és la identificació per a MPC. Com que un model és una aproximació del comportament d'un procés, aquesta aproximació es pot fer tenint en compte la finalitat que se li donarà al model. En MPC, el model s'utilitza per a realitzar prediccions dins d'una finestra, doncs la identificació per a MPC (MRI) té en compte aquest ús del model i considera els errors de predicció dins d'aquesta finestra per a l'ajust dels paràmetres del model. En aquesta tesi, es cobreixen tres temes dins de

MRI. Primer es defineix MRI i les diferents formes d'abordar-lo. Després es compara en termes de MRI l'ajust d'un model amb múltiples entrades i múltiples sortides amb l'ajust de diversos models amb múltiples entrades i una sortida concloent que l'ajust d'un únic model amb múltiples entrades i múltiples sortides proporciona millors resultats en termes de MRI per a horitzons de predicció suficientment grans. Finalment, es proposa l'algorisme PLS-PH per implementar MRI amb models paramètrics en el cas de correlació en les dades d'identificació. PLS-PH és un mètode d'optimització numèrica per recerca lineal basat en PLS (mínims quadrats parcials). Es mostra en un exemple com PLS-PH és capaç de proporcionar millors models que les tècniques convencionals de MRI en models paramètrics en el cas de correlació en les dades d'identificació.

Una vegada obtingut el model es pot formular el controlador predictiu. En aquesta tesi es proposa LV-MPC, un controlador predictiu per a processos continus que implementa l'optimització en l'espai de les components principals. En LV-MPC s'obtenen les matrius dinàmiques del model directament des de les dades d'identificació mitjançant PLS. Les matrius de transformació de l'espai obtingudes en PLS s'utilitzen posteriorment al controlador per implementar l'optimització en aquest espai reduït. Implementar identificació i control en l'espai de les components principals: facilita la identificació en el cas d'existència de correlació en el conjunt de dades d'identificació, actua com a filtre reduint els efectes de dades amb soroll, redueix la càrrega computacional de l'algorisme de control i assegura la utilització del model en la regió en què ha estat identificat millorant la resposta en bucle tancat. Diversos exemples mostren com LV-MPC pot millorar la implementació tradicional de MPC en termes de càrrega computacional, resposta en bucle tancat i sent una metodologia fàcil d'ajustar. LV-MPC és una nova metodologia que combina el millor de dues eines àmpliament utilitzades en la indústria: control predictiu i tècniques de reducció de variables. En aquesta tesi es descriu LV-MPC i s'aborden alguns dels reptes en la implementació de control predictiu, però queden línies obertes per a futurs treballs.

ACRONYMS

ARX	*Autoregressive Exogenous*
ARMAX	*Autoregressive Moving Average Exogenous*
BJ	*Box-Jenkins*
CRI	*Control Relevant Identification*
CVs	*Controlled Variables*
DM	*Dynamic Matrix*
LMOCV	*Leave-Many-Out Cross Validation*
LTI	*Linear Time Invariant*
LS	*Least Squares*
LS-PH	*Least Squares Prediction Horizon*
LTI	*Linear Time Invariant*
LVMs	*Latent Variable Methods*
LV-MPC	*Latent Variable Model Predictive Control*
MPC	*Model-based Predictive Control*
MIMO	*Multiple Input Multiple Output*
MISO	*Multiple Input Single Output*
MVs	*Manipulated variables*
MSEP	*Mean Square Error of Prediction*

NIPALS	*Nonlinear Iterative Partial Least Squares*
PEMFC	*Proton Exchange Membrane Fuel Cell*
PEMs	*Prediction Error Methods*
PLS	*Partial Least Squares*
PLS-PH	*Partial Least Squares Prediction Horizon*
QP	*Quadratic programming*
SEP	*Squared Error of Prediction*
SIMs	*Subspace Identification Methods*
SISO	*Single Input Single Output*
SNR	*Signal to noise ratio*
SSEP	*Sum of Squared Error of Prediction*

NOMENCLATURE

$\mathbf{E}$	input residuals
$\mathbf{F}$	Output residuals
N	Number of rows in the regression matrix
n_a	Number of lagged outputs in the model
n_b	Number of lagged inputs in the model
n_e	Number of error terms used to fit the parameters of the model
n_i	Number of inputs in the process
n_f	Prediction horizon
n_{lv}	Number of latent variables
n_o	Number of outputs in the process
n_p	Number of parameters in the model
n_u	Cotrol horizon
n_x	Number of columns in the regression matrix
n_y	Number of columns in the matrix of outputs for linear regression
$\mathbf{P}$	Input loadings
$\mathbf{Q}$	Output loadings
$\mathbf{T}$	Input scores
T_s	Sampling period

$\mathbf{U}$	Output scores
$\mathbf{u}_{\mathrm{k}}$	Row vector of inputs at instant k
$u_{\mathrm{i}}(k)$	Input i at instant k
$\mathbf{W}_{\mathrm{u}}$	Relative Weight for control moves in MPC
$\mathbf{W}_{\mathrm{y}}$	Relative Weight for predicted deviations in MPC
$\mathbf{X}$	Regression matrix
$\mathbf{Y}$	Output space
$\mathbf{y}_{\mathrm{k}}$	Row vector of outputs at instant k
$y_{\mathrm{i}}(k)$	Output i at instant k
λ_u	Absolute weight for control moves in MPC

CONTENTS

Part I

Introduction

CHAPTER 1

MOTIVATION AND AIM

Model-based Predictive Control (MPC) is widely used in Industry due to its ability to handle multivariate systems subject to input and output constraints [Camacho 04, Jämsä-Jounela 07, Thwaites 07]. The most common challenges that appear when implementing MPC are

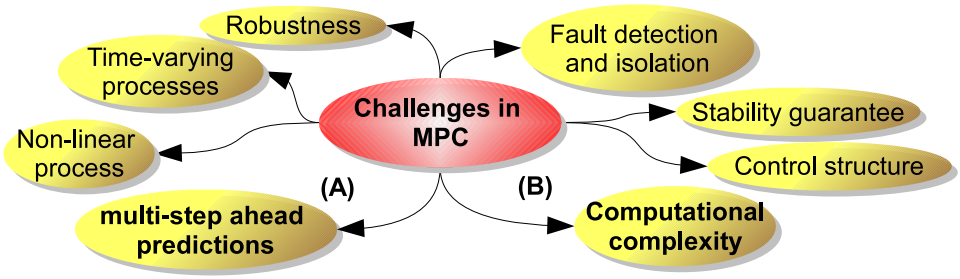


Figure 1.1: Challenges in MPC

Among the challenges of the implementation of MPC in Industry, two of them form the basis of the present thesis: **(A)** Obtain reliable multi-step ahead predictions, and **(B)** Reduce computational complexity. Challenge **(A)** can be split into two parts:

(A) Obtain reliable multi-step ahead predictions

- **(A.1)** Obtain a reliable multi-step ahead predictor
- **(A.2)** Use the predictor in a region in which it is valid

(B) Reduce computational complexity.

Model development is by far the most critical and time-consuming step in implementing a model predictive controller [Morari 99, Zhu 02, Bars 06]. Control Relevant Identification (CRI) is intended to create synergy between the identification and control algorithms; thus providing a model that is commensurate with the control cost function. The acronym MRI (MPC Relevant Identification) refers to CRI in the case of an MPC controller, and tackles challenge **(A.1)**.

As depicted in Figure 1.2, MRI can be attained in two different ways: the parametric model approach, and the dynamic matrix approach. The parametric model approach takes into account the further use of the model to perform predictions in a given prediction window, i.e., it is MRI. The parametric model is then used in MPC to obtain the dynamic matrices used to perform multi-step ahead predictions. There are three ways to obtain MRI parametric models: pre-filter the identification data set [Shook 92], minimize a multi-step ahead prediction error cost function [Laurí 10a], and use subspace identification methods (SIMs) [Huang 08]. In the dynamic matrix approach, the dynamic matrices used in MPC are obtained directly from process data [Rossiter 01, Kadali 03], and either a singlemodel to perform predictions in the prediction horizon can be obtained, or as many models as the length of the prediction horizon in the multimodel approach. The advantage of the multi-

model approach is it ensures causality of the overall model in that predictions in the near horizon cannot depend on inputs in the far horizon.

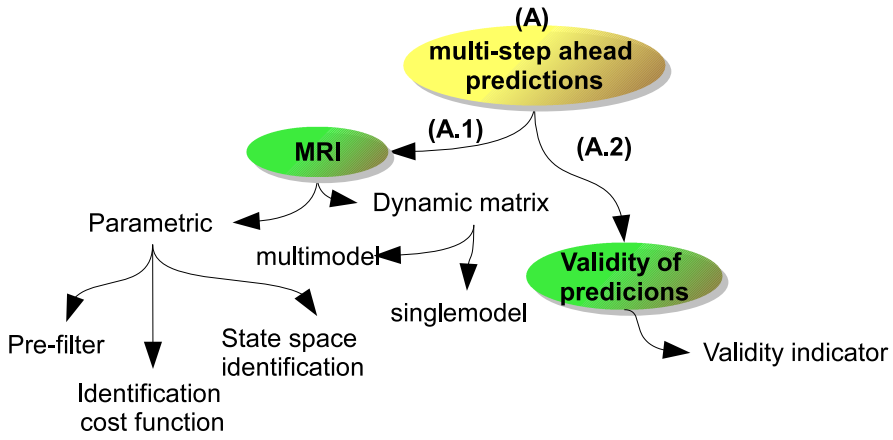


Figure 1.2: Challenge **(A)**: Obtain reliable multi-step ahead predictions.

Lets assume the model has been identified using MRI and multi-step ahead predictions are close enough to the real outputs of the process. Nevertheless, the MPC controller under tight control specifications may use the predictor in a region far from the identification region in which it is not valid[1], then poor multi-step predictions are obtained and consequently poor closed-loop performance. This forms the second branch **(A.2)** under the challenge to obtain reliable multi-step ahead predictions. The approach to tackle this challenge is to consider an indicator of validity of predictions in the on-line optimization of the controller.

In large processes, processes with fast dynamics, or in case a large control window is desired, the MPC problem has many degrees of freedom (d.o.f.). The d.o.f. are the dominating factor for computational complexity, challenge **(B)**. It is common practice to reduce the d.o.f. to deal with computational complexity. One approach to reduce computational complexity is PFC (Pre-

[1]Note MPC based on linear models is often used, and all processes are non-linear, then there is model-process mismatch.

dictive Functional Control) [Camacho 04]. In PFC a subset of points in the prediction horizon is computed to simplify calculation, and the control signal is parametrized using a subset of polynomial basis functions. Another approach is to parametrize the control signal using Laguerre functions [Wang 09]. Move blocking strategies reduce the d.o.f. by fixing the input or its derivatives to be constant over several time-steps, a survey of various move blocking strategies is presented in [Cagienard 07]. Another solution is to implement explicit MPC by means of multi-parametric programming [Kvasnica 04]. In multi-parametric programming, the MPC quadratic optimization problem is solved off-line. The associated solution takes the form of a PWA state feedback law. In particular, the state-space is partitioned into polyhedral sets and for each of those sets the optimal control law is given as one affine function of the state. In the on-line implementation of the explicit MPC controller, computation of the controller action reduces to a set-membership test. Another solution to reduce computational complexity is SVD-GPC [Sanchis 02]. In SVD-GPC, singular value decomposition is used to reduce the dimensional space of the matrices of the controller, which reduces computational burden.

Another solution to reduce complexity is to use LVMs in the identification stage and perform the minimization in the space of the latent variables: in [Flores-Cerrillo 05, MacGregor 09] LV-MPC for batch processes is presented; and in [Song 02] a neural network PLS model is obtained and control is implemented in the space of the latent variables. The MPC based control methodology proposed in Part **III** of this thesis deals with challenge **(B)** by implementing the on-line optimization in the reduced latent variable space.

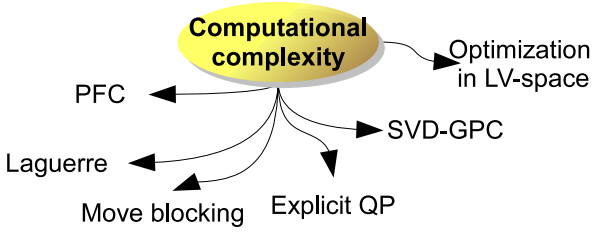


Figure 1.3: Challenge **(B)**: Reduce computational complexity.

Figure 1.4 contains the mindmap of the challenges and possible solutions in MPC. This thesis focuses on challenges (A) and (B), but some solutions to the other challenges are depicted in the figure and commented below, note this list of solutions is not intended to be complete nor exhaustive. As shown in the figure, the control solution proposed in **Part III** of this thesis, and denoted LV-MPC, accounts for challenge (A) in that it uses the single model under the dynamic matrix approach and ensures its validity; and challenge (B) in that the optimization is implemented in the reduced space of the latent variables.

- **Non-linear process**: how to cope with a non-linearity depends on the nature of the non-linearity. If the non-linearity is caused by saturation of the actuators, one solution is to add constraints to the optimization problem in MPC. If there are static non-linearities one can consider adding Hammerstein/Wiener static compensators. For dynamic non-linearities one can use gain-scheduling which interpolates linear controllers [Albertos 04], or use the non-linear model and implement non-linear MPC [Blet 02]. Note using a non-linear model transforms the convex optimization problem to be solved on-line (normally a QP) into a non-convex optimization problem, and this considerably increases computational burden. Hence, although most Industrial processes are non-linear, most implemented control solutions are based on linear models [Qin 03], either one model if the process operates in a region in which it is almost linear, or a set of models for the different operating points in gain-scheduling. LV-MPC accounts for constraints then deals with saturation of the actuators, can be combined with the use of static compensators for static non-linearities, and can be extended to switch among different models if gain-scheduling is to be implemented. Thus, although LV-MPC is presented as a linear control methodology, some adjusts can be made to tackle processes with a strong non-linear behaviour.
- **Time-varying process**: Processes may change their dynamic response over time due to changes in the process, wear of some parts, or persistent unmeasured disturbances that force the plant to a different operating point. Such changes lead to performance degradation provided

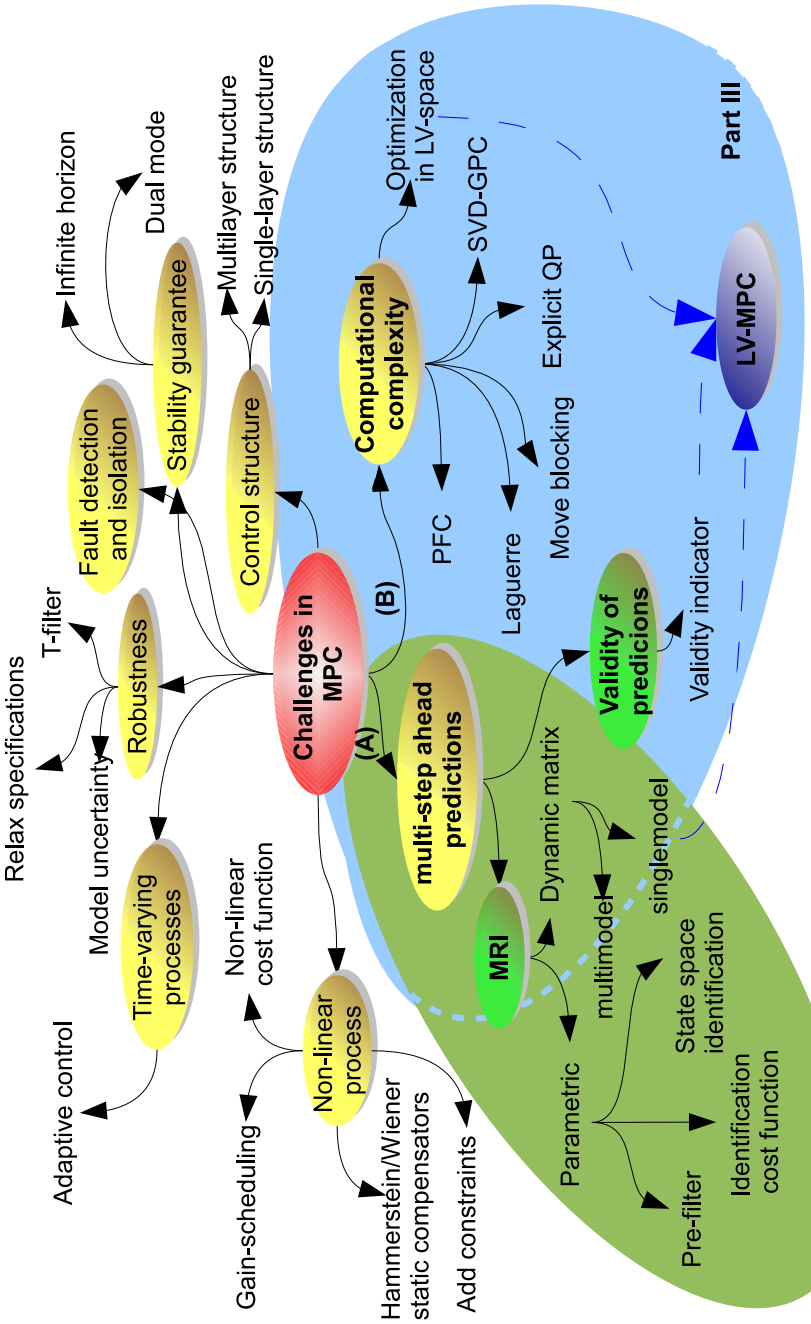


Figure 1.4: Mindmap of challenges and solutions in MPC

multi-step ahead predictions differ from real plant behaviour. The most common approach to cope with time-varying processes is model re-identification [Sotomayor 09], which can be applied to LV-MPC if needed.

- **Robustness**: A control system is robust if stability is maintained and performance specifications are met for a specified range of model variations and a class of noise signals [Bemporad 99]. One way to increase robustness is to include uncertainties to the model and consider them in the design of the controller. Two typical descriptions of uncertainty extensively considered in the field of robust model predictive control are: state space polytope, and bounded unstructured uncertainty [Veselý 10]. Two drawbacks derive from considering uncertainties: solving the problem is computationally much more demanding, and the control action may be excessively conservative. Robustness can also be increased without modelling uncertainties, for instance using large values for the weighting of the control actions in MPC [Veselý 10], i.e., relax control specifications; or using a T-filter [Rossiter 03]. LV-MPC presented in this thesis does not model uncertainties, but implements the optimization in the latent variable space which can reduce the effect of uncertainties (e.g. noise in the controlled variables) in the optimization, and forces the model to be used in the region in which it has been identified ensuring model validity.

- **Fault detection and isolation**: A failure in one actuator or one sensor can lead a process to instability specially in multivariable processes controlled in a multivariable control structure. A condition monitoring tool can be used by control room operators to detect abnormalities and deal with them as soon as possible. Latent variable techniques and the statistic indicators associated, Hotelling's T^2 and squared error of projection, are often used in process monitoring [AlGhazzawi 09]. LV-MPC uses these two indicators to ensure model validity, then it is straight forward to implement fault detection and isolation in LV-MPC.

- **Stability guarantee**: A priori stability guarantee of an MPC control law is established if the MPC cost function is a Lyapunov cost function which implies: it is positive-definite, and its time-derivative is negative semidefinite, which is attained if a infinite horizon is used. However, the

use of infinite horizons make MPC intractable in the case of constraints. One solution is to implement dual mode MPC. In the first mode of dual mode MPC, there is a finite control horizon and constrained on-line optimization is implemented. In the second mode, feedback is given by an explicit control law. At the end of the first mode the process must lay in a terminal set to ensure constraints hold in the second mode. Stability proof of LV-MPC under some assumptions is based on dual model MPC as explained in Part **III**.

- **Control structure**: When controlling multivariate plants, the control structure is an important decision to make. One can use a decentralised control structure with several layers, in the lower layers there are controllers which control parts of the plant, and multivariate controllers are placed in the upper layers. Alternatively, one can use one single layer and implement multivariate control. Whereas decentralised control is easer to tune and commission, centralised control can achieve better results. LV-MPC applies in any of the two structures.

CHAPTER 2

STRUCTURE OF THE THESIS

The thesis consists of four parts and each of them has been divided into chapters to ease reading:

Part I : Introduction

Part II : MRI: Model predictive control Relevant Identification

Part III : LV-MPC: Latent Variable Model Predictive Control

Part IV : Concluding remarks

Part I is the introduction which is divided into four chapters. In the first chapter the motivation of the thesis is provided along with the aim of the thesis. This second chapter outlines the structure of the thesis. The third and fourth chapters briefly describe two tools widely used in industry which form

the basis for the present work: Model Predictive Control (MPC), and Latent Variable Methods (LVMs).

Part II deals with model predictive control relevant identification (MRI) and is enclosed in green in Figure 1.4. This part is also divided into four chapters. In Chapter 5 the concept MRI and the different ways to implement it are introduced along with the model and predictor used in this thesis. Once defined MRI, the MIMO and the multiple MISO identification approaches are compared in Chapter 6; this chapter concludes that MIMO identification is preferable in MRI for a sufficiently large prediction horizon. An identification algorithm to perform parametric MRI in the event of correlated data is presented in Chapter 7. A wrap up for this part is presented in Chapter 8.

Part III introduces the proposed model predictive control methodology implemented in the space of the latent variables (LV-MPC) and is enclosed in blue in Figure 1.4. This part is also divided into three chapters. In Chapter 9 first the basic LV-MPC methodology is defined and then LV-MPC is enhanced with some additional features. For ease of reading despite all the mathematical formulation needed in this part, results are provided at the beginning of each section and the formal proof is given in propositions at the end of each section. Chapter 10 introduces some examples that compare LV-MPC to existing control methodologies proving LV-MPC can reduce computational complexity and provide better closed-loop time response, specially in the event of measurement error, or in the presence of uncertainties such as additive perturbations. Chapter 11 provides the conclusions of this third part of the thesis.

Finally, **Part IV** wraps up the thesis outlining the main conclusions, highlighting the contributions of this thesis and defining further works that either are being or can be done.

CHAPTER 3

MODEL PREDICTIVE CONTROL

3.1. MPC philosophy

MPC is a control philosophy rather than a specific controller. MPC is widely used in Industry because it is based in very intuitive ideas [Camacho 04], and it can handle multivariate systems subject to input and output constraints [Jämsä-Jounela 07, Thwaites 07]. The three basic premises in an MPC controller are:

- A model of the process is used to perform predictions on the behaviour of the process within a prediction window:[1]

$$\hat{\mathbf{y}}_{\mathrm{k+i}} \quad \forall i \in [1, 2, \ldots, n_{\mathrm{f}}]$$

[1] Syntax $\hat{\mathbf{y}}(k+i|k)$ is often used in MPC and reads predicted outputs at instant $k+i$ with information available up to instant k. In this work however, the more compact notation $\hat{\mathbf{y}}_{\mathrm{k+i}}$ is used.

where: $\hat{\mathbf{y}}_{\mathrm{k+i}} \in \mathbb{R}^{1\times n_{\mathrm{o}}}$, predictions at $k+i$ with information available up to instant k; n_{o}, the number of outputs; n_{f}, the prediction horizon. Predictions are expressed in terms of known input and output past data (there is information available up to instant k), and a future control sequence, which is actually the degree of freedom in the MPC problem.

- The control sequence comes from the minimization of a cost function. The basic cost function weights: output deviations from a desired trajectory and the control effort. Should the cost function be quadratic, the model linear, and in the absence of inequality constraints, there is an analytic expression for the MPC controller. Otherwise, numerical optimization can be used to tackle the minimization of the cost function.
- A control sequence is obtained at each instant, but only the first control action is eventually applied to the process. This feature of MPC is defined as receding horizon. The reasoning for the receding horizon policy is at instant $k+1$ there will be more information of the outputs available which was not available at instant k, thus, error in predictions can be accounted for.

The three basic components in an MPC are:

- The **predictor** which is used to estimate the dynamic behaviour of the process within a prediction window.
- The **cost function** from which the control action is obtained.
- The **optimization method** which is used to minimize the cost function.

3.2. Predictor

The predictor is the basic component in an MPC project provided the decision of the control action is based on predictions for the CVs. The better the

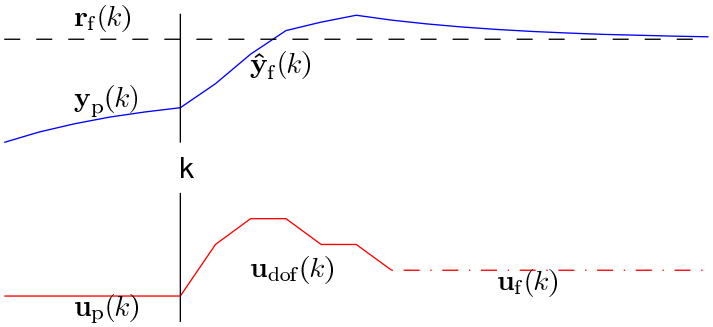


Figure 3.1: Model predictive control

predictions are, the better control performance can be achieved; thus, obtaining a reliable predictor is of importance in MPC. A general form of future predictions can be expressed as a function of input and output past data, and a future control sequence, which is actually the degree of freedom in the MPC problem (see Fig. 3.1).

$$\hat{\mathbf{y}}_\mathrm{f}(k) = f(\mathbf{u}_\mathrm{p}(k), \mathbf{y}_\mathrm{p}(k), \mathbf{u}_\mathrm{f}(k), \mathbf{u}_\mathrm{dof}(k)) \tag{3.1}$$

where all the row vectors are defined

$\hat{\mathbf{y}}_\mathrm{f}(k)$	$=$	$[\hat{\mathbf{y}}_{\mathrm{k}+1} \ \dots \ \hat{\mathbf{y}}_{\mathrm{k}+\mathrm{n_f}}]$	$\rightarrow$	Predicted outputs
$\mathbf{u}_\mathrm{p}(k)$	$=$	$[\mathbf{u}_{\mathrm{k}-1} \ \dots \ \mathbf{u}_{\mathrm{k}-\mathrm{n_b}+1}]$	$\}\rightarrow$	Past known data
$\mathbf{y}_\mathrm{p}(k)$	$=$	$[\mathbf{y}_{\mathrm{k}-1} \ \dots \ \mathbf{y}_{\mathrm{k}-\mathrm{n_a}}]$		
$\mathbf{u}_\mathrm{f}(k)$	$=$	$[\mathbf{u}_{\mathrm{k}+\mathrm{n_f}-1} \ \dots \ \mathbf{u}_{\mathrm{k}+\mathrm{n_u}}]$	$\}\rightarrow$	Future control sequence
$\mathbf{u}_\mathrm{dof}(k)$	$=$	$[\mathbf{u}_{\mathrm{k}+\mathrm{n_u}-1} \ \dots \ \mathbf{u}_{\mathrm{k}}]$		

being n_b, past horizon for inputs; n_a, past horizon for outputs; and n_u, control horizon. The details on how to obtain $f(.)$ depend on the identification strategy and/or model used. The following subsection explains how to obtain a linear model to be used as the predictor.

3.2.1. Obtaining a linear predictor

Assuming a linear structure, the predictor in Equation (3.1) yields:

$$\hat{\mathbf{y}}_{\mathrm{f}}(k) = \underbrace{[\mathbf{u}_{\mathrm{p}}(k) \quad \mathbf{y}_{\mathrm{p}}(k) \quad \mathbf{u}_{\mathrm{f}}(k) \quad \mathbf{u}_{\mathrm{dof}}(k)]}_{\mathbf{x}(k)}\boldsymbol{\theta} \tag{3.2}$$

where $\boldsymbol{\theta}$ is the matrix containing the parameters of the linear predictor, and $\mathbf{x}(k)$ is the regression row vector at instant k.

For an identification data set, the following matrices can be formed:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{\mathrm{f}}(1) \\ \vdots \\ \mathbf{y}_{\mathrm{f}}(N) \end{bmatrix} \quad ; \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}(1) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}$$

where each of the row vectors is obtained as follows

$$\mathbf{y}_{\mathrm{f}}(k) = [\mathbf{y}_{\mathrm{k+1}} \ \cdots \ \mathbf{y}_{\mathrm{k+n_f}}]$$

$$\mathbf{x}(k) = [\underbrace{\mathbf{u}_{\mathrm{k-1}} \ \cdots \ \mathbf{u}_{\mathrm{k-n_b+1}}}_{\mathbf{u}_{\mathrm{p}}(k)}, \underbrace{\mathbf{y}_{\mathrm{k-1}} \ \cdots \ \mathbf{y}_{\mathrm{k-n_a}}}_{\mathbf{y}_{\mathrm{p}}(k)}, \underbrace{\mathbf{u}_{\mathrm{k+n_f-1}} \ \cdots \ \mathbf{u}_{\mathrm{k+n_u}}}_{\mathbf{u}_{\mathrm{f}}(k)}, \underbrace{\mathbf{u}_{\mathrm{k+n_u-1}} \ \cdots \ \mathbf{u}_{\mathrm{k}}}_{\mathbf{u}_{\mathrm{dof}}(k)}]$$

$\forall k \in [1 \ldots N]$. Note $\mathbf{y}_{\mathrm{f}}(k)$ is a row vector with $n_{\mathrm{y}} \triangleq n_{\mathrm{o}} n_{\mathrm{f}}$ columns, and $\mathbf{x}(k)$ is a row vector with $n_{\mathrm{x}} \triangleq n_{\mathrm{i}}(n_{\mathrm{b}} - 1) + n_{\mathrm{o}} n_{\mathrm{a}} + n_{\mathrm{i}} n_{\mathrm{f}}$ columns. Then $\mathbf{Y} \in \mathbb{R}^{N \times n_{\mathrm{y}}}$, and $\mathbf{X} \in \mathbb{R}^{N \times n_{\mathrm{x}}}$.

Assuming there is process-model mismatch:

$$\mathbf{Y} = \underbrace{\mathbf{X}\boldsymbol{\theta}}_{\triangleq \hat{\mathbf{Y}}} + \mathbf{F}$$

where $\mathbf{F}$ contains the differences between predicted and real outputs in the prediction horizon, and $\hat{\mathbf{Y}}$ is the matrix of predictions. Given the matrices $\mathbf{X}$ and $\mathbf{Y}$, the simplest identification approach is to use LS (least squares) to fit the model:

$$\boldsymbol{\theta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}.$$

Summing up, from an identification data set, one can form matrices $\mathbf{X}$ and $\mathbf{Y}$ and obtain the matrix of parameters $\boldsymbol{\theta}$ of the linear predictor in Equation (3.2).

3.3. Cost function

A general expression for the performance index (cost function) at a given instant is

$$J_{\mathrm{C}} = g(\mathbf{r}_{\mathrm{f}}(k), \hat{\mathbf{y}}_{\mathrm{f}}(k), \mathbf{u}_{\mathrm{dof}}(k)) \tag{3.3}$$

where $\mathbf{r}_{\mathrm{f}}(k)$ is the set point and is defined accordingly to $\hat{\mathbf{y}}_{\mathrm{f}}(k)$.

Provided linear models are often used [Blet 02, Qin 03], and the minimization of a quadratic function with constraints is mathematically tractable, the typical cost function in MPC is:

$$J_{\mathrm{C}} = ||[\mathbf{r}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)]\mathbf{W}_{\mathrm{y}}||_F^2 + \lambda_u ||\Delta\mathbf{u}_{\mathrm{dof}}(k)\mathbf{W}_{\mathrm{u}}||_F^2 \tag{3.4}$$

being $\mathbf{W}_{\mathrm{y}}$ and $\mathbf{W}_{\mathrm{u}}$ positive definite diagonal matrices of appropriate dimensions to weight predicted deviations from the set point and control actions respectively. Whereas $\mathbf{W}_{\mathrm{u}}$ weights the different control actions relatively to other control actions, the scalar λ_u is the overall weight for all the control actions in the cost function, then trades tracking error for control effort. Increments of the control action are often used instead of the control action itself to attain offset-free tracking [Rossiter 03]. The cost function J_{C} has two terms:

- Sum of squared predicted tracking errors in the prediction horizon n_{f}
- Sum of squared increments of the control action in the control horizon n_{u}.

J_C however, does not consider descriptive parameters of the dynamic evolution of the CVs such as settling time or overshoot. A straight approach to consider the dynamic evolution of the CVs whilst minimizing J_C, is to prefilter the reference. $\mathbf{r}_f(k)$ in Equation (3.4) may be replaced by

$$\mathbf{w}_f(k) = F(z)\mathbf{r}_f(k)$$

The dynamic behaviour of the filtered reference, $\mathbf{w}_f$, is set as the desired dynamic behaviour of the CVs.

3.4. Optimizer

The minimization of the cost function in Eq. (3.3) subject to inequality constraints for either the MVs or the CVs can be expressed:

$$\min_{\Delta \mathbf{u}_{dof}(k)} J_C(k) \quad s.t. \quad constraints \tag{3.5}$$

In the general case an optimizer need be used on-line to solve the minimization problem and obtain the control sequence $\mathbf{u}_{dof}(k)$. This minimization problem can be solved using QP (Quadratic Programming) if the model is linear, and the cost function in Eq. (3.4) is used. Provided a QP can be solved with a reasonable computational complexity, linear models and the quadratic cost function are normally used in Industry for MPC implementations.

If constraints are not needed, the model is linear, and the cost function in Eq. (3.4) is to be minimized; then, an analytical expression to obtain $\mathbf{u}_{dof}(k)$ can be computed off-line. This can be a solution for systems with fast dynamics in which constraints are not crucial. In general however, constraints are very important and a QP is often solved on-line.

CHAPTER 4

LATENT VARIABLE METHODS

4.1. Introduction

In large-scale manufacturing processes such as chemical, food, or steel making processes, there are a large number of CVs (Controlled Variables) and MVs (Manipulated Variables). Due to the multivariate nature of the data, variables are highly correlated, and the effective dimension of the space in which they move is very small. Consequently, problems appear when fitting mathematical models to a size-limited identification data set.

In the event of correlation in the identification data set, LVMs (Latent Variable Methods) are often used [Kiers 07]. LVMs transform noisy and correlated data into a smaller informative set in which identification can be successfully performed. Additionally to identification in the case of correlation in the data set, LVMs are successfully used to:

- Control batch processes [Flores-Cerrillo 05]
- Perform process control without on-line measurement of quality variables [Chen 98, McAvoy 02]
- Create soft-sensors [Kano 09]
- Detect process faults and abnormalities [Kourti 05]
- Perform DDQI (Data-Driven Quality Improvement) [Kano 08]
- Deal with missing values in the data set [Nelson 96]
- Monitor the condition of an industrial MPC system based on data collected from an industrial process [AlGhazzawi 09]

This chapter provides a brief understanding of LVMs as a tool for model identification, and introduces PLS as a LVM that determines the latent variables by maximizing the correlation between the input and output scores.

4.2. Identification in the space of the latent variables

As shown in Subection 3.2.1, least squares can be used to obtain the matrix of parameters $\boldsymbol{\theta}$ in

$$\mathbf{Y} = \underbrace{\mathbf{X}\boldsymbol{\theta}}_{\triangleq\hat{\mathbf{Y}}} + \mathbf{F}$$

as

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

However, in the event of correlation in the identification data set, matrix $(\mathbf{X}^T\mathbf{X})$ is ill-conditioned, hence fitting $\boldsymbol{\theta}$ by means of LS is not successful causing a large variance error. This section explains how the predictor can be obtained performing the regression in the latent variable space.

Given $\mathbf{Y}$ and $\mathbf{X}$, variable reduction can be performed as follows:

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathbf{T}} + \mathbf{F} \tag{4.1}$$

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \tag{4.2}$$

where $\mathbf{Y} \in \mathbb{R}^{(N \times n_y)}$, output space; $\mathbf{U} \in \mathbb{R}^{(N \times n_{\mathrm{lv}})}$, output scores; $\mathbf{Q} \in \mathbb{R}^{(n_y \times n_{\mathrm{lv}})}$, output loadings; $\mathbf{F} \in \mathbb{R}^{(N \times n_y)}$, output residuals; $\mathbf{X} \in \mathbb{R}^{(N \times n_x)}$, input space; $\mathbf{T} \in \mathbb{R}^{(N \times n_{\mathrm{lv}})}$, input scores; $\mathbf{P} \in \mathbb{R}^{(n_x \times n_{\mathrm{lv}})}$, input loadings; $\mathbf{E} \in \mathbb{R}^{(N \times n_x)}$, input residuals; and n_{lv}, number of latent variables.

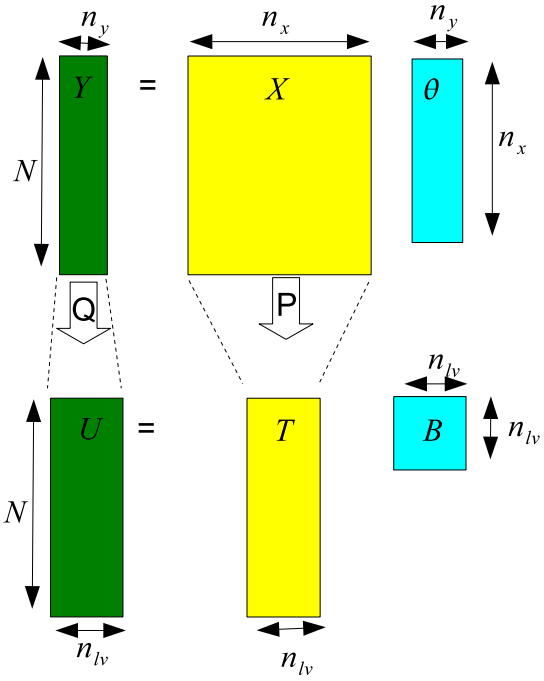


Figure 4.1: LVMs: projection onto the latent variable space.

As shown in Figure 4.1, the first step in latent variable methods is to project the original space onto the reduced latent variable space. Note the regression matrix in the outer space has n_{x} columns whereas it has n_{lv} columns in the latent variable space. n_{lv} is a design choice and is normally set below n_{x} assuming there is collinearity in the identification data set.

The latent variable space is defined by scores $\mathbf{T}$ and $\mathbf{U}$. The model in the inner space can be obtained

$$\mathbf{U} = \underbrace{\mathbf{TB}}_{\triangleq\hat{\mathbf{U}}} + residues \quad \overset{LS}{\Longrightarrow} \mathbf{B} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{U}. \tag{4.3}$$

Note that if identification is performed in the outer space, $\boldsymbol{\theta}$ is fit using $\mathbf{Y}$ and $\mathbf{X}$; then, correlation among column vectors in $\mathbf{X}$ affects the fitting. However, if identification is performed in the inner space, $\mathbf{B}$ is fit using $\mathbf{U}$ and $\mathbf{T}$; and column vectors in $\mathbf{T}$ are orthogonal; thus, fitting $\mathbf{B}$ is not affected by correlation among column vectors in $\mathbf{X}$ thanks to the projection onto the inner space.

Performing the identification in the latent variable space can additionally improve the MSEP (Mean Squared Error of Prediction)

$$MSEP = \frac{1}{N}\sum_{k=1}^{N} ||\mathbf{y}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)||_{\mathrm{F}}^2$$

Given a data set and a predictor, one can calculate MSEP. To validate a predictor by means of MSEP, a validation data set different to that used for identification is often used. Alternatively, one can use the identification data set and perform crossed-validation [Tropsha 03].

According to [Höskuldsson 88], MSEP can be split into the sum of squared bias error and residual variance error. Bias and variance errors are related to both the number of parameters in the model, and the size of the identification data set. For a given identification data set, the number of parameters in the model determines the solution in the bias-variance trade-off. From Figure 4.1, the size of $\boldsymbol{\theta}$ is $n_{\mathrm{x}} \times n_{\mathrm{y}}$, and the size of $\mathbf{B}$ is $n_{\mathrm{lv}} \times n_{\mathrm{lv}}$, then setting n_{lv} provides a mean to improve the MSEP. However, $\mathbf{P}$ and $\mathbf{Q}$ should also be considered as they are also fitted to the identification data set, and this affects in the reduction of the number of parameters to fit.

In order to set n_{lv}, MSEP is evaluated for each value of n_{lv}. For small values of n_{lv} and as it increases, MSEP decreases. n_{lv} is often chosen as the minimum value which provides a significant drop in MSEP.

Summing up, if identification is performed in the space of the latent variables: $\mathbf{Y}$ and $\mathbf{X}$ are projected onto a reduced n_{lv}-space, and the model is fit in that inner space. There are different approaches to define the projection: PLS, PCR, RRR, PCovR, and Power Regression [Kiers 07]. According to the comparison of the various methods in [Kiers 07], PLS and PCR are particularly indicated in cases with much collinearity, whereas in other cases it is better to use ordinary regression. In this thesis $\mathbf{X}$ has many columns, specially for multivariate processes and large control horizons, then collinearity appears and use of PLS or PCR is of interest.

In PLS the projection is performed so that the correlation between the input and output scores is maximized, whilst PCR aims at maximizing the explained variance of $\mathbf{X}$. Although, PCR could have been used in this thesis, PLS has been used provided it performs the projection taking into consideration both the input and the output space. The PLS approach is introduced in the following section.

4.3. PLS

The different approaches to determine $\mathbf{P}$ and $\mathbf{Q}$ in Equations (4.2) and (4.1) give name to the different LVMs. In PLS, latent variable i is obtained such that correlation between input scores $\mathbf{t}_{\mathrm{i}}$ and output scores $\mathbf{u}_{\mathrm{i}}$ is maximized. Note $\mathbf{t}_{\mathrm{i}}$ is column i in $\mathbf{T}$, and $\mathbf{u}_{\mathrm{i}}$ column i in $\mathbf{U}$.

The PLS algorithm is thoroughly introduced in [Höskuldsson 88, Geladi 86], in which an additional matrix $\mathbf{W}$ is used in place of $\mathbf{P}$ in the modified NIPALS algorithm to obtain $\mathbf{T}$ with orthogonal column vectors. As explained

in [Martens 01], $\mathbf{T}$ can be obtained from $\mathbf{X}$:

$$\mathbf{T} = \mathbf{X}\underbrace{\mathbf{W}(\boldsymbol{P}^T\mathbf{W})^{-1}}_{\triangleq \mathbf{Z}}. \tag{4.4}$$

From Equations (4.2) (4.1) (4.3) (4.4), the model in the outer space $\boldsymbol{\theta}$ can be obtained using PLS as:

$$\hat{\mathbf{Y}} = \hat{\mathbf{U}}\mathbf{Q}^T = \mathbf{T}\mathbf{B}\mathbf{Q}^T = \mathbf{X}\underbrace{\mathbf{Z}\mathbf{B}\mathbf{Q}^T}_{\triangleq \boldsymbol{\theta}} \tag{4.5}$$

$$\boldsymbol{\theta} = \mathbf{Z}\mathbf{B}\mathbf{Q}^T \tag{4.6}$$

The PLS model is finally defined by:

- Model in the outer space: $\boldsymbol{\theta}$
- Projection matrices: $\mathbf{P}$, $\mathbf{W}$, and $\mathbf{Q}$
- Model in the inner space: $\mathbf{B}$

which matrices to use depends on what task the model is to be used for. If the task is only to perform predictions, just the model in the outer space, $\boldsymbol{\theta}$, is needed. In the control methodology proposed in **Part III**, all the matrices are used since predictions and optimization are performed in the inner space.

Part II

MRI: Model predictive control Relevant Identification

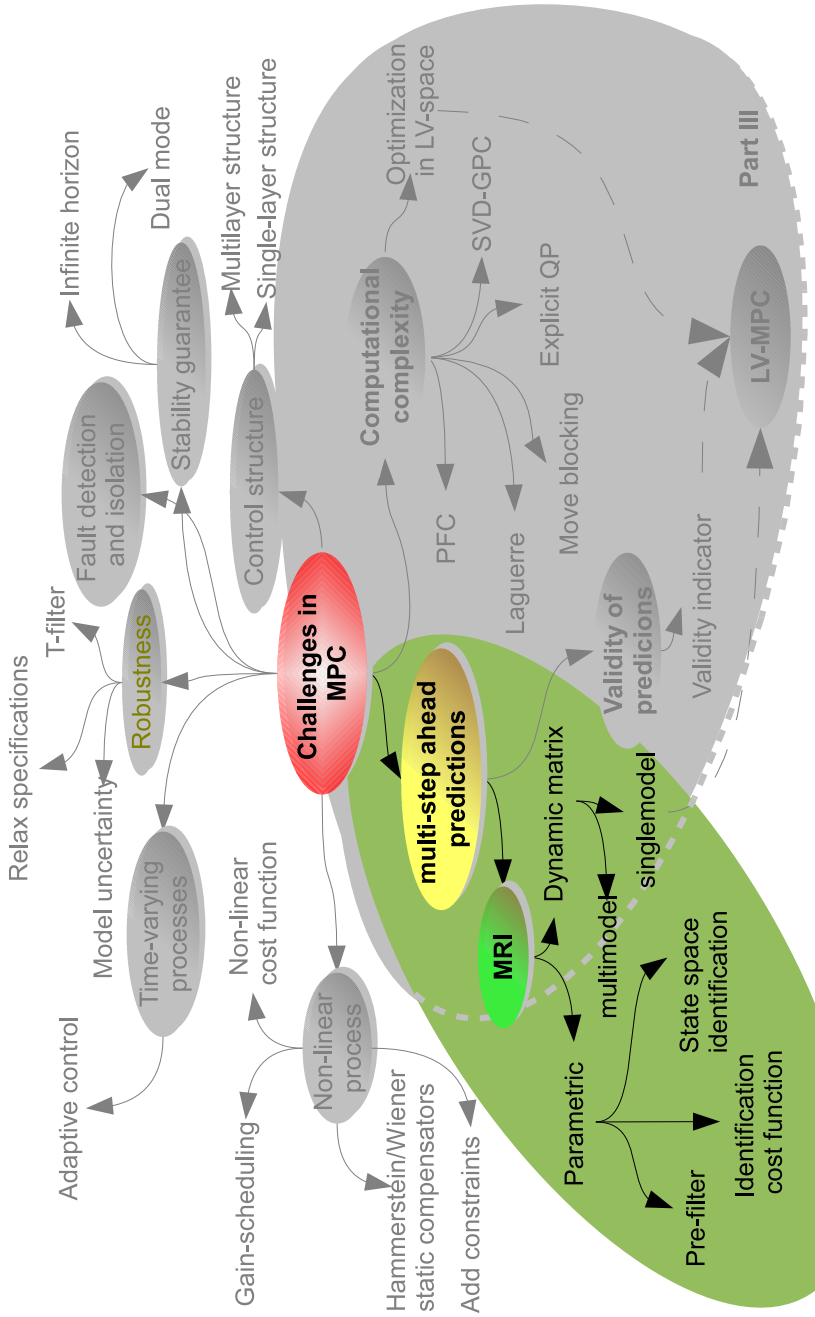


Figure 4.2: Location of PartII in the general mindmap in Figure 1.4.

CHAPTER 5

MODEL PREDICTIVE CONTROL RELEVANT IDENTIFICATION

5.1. Introduction

Model development is by far the most critical and time-consuming step in implementing a model predictive controller [Morari 99, Zhu 02, Bars 06]. There are different model structures to approximate a process, which one to choose is determined by the nature of the process and the purpose of its use. A precise model of a process may be used for simulating its behaviour. If the target is to design a controller however, a simplified model is often preferable provided it simplifies the process of designing the controller and the resulting implementation of the control strategy. In this thesis, the models used are LTI (linear time invariant):

Linear All industrial processes are known to be non-linear, however, around a working point they can often be approximated by a linear model. Local linear models are used in this thesis, thus the subsequent control can only be applied whenever the process is around the working point for which the model has been identified. Start-up, shut-down and alarm strategies are also needed for a real implementation of control in an industrial process. This thesis however, focuses on operation around a working point.

Time invariant Most industrial processes change with time. The expression to update model parameters with time is not often known, then process re-identification is often implemented. Either periodically as in adaptive control or when the model performance deteriorates. In this thesis processes are assumed to be time invariant and in the event of changes tuning of the model is assumed to be implemented.

Apart from the structure of the model, it is important to decide how to fit the parameters of the model. Traditional system identification aims at minimizing the bias and variance errors to fit the model. CRI (Control relevant Identification) however, aims at obtaining a model that suits the control problem at hand [Rivera 92]; so creating synergy between the identification and control algorithms. The acronym MRI (Model predictive control Relevant Identification) refers to CRI in the case of an MPC controller. Two branches to implement MRI can be distinguished Figure 4.2:

- **The parametric model approach**: A one-step ahead model is fit taking into account the further use of the model to perform predictions in a given prediction window. The parametric model is then used in MPC to obtain the dynamic matrix needed to perform multi-step ahead predictions
- **The dynamic matrix approach**: The dynamic matrix to perform multi-step ahead predictions are obtained directly from process data.

As shown in Figure 4.2, there are three ways to attain MRI in the parametric branch:

- Minimize a multi-step ahead prediction error cost function
- Pre-filter the identification data set
- Use subspace identification methods (SIMs)

In traditional identification, a one-step ahead prediction error cost function is minimized. In MPC, the model needs to perform predictions not only one-step ahead, but in a prediction window. A straight approach for MRI is then to minimize a multi-step ahead prediction error cost function to fit the model to the identification data set. In [Gopaluni 02a, Gopaluni 04], the multi-step ahead prediction error cost function is minimized, and the properties of MRI are analysed demonstrating the potential value of the multi-step ahead prediction error cost function when there is model-plant structural mismatch, which is normal in the process Industry. One disadvantage of this approach is the identification cost function is non-linear in its parameters, and the minimization problem is non-convex in general; thus, non-convex optimization is needed to obtain the model. However, computational complexity in non-convex optimization may not be a problem provided model identification is performed off-line or on-line in adaptive control, but with a slower sampling rate than the controller.

As proven in [Gopaluni 04, Shook 92], minimizing the multi-step ahead cost function is equivalent to using an approach based on prefiltering in which the noise model is assumed to be a known LTI (Linear Time Invariant) model. In a real application however, the noise model is unknown and may not be LTI. Besides, although Shook et al. state their conclusions also apply to the MIMO case, the algorithm in [Shook 91] can only deal with the SISO (Single Input Single Output) case.

In subspace identification, the dynamic matrix to perform multi-step ahead predictions is first fitted to the identification data set; and then the parametric model is fitted to the dynamic matrix [Huang 08]. In MPC, the model is expanded to obtain the dynamic matrix to perform multi-step ahead predictions. Although SIMs are a parametric MRI approach, the intermediate step is based in the dynamic matrix approach.

As shown in Fig. 4.2, there are two ways to attain MRI in the dynamic matrix branch:

- **Multimodel**: As many models as the length of the prediction horizon are obtained [Rossiter 01]. Although each of the models is obtained using the one-step ahead approach, the global prediction model is commensurate with the control cost function for it minimizes the multi-step ahead identification error.
- **Singlemodel**: The whole dynamic matrix is directly fitted to the identification data set in one step. This approach is used in the control methodology proposed in part **III** in this thesis, and is also used in the data-driven subspace approach in [Huang 08].

Figure 5.1 helps understand the different MRI approaches by comparing them in a common framework. The aim is to obtain the dynamic matrix (DM) of the predictor in the upper-right part of the figure. Note the model is LTI, then the predictor is a matrix which multiplies a vector of known data to provide predictions.

The two approaches on the left side represent parametric MRI. In parametric MRI the one-step ahead model, $\boldsymbol{\theta}$, is obtained and then expanded to obtain the dynamic matrix of the predictor (DM). In the upper-left part the data set is filtered and used to form the matrices, $\mathbf{Y}$ in green and $\mathbf{X}$ in yellow, that serve to fit the parameters of the model. In the lower-left part the minimization of the multi-step ahead cost function is depicted.

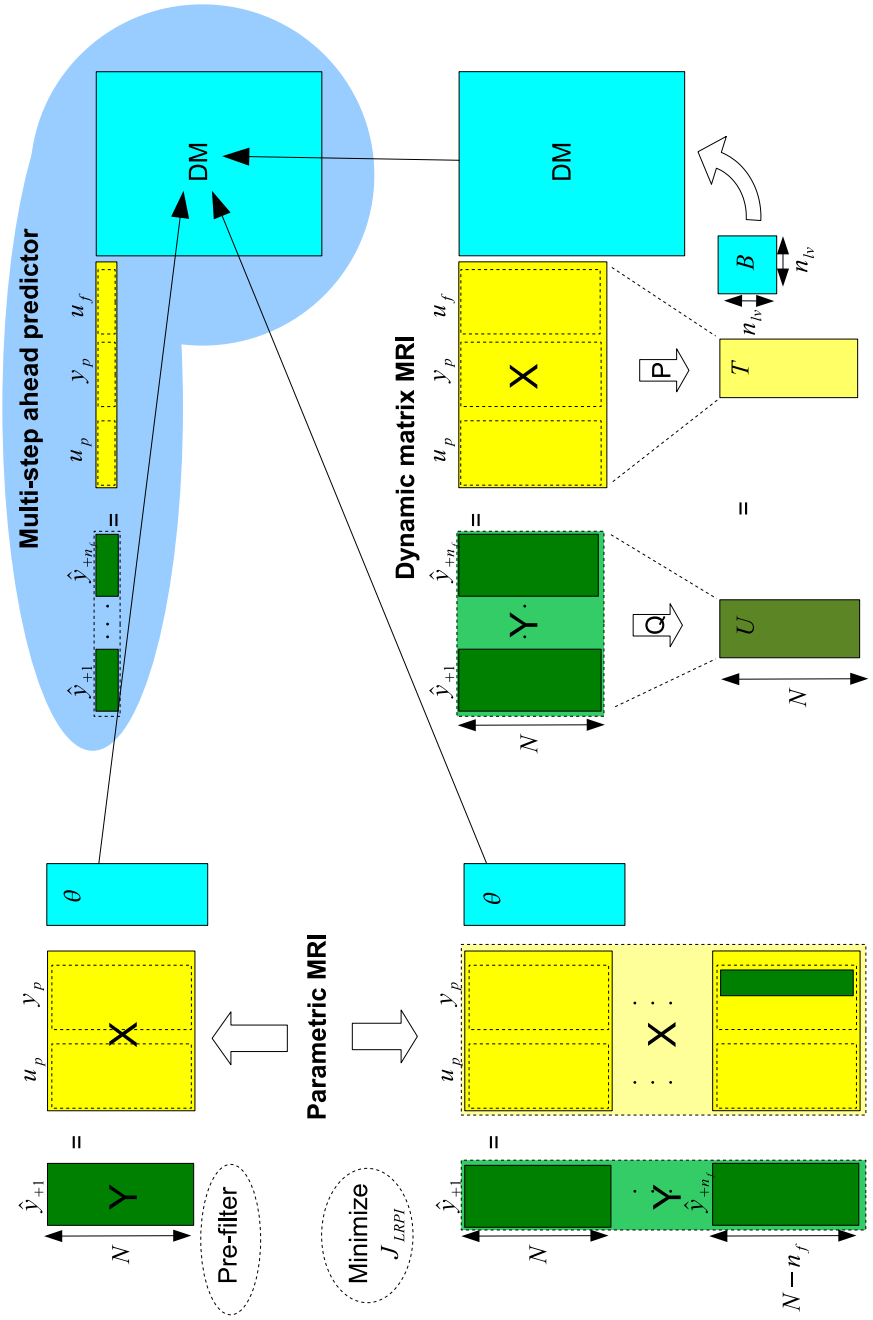


Figure 5.1: Model predictive control relevant identification.

To depict the multi-step ahead minimization procedure n_f blocks of samples are used. The first green block contains predictions at $k+1$, and the latter green block contains predictions at $k+n_\mathrm{f}$. Note predictions at $k+n_\mathrm{f}$ depend on predictions at $k+i, \quad \forall i \in [1, n_\mathrm{f}-1]$, then green blocks appear also on $\mathbf{X}$, which themselves depend on $\boldsymbol{\theta}$, hence this structure is non-linear in $\boldsymbol{\theta}$. Comparing the two parametric approaches on the left side, one can see the prefilter option has identification matrices with N samples, whereas the multi-step ahead minimization approach has $n_\mathrm{f}N$ samples[1].

The two approaches on the right side represent the dynamic matrix approach, then the dynamic matrix is obtained directly from identification data. In the middle-right part the dynamic approach using LS on $\mathbf{Y}$ and $\mathbf{X}$ is depicted. In the lower-right part the projection is performed, the dynamic matrix is fitted in the space of its latent variables $\mathbf{B}$, and then the DM matrix is obtained.

This Chapter is structured in two sections: 5.2 describes the parametric model approach for MRI, and 5.3 the dynamic matrix approach. In each section firstly the structure of the model is defined and secondly a fitting procedure.

5.2. MRI: parametric approach

In this section first the model structure is defined, then the predictor is obtained, and the multistep-ahead function is defined and minimized.

[1]The number of samples is actually $\sum_{i=0}^{n_\mathrm{f}-1} N-i$, but assuming $n_\mathrm{f} << N$, the previous expression can be approximated by $n_\mathrm{f}N$.

5.2.1. Model

Let us start from the SISO Box–Jenkins model as the general expression for an LTI (Linear Time Invariant) model in the transfer function form:

$$y(k) = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})}u(k) + \frac{\mathbf{C}(z^{-1})}{\mathbf{D}(z^{-1})}\xi(k)$$

Where: $y(k)$, process output; $u(k)$, process input; and $\xi(k)$, white noise.

Since the estimation of $\mathbf{C}(z^{-1})$ is frequently unsuccessful [Shook 91], and the disturbance characteristics change very frequently [Qin 06], it is common practice in MPC to use the CARIMA model [Rossiter 03]:

$$\mathbf{C}(z^{-1}) = \mathbf{T}(z^{-1}) \quad ; \quad \mathbf{D}(z^{-1}) = \mathbf{A}(z^{-1})(1 - z^{-1}) \tag{5.1}$$

Where $\mathbf{T}(z^{-1})$ is considered as a design choice by the control engineer. The SISO model can be transformed into the ARX form:

$$y(k) = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})}u(k) + \frac{\mathbf{T}(z^{-1})}{\mathbf{A}(z^{-1})(1 - z^{-1})}\xi(k)$$

$$\frac{(1 - z^{-1})}{\mathbf{T}(z^{-1})}y(k) = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})}\frac{(1 - z^{-1})}{\mathbf{T}(z^{-1})}u(k) + \frac{1}{\mathbf{A}(z^{-1})}\xi(k)$$

$$y_{\mathrm{f}}(k) = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})}u_{\mathrm{f}}(k) + \frac{1}{\mathbf{A}(z^{-1})}\xi(k).$$

Providing the identification data is previously filtered, an ARX structure can be assumed.

$$y_{\mathrm{f}}(k) = \frac{1 - z^{-1}}{\mathbf{T}(z^{-1})}y(k) \quad ; \quad u_{\mathrm{f}}(k) = \frac{1 - z^{-1}}{\mathbf{T}(z^{-1})}u(k)$$

In the sequel, the ARX structure is used, being $u(k)$ the process input and $y(k)$ the process output; or a filtered version if the CARIMA model is considered. Appendix A shows the equivalence between the ARX structure and the following expression for the MIMO case:

$$\mathbf{y}_{\mathrm{k}} = \mathbf{x}_{\mathrm{k}-1}\boldsymbol{\theta} + \boldsymbol{\xi}_k \tag{5.2}$$

Where:

- $\mathbf{y}_{\mathrm{k}} = [y_1(k) \dots y_{\mathrm{n_o}}(k)]$
- $\mathbf{x}_{\mathrm{k-1}} = [\mathbf{u}_{\mathrm{k-1}} \dots \mathbf{u}_{\mathrm{k-n_b}}, \mathbf{y}_{\mathrm{k-1}} \dots \mathbf{y}_{\mathrm{k-n_a}}]$, is the regression vector where:
  - $\mathbf{u}_{\mathrm{k}} = [u_1(k) \dots u_{\mathrm{n_i}}(k)]$
  - n_{b}: number of lagged inputs in the model
  - n_{a}: number of lagged outputs in the model
- $\boldsymbol{\theta}$ is the matrix of parameters that defines the model. Its size is $n_{\mathrm{x}} \times n_{\mathrm{o}}$ (being n_{x} the number of columns in $\mathbf{x}_{\mathrm{k}}$, $n_{\mathrm{x}} = n_{\mathrm{i}} n_{\mathrm{b}} + n_{\mathrm{o}} n_{\mathrm{a}}$)
- $\boldsymbol{\xi}_k = [\xi_1(k) \dots \xi_{\mathrm{n_o}}(k)]$

5.2.2. Predictor

One-step ahead predictor

If the estimated model is defined by $\boldsymbol{\theta}$, then prediction of the outputs at instant $k+1$ with information available up to instant k is obtained as:

$$\hat{\mathbf{y}}_{\mathrm{k+1}} = \mathbf{x}_{\mathrm{k}} \boldsymbol{\theta} \tag{5.3}$$

Where:

$$\mathbf{x}_{\mathrm{k}} = [\mathbf{u}_{\mathrm{k}} \dots \mathbf{u}_{\mathrm{k+1-n_b}}, \mathbf{y}_{\mathrm{k}} \dots \mathbf{y}_{\mathrm{k+1-n_a}}]$$

Should there be model-plant mismatch, $\boldsymbol{\xi}(k)$ in Equation (5.2) is no longer white noise, and the real process outputs are stated as:

$$\mathbf{y}_{\mathrm{k+1}} = \hat{\mathbf{y}}_{\mathrm{k+1}} + \mathbf{e}_{\mathrm{k+1}}$$

Being $\mathbf{e}_{\mathrm{k+1}}$ the identification error at instant $k+1$ with output data available up to instant k.

Multi-step ahead predictor

Turning to the expression in Equation (5.3), the multi-step ahead prediction model with outputs available up to instant k, and inputs up to instant $k+n_{\mathrm{f}}$ can be formulated as:

$$\hat{\mathbf{y}}_{\mathrm{k+j}} = \mathbf{x}_{\mathrm{k+j-1}}\boldsymbol{\theta}, \qquad \forall j \in [1, 2, \ldots, n_{\mathrm{f}}] \tag{5.4}$$

Where:

$$\mathbf{x}_{\mathrm{k+j-1}} = [\mathbf{u}_{\mathrm{k+j-1}} \ldots \mathbf{u}_{\mathrm{k+j-n_b}}, \bar{\mathbf{y}}_{\mathrm{k+j-1}} \ldots \bar{\mathbf{y}}_{\mathrm{k+j-n_a}}] \tag{5.5}$$

$$\bar{\mathbf{y}}_{\alpha} = \begin{cases} \hat{\mathbf{y}}_{\alpha} & for \quad \alpha > k \\ \mathbf{y}_{\alpha} & for \quad \alpha \leq k \end{cases}$$

$\hat{\mathbf{y}}_{\mathrm{k+n_f}}$ can be obtained recursively starting at $\hat{\mathbf{y}}_{\mathrm{k+1}}$, and using Equation (5.4). The real outputs can be expressed as:

$$\mathbf{y}_{\mathrm{k+j}} = \hat{\mathbf{y}}_{\mathrm{k+j}} + \mathbf{e}_{\mathrm{k+j}} \tag{5.6}$$

Being $\mathbf{e}_{\mathrm{k+j}}$ the identification error at instant $k+j$ with output data available up to instant k.

5.2.3. Minimize the multi-step ahead cost function

The MRI parametric approach by minimizing the multi-step ahead cost function is explained in detail in this subsection.

Lets take the basic cost function normally used in MPC from Equation (3.4):

$$J_{\mathrm{C}} = ||[\mathbf{r}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)]\mathbf{W}_{\mathrm{y}}||_F^2 + \lambda_u||\Delta\mathbf{u}_{\mathrm{dof}}(k)\mathbf{W}_{\mathrm{u}}||_F^2$$

In order to simplify the analysis lets assume all outputs are given the same weight, then $\mathbf{W}_{\mathrm{y}} = \mathbf{I}$; and lets assume the overall weight for control actions is taken to be zero, $\lambda_u = 0$. The previous equation yields

$$J_{\mathrm{C}} = ||\mathbf{r}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)||_F^2$$

which can be reformulated as

$$J_{\mathrm{C}} = \sum_{j=1}^{n_{\mathrm{f}}} ||\underbrace{\mathbf{r}_{\mathrm{k+j}} - \hat{\mathbf{y}}_{\mathrm{k+j}}}_{\triangleq \mathbf{c}_{\mathrm{k+j}}}||_{\mathrm{F}}^2 \tag{5.7}$$

In this case MPC minimizes the difference between predictions and their reference.

Assuming there is model-process mismatch, instead of minimizing the deviation of predictions from their reference, one can think to minimize the deviation of the actual values of the outputs from their references:

$$\tilde{J}_{\mathrm{C}} \triangleq \sum_{j=1}^{n_{\mathrm{f}}} ||\mathbf{r}_{\mathrm{k+j}} - \mathbf{y}_{\mathrm{k+j}}||_{\mathrm{F}}^2$$

Substituting in from Equation (5.6), the cost function can be expressed as a function of the control and identification errors:

$$\begin{aligned}
\tilde{J}_{\mathrm{C}} &= \sum_{j=1}^{n_{\mathrm{f}}} ||\mathbf{r}_{\mathrm{k+j}} - (\hat{\mathbf{y}}_{\mathrm{k+j}} + \mathbf{e}_{\mathrm{k+j}})||_{\mathrm{F}}^2 && (5.8)\\
&= \sum_{j=1}^{n_{\mathrm{f}}} ||\underbrace{\mathbf{r}_{\mathrm{k+j}} - \hat{\mathbf{y}}_{\mathrm{k+j}}}_{\mathbf{c}_{\mathrm{k+j}}} - \mathbf{e}_{\mathrm{k+j}}||_{\mathrm{F}}^2 && (5.9)\\
&= \sum_{j=1}^{n_{\mathrm{f}}} ||\mathbf{c}_{\mathrm{k+j}} - \mathbf{e}_{\mathrm{k+j}}||_{\mathrm{F}}^2 && (5.10)
\end{aligned}$$

which can be rewritten as:[2]

$$\begin{aligned}
\tilde{J}_{\mathrm{C}} &= \sum_{j=1}^{n_{\mathrm{f}}} tr(\mathbf{c}_{\mathrm{k+j}}^{\mathrm{T}}\mathbf{c}_{\mathrm{k+j}}) && (5.11a)\\
&\quad + \sum_{j=1}^{n_{\mathrm{f}}} tr(\mathbf{e}_{\mathrm{k+j}}^{\mathrm{T}}\mathbf{e}_{\mathrm{k+j}}) && (5.11b)\\
&\quad - \sum_{j=1}^{n_{\mathrm{f}}} tr(\mathbf{c}_{\mathrm{k+j}}^{\mathrm{T}}\mathbf{e}_{\mathrm{k+j}} + \mathbf{e}_{\mathrm{k+j}}^{\mathrm{T}}\mathbf{c}_{\mathrm{k+j}}) && (5.11c)
\end{aligned}$$

[2] The Frobenius norm can be expressed using the trace operator: $||\mathbf{A}||_{\mathrm{F}}^2 = tr(\mathbf{A}^{\mathrm{T}}\mathbf{A})$.

Term (5.11a) is equal to the cost function minimized in MPC obtained in Equation (5.7). Term (5.11c) represents the cross correlation between identification and control errors, and according to [Gevers 02] this can be handled using iterative tuning. This is, however, outside the scope this thesis.

Term (5.11b) is the most relevant part of $\tilde{J}_{\mathrm{C}}$ for this section as it depends on the identification error, and it will now be referred to as the **L**ong **R**ange **P**rediction **I**dentification cost index (J_{lrpi}). The **O**ne **S**tep **A**head **P**rediction **I**dentification cost index (J_{osapi}), normally used in PEMs (Prediction Error Methods) such as LS (Least Squares), is obtained by setting $n_{\mathrm{f}} = 1$.

$$J_{\mathrm{lrpi}}(k) = \sum_{j=1}^{n_{\mathrm{f}}} ||\mathbf{e}_{\mathrm{k+j}}||_{\mathrm{F}}^2 \tag{5.12}$$

$$J_{\mathrm{osapi}}(k) = ||\mathbf{e}_{\mathrm{k+1}}||_{\mathrm{F}}^2 \tag{5.13}$$

It can be seen that the former cost function contains the latter plus some additional terms, and so J_{osapi} is an incomplete cost index if the model is to be used in an MPC framework. MRI was thoroughly studied for the SISO case in [Gopaluni 04] and it was shown that a better control performance can be obtained if J_{lrpi} instead of J_{osapi} is minimized in the estimation of the parameters of the model. According to [Shook 91], the model fitted minimizing J_{lrpi} is better than that fitted minimizing J_{osapi}. This is true providing there are unmodelled dynamics, and/or the identification data set is not sufficiently exciting — which is normal in an Industrial application. It is also proven in [Gopaluni 02b] that any experiment that is informative enough for the one-step ahead approach, is also informative enough for the multi-step ahead approach. Hence, if the model is to be used for MPC, the index to minimize in the identification stage is J_{lrpi}.

The index can be evaluated for a complete data set:

$$J_{\mathrm{LRPI}} = \sum_{k=1}^{N} J_{\mathrm{lrpi}}(k) = \sum_{k=1}^{N} \sum_{j=1}^{n_{\mathrm{f}}} ||\mathbf{e}_{\mathrm{k+j}}||_{\mathrm{F}}^2 \tag{5.14}$$

Where:

$$\mathbf{e}_{\mathrm{k+j}} = \mathbf{y}_{\mathrm{k+j}} - \mathbf{x}_{\mathrm{k+j-1}}\boldsymbol{\theta}. \tag{5.15}$$

From Equation (5.5) $\mathbf{x}_{\mathrm{k+j-1}}$ contains $\hat{\mathbf{y}}_{\mathrm{k+j-}\beta}, \forall \beta \in [1, 2, \ldots, j-1]$, which itself depends on $\boldsymbol{\theta}$; thus, this problem is non-linear in $\boldsymbol{\theta}$. There is no closed-form solution as in the one-step ahead approach.[3] The Levenberg-Marquardt algorithm implemented in Matlab® can be used to minimize J_{LRPI}.

Summing up, given an identification data set, the matrix of parameters $\boldsymbol{\theta}$ is fitted minimizing J_{LRPI} in Equation (5.14). To calculate J_{LRPI}, the expression for the error in Equation (5.15) and the multi-step ahead predictor in Equation (5.4) are needed. The implemented Matlab® code to obtain $\boldsymbol{\theta}$ that minimizes J_{LRPI} can be downloaded at Matlab® central,[4] after searching for file: MIMO MRI.

5.3. MRI: dynamic matrix approach

In this section first the predictor is defined, and then the procedure to obtain the dynamic matrix of the predictor by means of latent variable methods is explained.

5.3.1. Predictor

Assuming the same linear structure for the model assumed in Section 5.2, the multi-step ahead predictor to be used in the dynamic matrix approach can be

[3] The one-step ahead approach is linear in $\boldsymbol{\theta}$ because only $\mathbf{x}_{\mathrm{k+1}}$ is needed, and it is independent of $\boldsymbol{\theta}$.

[4] http://www.mathworks.com/matlabcentral/fileexchange/24274

written:

$$\hat{\mathbf{y}}_{\mathrm{f}}(k) = [\overbrace{\underbrace{\mathbf{u}_{\mathrm{p}}(k) \quad \mathbf{y}_{\mathrm{p}}(k)}_{\triangleq \mathbf{x}_{\mathrm{p}}(k)} \quad \mathbf{x}_{\mathrm{f}}(k) \quad \mathbf{x}_{\mathrm{dof}}(k)}^{\mathbf{x}(k)}]\boldsymbol{\theta} \tag{5.16}$$

where $\boldsymbol{\theta}$ is the dynamic matrix with appropriate dimensions, and

$$\hat{\mathbf{y}}_{\mathrm{f}}(k) = [\mathbf{y}_{\mathrm{k+1}} \ \cdots \ \mathbf{y}_{\mathrm{k+n_f}}] \tag{5.17}$$

$$\mathbf{u}_{\mathrm{p}}(k) = [\mathbf{u}_{\mathrm{k-1}} \quad \mathbf{u}_{\mathrm{k-n_b+1}}] \tag{5.18}$$

$$\mathbf{y}_{\mathrm{p}}(k) = [\mathbf{y}_{\mathrm{k-1}} \quad \mathbf{y}_{\mathrm{k-n_a}}] \tag{5.19}$$

$$\mathbf{x}_{\mathrm{f}}(k) = [\mathbf{u}_{\mathrm{k+n_f-1}} \ \cdots \ \mathbf{u}_{\mathrm{k+n_u}}] \tag{5.20}$$

$$\mathbf{x}_{\mathrm{dof}}(k) = [\mathbf{u}_{\mathrm{k+n_u-1}} \ \cdots \ \mathbf{u}_{\mathrm{k}}] \tag{5.21}$$

where: n_{f}, prediction horizon; n_{b}, past horizon for inputs; n_{a}, past horizon for outputs; and n_{u}, control horizon. Note $\mathbf{y}_{\mathrm{p}}(k)$ starts at $\mathbf{y}_{\mathrm{k-1}}$ instead of $\mathbf{y}_{\mathrm{k}}$, the reason is computing time of the control action is considered to be approximately one sample time, then if the controller is to decide the control action to apply at k, the output at k is still not available.

5.3.2. Identification in the space of the latent variables

It is common knowledge that if the MVs (Manipulated Variables) are not exciting enough for the structure of the model to fit, correlation among variables in the identification data set appears and the identification cannot be done successfully. In industrial processes, it is not normally possible to excite the MVs as much as desired and thus, the identification procedure needs to account for correlation among variables. Additionally, from [Ljung 99] the asymptotic parameter variance is proportional to the ratio between the number of parameters to estimate and the data length. In the dynamic matrix approach there are more parameters to fit than in the parametric approach, especially in the case of large prediction horizon, compare the size of matrices $\boldsymbol{\theta}$ and DM in Figure 5.1. A large number of parameters to fit can lead to estimated parameters with poor statistical properties.

LVMs cope with collinearity in the data set and perform identification in the reduced latent variable space, then reduce the number of parameters to fit improving statistical properties of the model. In the lower-right side of Figure 5.1 the matrices of data are projected to the latent variable space and matrix $\mathbf{B}$ is fitted. Note the size of matrix $\mathbf{B}$ is smaller than matrix DM if variable reduction is performed, what improves statistical properties of of the fitted model. Additionally, if the identification is performed in the latent variable space, a much more modest identification data set can be used to successfully fit the model [Flores-Cerrillo 05]. More information on identification in the latent variable space can be found in Section 4.2.

Identification data matrices can be obtained from Equation (5.16) for $k \in [1 \dots N]$.

$$\mathbf{Y} = \underbrace{\mathbf{X}\boldsymbol{\theta}}_{\hat{\mathbf{Y}}} + \mathbf{F} \tag{5.22}$$

where $\mathbf{F}$ contains the identification error, and

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{\mathrm{f}}(1) \\ \vdots \\ \mathbf{y}_{\mathrm{f}}(N) \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{\mathbf{y}}_{\mathrm{f}}(1) \\ \vdots \\ \hat{\mathbf{y}}_{\mathrm{f}}(N) \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(1) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}.$$

Identification of $\boldsymbol{\theta}$ is performed in this section by means of PLS introduced in Section 4.3 and the dynamic matrix of the predictor, $\boldsymbol{\theta}$, can be expressed

$$\hat{\mathbf{Y}} = \underbrace{\mathbf{X}\mathbf{Z}\mathbf{B}\mathbf{Q}^T}_{\boldsymbol{\theta}}. \tag{5.23}$$

In the receding horizon policy predictions in the near horizon are more important than those far away provided only the first control action in the sequence of control actions is eventually applied to the process. Also, in the parametric MRI approach that minimizes the multi-step ahead function in Section 5.2, predictions at $k+i$ depend on predictions at $k+j$, for $j < i$, then predictions in the near horizon are given more importance. This feature can be added to the dynamic matrix approach by giving more weight to predictions in the near horizon for identification. To do so a weighting matrix can be used to transform $\mathbf{Y}$

$$\check{\mathbf{Y}} = \mathbf{Y}\boldsymbol{\Lambda} \tag{5.24}$$

$\boldsymbol{\Lambda} \in \mathbb{R}^{n_y \times n_y}$ is assumed to be a definite positive matrix. $\boldsymbol{\Lambda}$ can be a diagonal matrix with decreasing values so that predictions in the near horizon are weighted more than predictions in the far horizon. If $\check{\mathbf{Y}}$ is used instead of $\mathbf{Y}$ in PLS, Equation (4.1) yields

$$\underbrace{\mathbf{Y}\boldsymbol{\Lambda}}_{\check{\mathbf{Y}}} = \mathbf{U}\,\underbrace{\mathbf{Q}^T\boldsymbol{\Lambda}}_{\check{\mathbf{Q}}^T} + \mathbf{F}\boldsymbol{\Lambda}$$

Then matrix $\mathbf{Q}$ of the model is obtained from $\check{\mathbf{Q}}$

$$\mathbf{Q} = \boldsymbol{\Lambda}^{-1}\check{\mathbf{Q}}$$

Finally, the PLS model is defined by: $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{B}$.

CHAPTER 6

MIMO VERSUS MULTIPLE MISO IN PARAMETRIC MRI

6.1. Introduction

When identifying a multiple input multiple output process, two approaches can be distinguished in terms of the structure of the model:

- **(I)**: One MIMO (Multiple Input Multiple Output) model
- **(II)**: As many MISO (Multiple Input Single Output) models as there are outputs in the process.

The purpose of this chapter is to compare both approaches, **(I)** and **(II)**, in traditional one-step ahead identification and in MRI (Model predictive control Relevant Identification) [Laurí 10c].

This chapter is organized as follows: Section 6.2, compares the MIMO and MISO approaches in one-step ahead identification. Section 6.3, compares the MIMO and MISO approaches in multi-step ahead identification (MRI). In Section 6.4, the identification of a 2×3 MIMO PEMFC (Proton Exchange Membrane Fuel Cell) is used to compare the performance of the MIMO and MISO identification approaches. Finally conclusions for this chapter are drawn in Section 6.5.

6.2. MIMO vs MISO in one-step ahead identification

Traditional PEMs (Prediction Error Methods) minimize the squared one-step ahead prediction error, denoted as J_{osapi} in this thesis, Equation (5.13)

$$J_{\text{osapi}}(k) = ||\mathbf{e}_{\text{k+1}}||^2_{\text{F}}$$

For an N-sample identification data set, the identification cost index can be expressed as the sum over N of the squared errors of one-step ahead predictions

$$J_{\text{OSAPI}} = \sum_{k=1}^{N} ||\mathbf{e}_{\text{k+1}}||^2_{\text{F}} = SSEP \tag{6.1}$$

The identification cost index J_{OSAPI} is often denoted as SSEP (Sum of Squared Error of Prediction).

The MSEP (Mean Squared Error of Prediction) evaluated for a validation data set can be used to evaluate the performance of the model in one-step ahead predictions. MSEP can be obtained from $SSEP$:

$$MSEP = \frac{1}{N} \sum_{k=1}^{N} ||\mathbf{e}_{\text{k+1}}||^2_{\text{F}} = \frac{SSEP}{N}.$$

MSEP can also be expressed as the sum of squared bias and residual variance errors [Höskuldsson 88]. The discrimination between bias and variance errors

is commonly used in system identification to quantify the error in the estimated model [Hjalmarsson 06]. The bias error is due to the model structure being restricted when compared to the system being estimated; and the variance error is due to noise and disturbances in the observed input-output data. Bias and variance errors are related to both the number of parameters in the model n_{p}, and the number of error terms used to fit the parameters of the model n_{e}:

- $\downarrow n_{\mathrm{p}} \rightarrow\uparrow$ Bias: Models with too few parameters are inaccurate because of a large bias error, they do not have enough flexibility.
- $\uparrow n_{\mathrm{p}} \rightarrow\uparrow$ Variance: Models with too many parameters are inaccurate because of a large variance error, they are overfitted to the identification data set. There has been some controversy in the literature about quantifying the variance error for finite model order [Ninness 04]. According to the most recent work [Mårtensson 09], the variance of minimum phase zeros and stable poles is very sensitive and grows exponentially with the model order. Moreover, the asymptotic variance of nonminimum phase zeros and unstable poles is only slightly affected by the model order.
- $\uparrow n_{\mathrm{e}} \rightarrow\downarrow$ Variance: The larger the number of prediction errors in the identification cost index n_{e}, the smaller the variance error [Hjalmarsson 06].

Figure 6.1 helps understand the above statements. The identification error is composed of Variance and Bias errors. Variance error can be reduced by decreasing n_{p} and/or increasing n_{e}. Bias error can be reduced increasing n_{p}. Then n_{p} needs be large to reduce Bias error, and n_{e} needs be large as well to reduce Variance error. However, n_{e} is limited by the number of samples in the identification data set N.

In one-step ahead identification, n_{e} equals the size of the identification data set N, see sum in Equation (6.1). Thus, for a given identification data set, n_{p} is the only design choice to act upon the bias variance trade-off. In the case of MIMO, the number of parameters can be reduced by taking a multiple

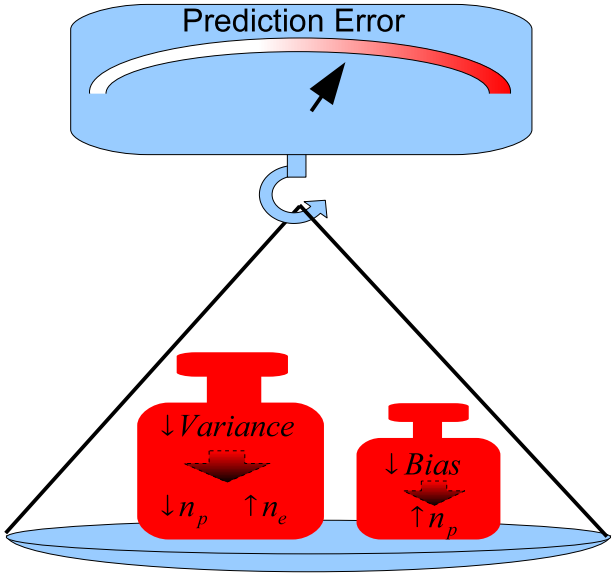


Figure 6.1: Effect of n_{p} and n_{e} on prediction error.

MISO approach. In a multiple MISO approach, unlike in the MIMO approach, instead of estimating one model with $n_{\mathrm{o}} \times n_{\mathrm{x}}$ parameters, n_o MISO models are obtained, and each MISO model has n_x parameters. Then n_o identifications of n_{x} parameters each are performed instead of one identification of $n_{\mathrm{o}} \times n_{\mathrm{x}}$ parameters, which reduces the number of parameters in each identification and consequently reduces the Variance error.

The MISO model for output s is defined as column s of $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{\mathrm{s}}$. Two cases of MISO identification are considered:

- Multiple MISO-I: Output interaction is considered. All the parameters in $\boldsymbol{\theta}_{\mathrm{s}}$ are fitted to the identification data set; thus, each MISO model has n_{x} parameters.

- Multiple MISO[1]: Output interaction is neglected. Only the parameters of $\boldsymbol{\theta}_{\mathrm{s}}$ associated with process output s and the inputs are fitted to the identification data set. Parameters associated to other outputs are set to zero, then $n_{\mathrm{a}}(n_{\mathrm{o}}-1)$ parameters are set to zero. Provided $n_{\mathrm{x}} = n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{o}}n_{\mathrm{a}}$, the number of parameters to fit in each model is $n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{a}}$.

Figure 6.2 helps compare the three identification approaches. The parameters of the linear model of a MIMO process are contained in $\boldsymbol{\theta}$. In the MIMO approach, $\boldsymbol{\theta}$ is fitted to the identification data set. In the multiple MISO-I approach, each column of $\boldsymbol{\theta}$ is obtained separately and then all the columns are bounded together forming the multiple input multiple output model $\boldsymbol{\theta}$. Note the number of parameters of the final model in these two approaches is the same. In the multiple MISO approach, each column of $\boldsymbol{\theta}$ is obtained separately, but output s cannot depend on any other output but s, then some zeros are forced, and thus the resulting structure has less parameters to fit than the previous two approaches.

Table 6.1 contains the number of error terms in the cost index n_{e}, and the number of parameters n_{p} for a comparison of the three identification approaches. The remainder of this section is divided into two subsections: firstly, the MIMO approach is compared with the multiple MISO-I approach; secondly, the MIMO approach is compared with the multiple MISO approach.

	n_{e}	n_{p}
MIMO	N	$\underbrace{(n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{o}}n_{\mathrm{a}})}_{n_{\mathrm{x}}}n_{\mathrm{o}}$
MISO-I	N	$\underbrace{(n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{o}}n_{\mathrm{a}})}_{n_{\mathrm{x}}}n_{\mathrm{o}}$
MISO	N	$(n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{a}})n_{\mathrm{o}}$

Table 6.1: one-step ahead identification

[1]Note the multiple MISO approach is considered in Appendix A in which matrices $\mathbb{A}_{\beta}$ are diagonal.

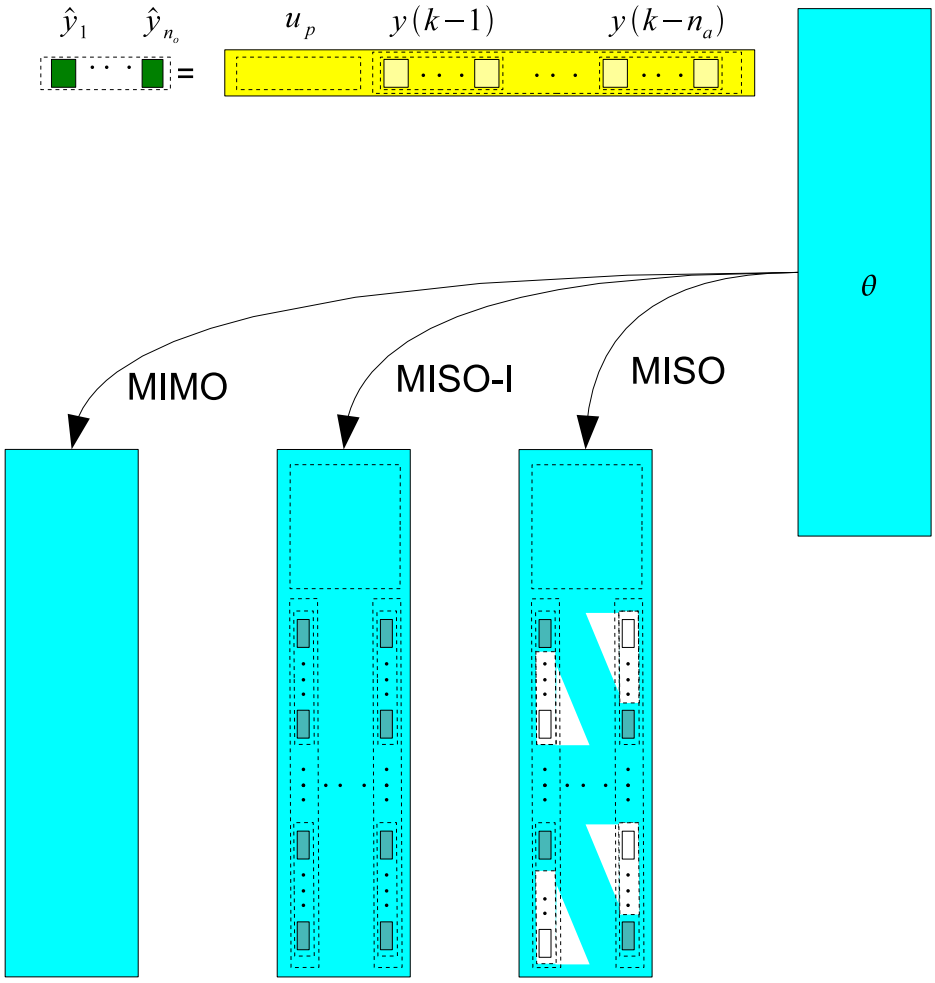


Figure 6.2: The three identification strategies for the linear MIMO model.

6.2.1. MIMO vs multiple MISO-I in one-step ahead identification

In the one-step ahead identification approach there is no difference between the MIMO identification approach and the multiple MISO-I identification approach. To prove this, J_{OSAPI} in Equation (6.1) can be expressed as:

$$J_{\text{OSAPI}} = \sum_{k=1}^{N} \sum_{s=1}^{n_o} e_{\text{s}}(k+1)^2$$

where $e_{\text{s}}(k+1)$ is column s of $\mathbf{e}_{\text{k+1}}$. According to Equations (5.6) and (5.4), $e_{\text{s}}(k+1)$ can be expressed:

$$e_{\text{s}}(k+1) = y_{\text{s}}(k+1) - \underbrace{\mathbf{x}_{\text{k}}\boldsymbol{\theta}_{\text{s}}}_{\hat{y}_{\text{s}}(k+1)}$$

thus, J_{OSAPI} can be expressed:

$$J_{\text{OSAPI}} = \sum_{k=1}^{N} \sum_{s=1}^{n_o} (y_{\text{s}}(k+1) - \mathbf{x}_{\text{k+1}}\boldsymbol{\theta}_{\text{s}})^2. \tag{6.2}$$

Element in row o and column i in $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{\text{o,i}}$, is obtained from the minimization of J_{OSAPI}:

$$\min_{\boldsymbol{\theta}_{\text{o,i}}} J_{\text{OSAPI}}$$

Such a minimum is attained when

$$\frac{\partial J_{\text{OSAPI}}}{\partial \boldsymbol{\theta}_{o,i}} = 0.$$

From (6.2), it can be seen that the only terms in $\frac{\partial J_{\text{OSAPI}}}{\partial \boldsymbol{\theta}_{\text{o,i}}}$ that depend on $\boldsymbol{\theta}_{\text{o,i}}$ are those for which $s = o$. Hence, only column o of $\boldsymbol{\theta}$ has an influence in obtaining $\boldsymbol{\theta}_{\text{o,i}}$. Thus, each column of $\boldsymbol{\theta}$ can be estimated separately.

It can be concluded from the above that MIMO identification is equivalent to multiple MISO-I identification in one-step ahead identification.

6.2.2. MIMO vs multiple MISO in one-step ahead identification

From Table 6.1, it can be seen that the multiple MISO model has

$$\underbrace{(n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{o}}n_{\mathrm{a}})n_{\mathrm{o}}}_{MIMO} - \underbrace{(n_{\mathrm{i}}n_{\mathrm{b}} + n_{\mathrm{a}})n_{\mathrm{o}}}_{MISO}$$

$$n_{\mathrm{o}}n_{\mathrm{a}}n_{\mathrm{o}} - n_{\mathrm{a}}n_{\mathrm{o}}$$

$$(n_{\mathrm{o}}n_{\mathrm{a}} - 1)n_{\mathrm{o}}$$

parameters fewer than the MIMO model. Thus, the multiple MISO approach presents a larger bias error, but a smaller variance error than the MIMO approach.

According to [Mårtensson 09], if the model structure is flexible enough to describe the true underlying dynamics, the variance error is the dominating part. Therefore, if the multiple MISO structure can describe the dynamics of the process, then the variance error is the dominating part in MSEP. If the variance error is the dominating part, the multiple MISO approach presents a smaller identification error than the MIMO approach due to a reduction in the variance error. Provided this is often the case, then traditional one-step ahead identification often resorts to multiple MISO identification.

An example in which the variance error is the dominating part is provided in Section 6.4. In the example, three models of different orders for the same process are obtained. The multiple MISO approach is shown to outperform the MIMO approach in one-step ahead identification for the three model orders.

6.3. MIMO vs MISO in multi-step ahead identification

It is argued in this section that, unlike in the one-step ahead identification approach, the MIMO identification approach is preferable to the multiple MISO approach in MRI if the prediction horizon n_f is sufficiently large. In this section: firstly, n_e for multi-step ahead identification is obtained; secondly, the MIMO approach is compared with the multiple MISO-I approach in multi-step ahead identification; and finally, the MIMO approach is compared with the multiple MISO in multi-step ahead identification.

The number of error terms in J_LRPI in Equation (5.14) can be expressed as

$$n_\mathrm{e} = \sum_{k=1}^{N} \sum_{j=1}^{n_\mathrm{f}} 1 = N n_\mathrm{f}.$$

The number of parameters to be estimated in each approach is the same in the multi-step ahead and one-step ahead approaches. Table 6.2 shows the number of error terms in the identification cost index n_e, and the number of parameters n_p in each approach.

	n_e	n_p
MIMO	Nn_f	$\underbrace{(n_\mathrm{i}n_\mathrm{b} + n_\mathrm{o}n_\mathrm{a})}_{n_\mathrm{x}}n_\mathrm{o}$
MISO-I	Nn_f	$\underbrace{(n_\mathrm{i}n_\mathrm{b} + n_\mathrm{o}n_\mathrm{a})}_{n_\mathrm{x}}n_\mathrm{o}$
MISO	Nn_f	$(n_\mathrm{i}n_\mathrm{b} + n_\mathrm{a})n_\mathrm{o}$

Table 6.2: Multi-step ahead identification

6.3.1. MIMO vs multiple MISO-I in multi-step ahead identification

The multiple MISO-I approach has as many degrees of freedom as the MIMO approach. In the multiple MISO-I approach, n_{o} MISO identifications are performed; thus, n_{o} minimization problems, with n_{x} parameters each, are solved. However, in the MIMO approach one minimization problem with $n_{\mathrm{o}} \times n_{\mathrm{x}}$ parameters is solved. The effect of approximating the minimization problem by n_{o} minimizations is studied by comparing J_{lrpi} in the MIMO case to J_{lrpi} in the multiple MISO-I case, $\underset{-}{J}_{\mathrm{lrpi}}$. The following (2×2) MIMO example is used to compare both cost indices:

$$[\hat{y}_1(k+1) \quad \hat{y}_2(k+1)] = [u_1(k) \quad u_2(k) \quad y_1(k) \quad y_2(k)] \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \quad (6.3)$$

The cost index in Equation (5.12) is particularized[2] for the above example with $n_{\mathrm{f}} = 2$:

$$J_{\mathrm{lrpi}}(k) = e_1(k+1)^2 + e_2(k+1)^2 + e_1(k+2)^2 + e_2(k+2)^2 \quad (6.4)$$

The minimization index in the multiple MISO-I approach is defined as:

$$\underset{-}{J}_{\mathrm{lrpi}}(k) = \underset{-}{e}_1(k+1)^2 + \underset{-}{e}_2(k+1)^2 + \underset{-}{e}_1(k+2)^2 + \underset{-}{e}_2(k+2)^2$$

The four error terms can be obtained as follows:

$$\underset{-}{e}_{\mathrm{i}}(k+l)^2 = [y_{\mathrm{i}}(k+l) - \underset{-}{\hat{y}}_{\mathrm{i}}(k+l)]^2; \quad \forall i \in [1,2], \quad l \in [1,2]$$

[2] $e_{\mathrm{s}}(k+j)$ is defined as the column s of $\boldsymbol{e}(k+j|k)$.

The MISO-I predictions are obtained:[3]

$$\begin{aligned}
\underset{-}{\hat{y}_{\mathrm{i}}}(k+1) &= \hat{y}_{\mathrm{i}}(k+1); \quad \forall i \in [1,2] \\
\underset{-}{\hat{y}_1}(k+2) &= b_{11}u_1(k+1) + b_{21}u_2(k+1) + a_{11}\underset{-}{\hat{y}_1}(k+1) + a_{21}y_2(k+1) \\
&= \hat{y}_1(k+2) + a_{21}e_2(k+1) \\
\underset{-}{\hat{y}_2}(k+2) &= b_{12}u_1(k+1) + b_{22}u_2(k+1) + a_{12}y_1(k+1) + a_{22}\underset{-}{\hat{y}_2}(k+1) \\
&= \hat{y}_2(k+2) + a_{12}e_1(k+1)
\end{aligned}$$

The four error terms can now be expressed:

$$\begin{aligned}
\underset{-}{e_i}(k+1)^2 &= e_i(k+1)^2; \quad \forall i \in [1,2] \\
\underset{-}{e_1}(k+2)^2 &= [e_1(k+2) - a_{21}e_2(k+1)]^2 \\
\underset{-}{e_2}(k+2)^2 &= [e_2(k+2) - a_{12}e_1(k+1)]^2
\end{aligned}$$

From Equation (6.4), the MISO-I index can be expressed as:

$$\begin{aligned}
\underset{-}{J_{\mathrm{lrpi}}}(k) &= J_{\mathrm{lrpi}}(k) + a_{21}^2 e_2(k+1)^2 + a_{12}^2 e_1(k+1)^2 \\
&\quad -2a_{21}e_1(k+2)e_2(k+1) - 2a_{12}e_2(k+2)e_1(k+1). \qquad (6.5)
\end{aligned}$$

The following conclusions can be drawn from the comparison of $\underset{-}{J_{\mathrm{lrpi}}}(k)$ and $J_{\mathrm{lrpi}}(k)$:

- The multiple MISO-I approach is equivalent to the MIMO MRI approach only if there is no interaction between process outputs. In the given example this implies $a_{12} = a_{21} = 0$.
- Should there be correlation among process outputs, the multiple MISO-I approach minimizes a weighted version of J_{lrpi} that weights more predictions in the near horizon, and a term that distorts J_{lrpi} is added. The

[3] When estimating output s, all outputs except s are assumed to be known if MISO-I identification is being performed.

shape of the cost function changes; and although this function may provide better results than the one-step ahead identification approach, it is not MRI.

For these reasons, the MIMO approach is preferable to the multiple MISO-I in MRI. In the example in Section 6.4, the MIMO approach is shown to outperform the multiple MISO-I approach in multi-step ahead identification.

6.3.2. MIMO vs multiple MISO in multi-step ahead identification

In the case of MIMO identification, the cost index has N error terms in one-step ahead identification (Table 6.1), and Nn_{f} error terms in multi-step ahead identification (Table 6.2). Provided increasing the number of error terms in the identification data set reduces the variance error [Hjalmarsson 06], the variance error in multi-step ahead identification is reduced as n_{f} increases.

As argued in Section 6.2.2, the multiple MISO approach that trades variance for bias is preferred in one-step ahead identification if the variance error is the dominating term. However, in multi-step ahead identification as the variance error is reduced by a ratio of n_{f}, for a sufficiently large n_{f}, the variance error loses importance and trading variance for bias worsens the overall identification error.

Consequently, for a sufficiently large n_{f}, the MIMO approach is preferable to the multiple MISO in MRI. In the example in Section 6.4, the MIMO approach is shown to outperform the multiple MISO approach in multi-step ahead identification.[4]

[4] Although only results for $n_{\mathrm{f}} = 10$ are shown in the example in Section 6.4, simulations have been conducted for $n_{\mathrm{f}} \in [1, .., 10]$, and better models are always obtained in the MIMO approach. Therefore, the minimum value of $n_{\mathrm{f}} = 1$ is sufficient in the given example for the MIMO approach to outperform the multiple MISO approach.

6.4. Identification Examples

The Simulink® MIMO (2×3) Proton Exchange Membrane Fuel Cell (PEMFC) described in [Pukrushpan 04] is used as the process to identify.

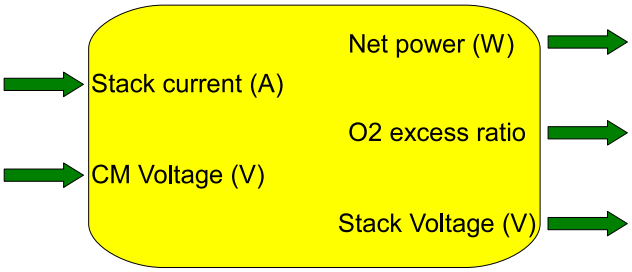


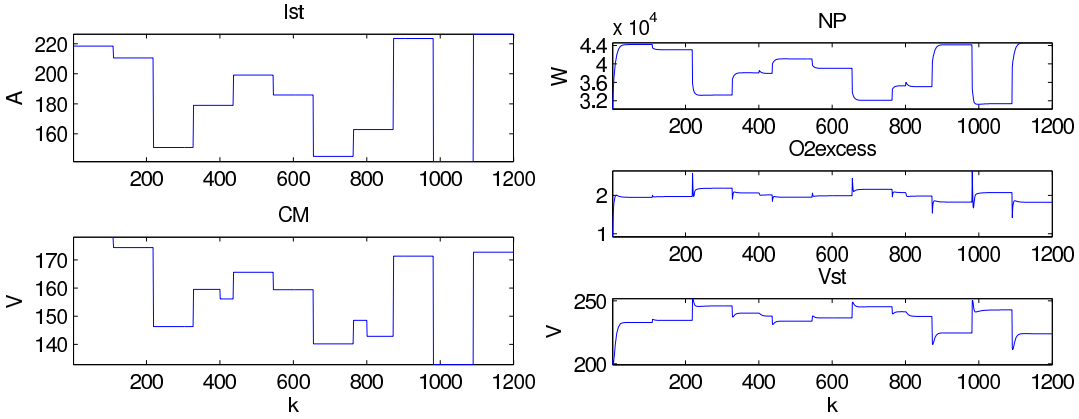
Figure 6.3: Block diagram of the PEMFC.

The working point around which the nonlinear process is to be approximated is:

- Inputs
  - stack current: Ist= 184 ± 40 A
  - compressor voltage: CM= 155 ± 23 V
- Outputs
  - net power: NP= 37.4 ± 7.2 kW
  - oxygen excess ratio: O2excess= 2.15 ± 0.45
  - stack voltage: Vst= 225 ± 26 V

Three linear models of orders $n_a = n_b = \{1, 2, 3\}$ are used to approximate the process. A prediction horizon of $n_f = 10$ is considered. The open-loop identification data set is shown in Figure 6.4. The sampling interval is $T_s = 0.05$s and the identification data set is mean-centered and scaled prior

to identification. Stack current is generated as a sequence of steps around the working point. CM is generated to keep the process in a safe operating area while avoiding linear dependence of inputs. CM is obtained 70% from the feed-forward controller on Ist designed in [Pukrushpan 04], and 30% using a step signal around the working point.



(a) Inputs

(b) Outputs

Figure 6.4: Identification data set

The various identification approaches to compare are:

- **LS:** The easiest MIMO approach as it minimizes the J_{OSAPI} index.
- **LS-MISO:** The multiple MISO LS approach that neglects output interaction. This is often preferred to the MIMO LS approach to reduce variance error.
- **LM-MIMO:** The MIMO MRI approach. The Levenberg Marquardt (LM) algorithm implemented in the Matlab® optimization toolbox is used to minimize J_{LRPI}.
- **LM-MISO:** The multiple MISO MRI approach that neglects output interaction. The Levenberg Marquardt algorithm is used to fit each of the MISO models.

- **LM-MISO-I:** The multiple MISO MRI approach that considers output interaction. The Levenberg Marquardt algorithm is used to fit each of the MISO models.

Two external validation indicators are used to measure the performance of each of the models obtained.

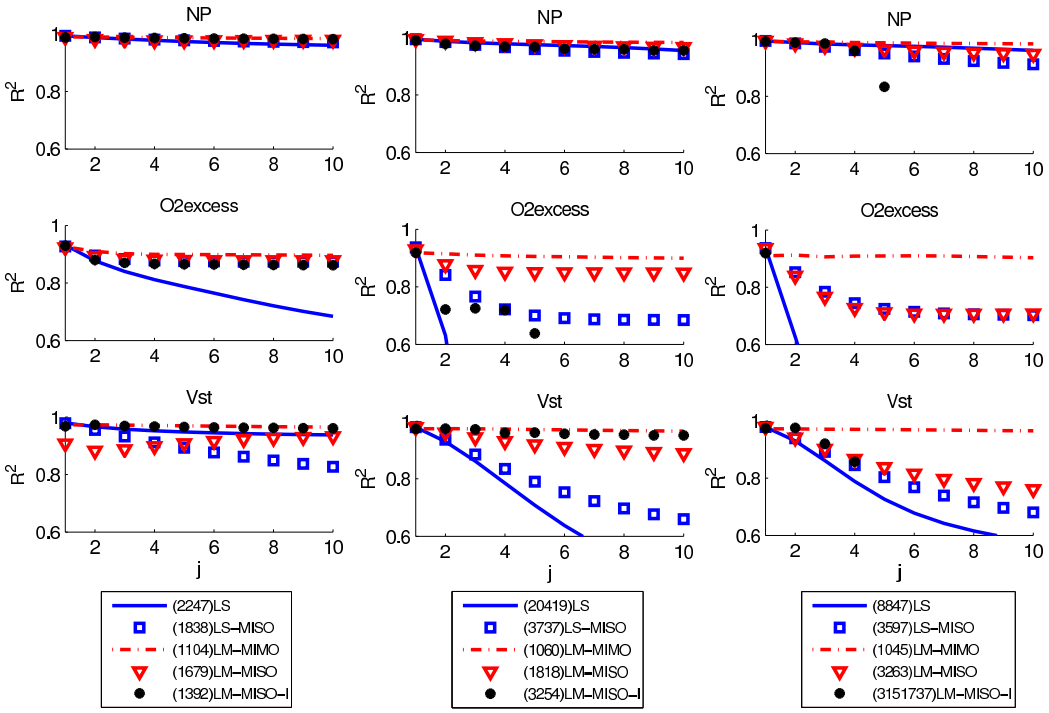
- $J_{\mathrm{LRPI_{EV}}}$ is J_{LRPI} evaluated for validation data. $J_{\mathrm{LRPI_{EV}}}$ is used to compare the performance of the models in terms of MRI.
- $\mathbf{R}^2$ is a $(n_{\mathrm{f}} \times n_{\mathrm{o}})$ matrix containing the values of the coefficient of determination. $\mathbf{R}^2$ shows the evolution of predictive performance for each output in the prediction horizon. The element located in column s and row j represents the coefficient of determination for output s when performing predictions at k+j:[5]

$$r^2_{\mathrm{s,j}} = 1 - \frac{\sum_{k=1}^{N}(e_{\mathrm{s}}(k+j))^2}{\sum_{k=1}^{N}(y_{\mathrm{s}}(k+j) - E\{y_{\mathrm{s}}\})^2} \quad \forall \mathrm{j} \in [1, n_{\mathrm{f}}],\ \forall \mathrm{s} \in [1, n_{\mathrm{o}}]$$

These indicators are evaluated for thirty experiments performed in a Monte Carlo simulation. The mean of the thirty indicators obtained are the external validation indicators shown in this example. These indicators show the performance of each model, i.e., the solution obtained in the bias and variance trade-off for each order of the model and identification technique. The validation indicators — $J_{\mathrm{LRPI_{EV}}}$ and $\mathbf{R}^2_{(10\times3)}$— are obtained for each of the model orders and identification methods. The results for $n_{\mathrm{a}} = n_{\mathrm{b}} = 1$ are shown in Figure 6.5(a). The results for $n_{\mathrm{a}} = n_{\mathrm{b}} = 2$ are shown in Figure 6.5(b). The results for $n_{\mathrm{a}} = n_{\mathrm{b}} = 3$ are shown in Figure 6.5(c).

The value of $J_{\mathrm{LRPI_{EV}}}$ is shown in brackets in the legend for each identification method. $\mathbf{R}^2_{(10\times3)}$ is represented in three plots, so $\mathbf{R}^2_{\mathrm{s}}$ is a ten element

[5] $E\{.\}$ is the mean operator.



(a) $n_a = n_b = 1$ (b) $n_a = n_b = 2$ (c) $n_a = n_b = 3$

Figure 6.5: Mean of the validation indicators obtained for thirty external validation experiments

column vector associated with output s. The upper plot in Figure 6.5(a) is for $s = 1$ (NP), the middle plot is for $s = 2$ (O2excess), and the lower plot is for $s = 3$ (Vst). The closer to one the $\mathbf{R}^2_{\mathrm{s}}$ lines are, the better the predictive performance of the model. According to $\mathbf{R}^2$ and $J_{\mathrm{LRPI_{EV}}}$, for a model of order $n_{\mathrm{a}} = n_{\mathrm{b}} = 1$ (Figure 6.5(a)), the model fitted with LM-MIMO presents the best validation results with $J_{\mathrm{LRPI_{EV}}} = 1104$, and slightly higher values of $\mathbf{R}^2$. It can be seen that the coefficient of determination for j=1 is almost equal for all methods; thus, all the models obtained are almost equivalent when performing predictions at $k + 1$. Nevertheless, as j increases, predictive performance of the models obtained with all methods but LM-MIMO decreases.

The following observations can be drawn by comparing LS and LS-MISO for the three model orders:

- Models fitted with LS-MISO outperform those fitted with LS in terms of $J_{\mathrm{LRPI_{EV}}}$ for all model orders.
- LS and LS-MISO provide their best model for $n_{\mathrm{a}} = n_{\mathrm{b}} = 1$.
- Models fitted with LS-MISO outperform those fitted with LS in terms of $\mathbf{R}^2$ for output $s = 2$; and also for output $s = 3$ and orders $n_{\mathrm{a}} = n_{\mathrm{b}} = \{2, 3\}$.

Therefore, in the one-step ahead approach with the identification data set in Figure 6.4, the models with fewer parameters present the better predictive performance. Provided LS-MISO fits fewer parameters than LS, the model $n_{\mathrm{a}} = n_{\mathrm{b}} = 1$ fitted with LS-MISO outperforms that fitted with LS. It can be seen in Figure 6.5(a) that the model fitted with LS outperforms that fitted with LS-MISO for output $s = 3$. Therefore, although in the one-step ahead approach multiple MISO identification is preferable, it is not better for all outputs.

The following observations can be drawn by comparing the three LM approaches for the three model orders:

- LM-MIMO outperforms both MISO MRI approaches for all outputs and model orders.
- LM-MISO-I only outperforms LM-MISO for $n_{\mathrm{a}} = n_{\mathrm{b}} = 1$.

LM-MISO and LM-MISO-I attempt to trade bias for variance to provide better models. However, LM-MIMO reaches a trade-off solution between bias and variance without reducing the degrees of freedom, and so obtaining more predictive models. The MIMO MRI approach is preferable for all model orders and outputs.

The following observations can be drawn by comparing the five identification approaches for the three model orders:

- Models fitted with LM-MIMO present the flattest and closest to one $\mathbf{R}^2$ line plots for all outputs and model orders.
- LM-MIMO is the only approach that slightly changes $J_{\mathrm{LRPI_{EV}}}$ with the order of the model. $J_{\mathrm{LRPI_{EV}}}$ ranges from 1104 in the worst model to 1045 in the best model. This range can be expressed as $(1104-1045)/1104 \rightarrow 5.3\%$. The range for the other methods is: LS 89.0%, LS-MISO 48.9%, LM-MISO 48.5%, LM-MISO-I 99.9%.
- The best predictive model is obtained using LM-MIMO and an order of $n_{\mathrm{a}} = n_{\mathrm{b}} = 3$ with $J_{\mathrm{LRPI_{EV}}} = 1045$.
- The worst model fitted with LM-MIMO is better than the best model fitted with any other identification approach.

Therefore, LM-MIMO provides the best predictive models, not only performing well at $k+1$, but within the prediction window. Unlike the one-step ahead approach, MIMO identification is preferable in MRI as shown by the indicators in Figure 6.5.

6.5. Conclusions

The MIMO and MISO identification approaches to model a MIMO plant have been compared both for traditional one-step ahead identification, and MRI. It is argued in this chapter that, unlike in one-step ahead identification, the MIMO approach is preferable to the multiple MISO approach in MRI for a sufficiently large prediction horizon. As the MIMO identification approach is preferable in MRI; the minimization of the multi-step ahead cost function, or the use of subspace identification is preferable to multiple MISO prefilter based MRI approaches.

A PEMFC example is provided to support the arguments. As shown in the example, the multiple MISO approach outperforms the MIMO approach in the case of one-step ahead identification for all the evaluated model orders. In MRI however, the MIMO approach outperforms the multiple MISO approach for all the evaluated model orders. When comparing all the models, the best model for MPC is obtained for MIMO MRI for an order of three.

CHAPTER 7

PLS-PH: MRI IN THE CASE OF CORRELATION

7.1. Introduction

In industrial multivariate processes with a large number of controllable and uncontrollable variables, the identification data set is often ill-conditioned due to collinearity. To cope with collinearity often LVMs (Latent Variable Methods) such as PLS (Partial Least Squares) are used in the identification stage [Song 02, Kiers 07].

This chapter proposes a solution for parametric MRI (Model predictive control Relevant Identification) resorting to LVMs to cope with collinearity. The PLS-PH (Partial Least Squares Prediction Horizon) approach is proposed in this chapter as a solution for minimizing the multi-step ahead prediction error cost function to obtain a parametric MRI model, even in the case of an ill-conditioned identification data set.

7.2. Proposed Algorithm: PLS-PH

PLS-PH is an iterative approach to minimize J_{LRPI} that lumps together numerical optimization and PLS [Laurí 10a]. Prior to introducing PLS-PH, some aspects of numerical optimization are reviewed. In the sake of clarity, the PLS-PH approach is explained in two steps. First, the minimization of J_{LRPI} is reformulated to use the LS formula; this first algorithm is denoted LS-PH. Second, LS in LS-PH is replaced with PLS; this second algorithm is denoted PLS-PH.

7.2.1. Numerical Optimization

Numerical optimization is intended to minimize cost functions that are nonlinear in their parameters:

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \tag{7.1}$$

The search is based on first or second order Taylor approximations of $f(\boldsymbol{\theta})$ in the region of $\boldsymbol{\theta}_{\mathrm{k}}$ ($\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}$), and moves towards better parameter sets in terms of $f(\boldsymbol{\theta}_{\mathrm{k+1}})$. The second order Taylor approximation of $f(\boldsymbol{\theta})$, and the iterations for $\boldsymbol{\theta}_{\mathrm{k+1}}$ can be expressed:

$$\tilde{f}_{\mathrm{k}}(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}) \approx f(\boldsymbol{\theta}_{\mathrm{k}}) + \mathbf{p}\frac{\partial f(\boldsymbol{\theta}_{\mathrm{k}})}{\partial \boldsymbol{\theta}} + \frac{1}{2}\mathbf{p}^{\mathrm{T}}\frac{\partial^2 f(\boldsymbol{\theta}_{\mathrm{k}})}{\partial \boldsymbol{\theta}^2}\mathbf{p}$$
$$\boldsymbol{\theta}_{\mathrm{k+1}} = \boldsymbol{\theta}_{\mathrm{k}} + \alpha_{\mathrm{k}}\,\mathbf{p}_{\mathrm{k}} \tag{7.2}$$

Two different approaches for the selection of the step length α_{k} and the search direction $\mathbf{p}_{\mathrm{k}}$ are considered [Nocedal 99].

- Line search methods initially define $\mathbf{p}_{\mathrm{k}}$, then find a suitable α_{k} that minimizes $f(\boldsymbol{\theta}_{\mathrm{k+1}})$. Line search methods can be separated into three

groups according to the definition of the search direction $\mathbf{p}_k$:

$$\mathbf{p}_k = -\mathbf{H}_k^{-1} \frac{\partial f(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}} \quad ; \quad \mathbf{H}_k = \begin{cases} \frac{\partial^2 f(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}^2} & \text{Newton} \\ \mathbf{B}_k \approx \frac{\partial^2 f(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}^2} & \text{Quasi-Newton} \\ \mathbf{I} & \text{steepest descendent} \end{cases}$$

The Newton approach presents a fast rate of local convergence —typically quadratic— when a neighbourhood of the solution is reached, and is insensitive to poor scaling in the data. The main drawback in the Newton approach is the need for the Hessian; thus, often Quasi-Newton solutions are used.

- Trust-region methods define a region Δ around $\boldsymbol{\theta}_k$ within which they trust $\tilde{f}_k(\boldsymbol{\theta}_k + \mathbf{p})$ to be an adequate representation of $f(\boldsymbol{\theta})$. And so α_k and $\mathbf{p}_k$ are chosen such that:

$$\alpha_k \, \mathbf{p}_k = \min_{\mathbf{p}} \tilde{f}_k(\boldsymbol{\theta}_k + \mathbf{p}) \quad s.t. \quad ||\mathbf{p}|| \leqq \Delta \tag{7.3}$$

If the move does not improve in terms of $f(\boldsymbol{\theta}_{k+1})$, $\boldsymbol{\theta}_{k+1}$ is not accepted and the trust region is reduced. Due to the reduction of the trust region, $\mathbf{p}_k$ obtained in the new iteration may point in a different direction. Levenberg Marquardt is considered a trust region method [Nocedal 99].

7.2.2. LS-PH

As stated in section 5.2.3, J_{LRPI} is a quadratic expression of a non-linear function in $\boldsymbol{\theta}$; thus, non-convex in general, and numerical optimization is to be used to obtain $\boldsymbol{\theta}$ in the minimization in Equation (7.1). LS-PH (Least Squares Prediction Horizon) is presented as a line search numerical optimization approach which is based on an approximation of J_{LRPI} different to Taylor's. The approximation used in this chapter does not need derivatives of J_{LRPI}, which —provided the problem has many dimensions— are laborious to obtain explicitly, or computationally demanding to approximate.

Prior to defining the approximation of J_{LRPI} proposed in this chapter, let us express Equation (5.14) in matrix notation:

$$J_{\mathrm{LRPI}}(\boldsymbol{\theta}) = \sum_{j=1}^{n_{\mathrm{f}}} \sum_{k=1}^{N} ||\mathbf{e}_{\mathrm{k+j}}||_{\mathrm{F}}^2 = ||\mathbf{E}_{\mathrm{a}}||_{\mathrm{F}}^2 \tag{7.4}$$

Where:

$$\mathbf{E}_{\mathrm{a}} = \begin{bmatrix} \mathbf{E}_{\mathrm{a_1}} \\ \vdots \\ \mathbf{E}_{\mathrm{n_f}} \end{bmatrix} \quad ; \quad \mathbf{E}_{\mathrm{a_j}} = \begin{bmatrix} \mathbf{e}_{\mathrm{1+j}} \\ \vdots \\ \mathbf{e}_{\mathrm{N+j}} \end{bmatrix}, \forall j \in [1, 2, \ldots, n_{\mathrm{f}}]$$

$\mathbf{E}_{\mathrm{a_j}}$ are matrices of dimensions $N \times n_{\mathrm{o}}$, and contain the identification errors based on predictions at $k+j$ with output information up to k. The one-step ahead prediction error matrix is obtained for $n_{\mathrm{f}} = 1$, $\mathbf{E}_{\mathrm{a}} = [\mathbf{E}_{\mathrm{a_1}}] = \mathbf{E}$. Each submatrix $\mathbf{E}_{\mathrm{a_j}}$ is obtained:

$$\mathbf{E}_{\mathrm{a_j}} = \mathbf{Y}_{\mathrm{a_j}} - \mathbf{X}_{\mathrm{a_j}} \boldsymbol{\theta} \tag{7.5}$$

Where:

$$\mathbf{Y}_{\mathrm{a_j}} = \begin{bmatrix} \mathbf{y}_{\mathrm{1+j}} \\ \vdots \\ \mathbf{y}_{\mathrm{N+j}} \end{bmatrix} \quad ; \quad \mathbf{X}_{\mathrm{a_j}} = \begin{bmatrix} \mathbf{x}_{\mathrm{j}} \\ \vdots \\ \mathbf{x}_{\mathrm{(N-1)+j}} \end{bmatrix} \tag{7.6}$$

The global identification error matrix $\mathbf{E}_{\mathrm{a}} \in \mathbb{R}^{N \cdot n_{\mathrm{f}} \times n_{\mathrm{o}}}$ is obtained:

$$\mathbf{E}_{\mathrm{a}} = \mathbf{Y}_{\mathrm{a}} - \mathbf{X}_{\mathrm{a}} \boldsymbol{\theta} \tag{7.7}$$

Being:

$$\mathbf{Y}_{\mathrm{a}} = \begin{bmatrix} \mathbf{Y}_{\mathrm{a_1}} \\ \vdots \\ \mathbf{Y}_{\mathrm{a_{n_f}}} \end{bmatrix} \quad ; \quad \mathbf{X}_{\mathrm{a}} = \begin{bmatrix} \mathbf{X}_{\mathrm{a_1}} \\ \vdots \\ \mathbf{X}_{\mathrm{a_{n_f}}} \end{bmatrix} \tag{7.8}$$

The cost index can now be expressed:

$$J_{\mathrm{LRPI}}(\boldsymbol{\theta}) = ||\mathbf{Y}_{\mathrm{a}} - \mathbf{X}_{\mathrm{a}}\boldsymbol{\theta}||_{\mathrm{F}}^2$$

Provided submatrices $\mathbf{X}_{\mathrm{a_j}}$, in $\mathbf{X}_{\mathrm{a}}$ $\forall j > 1$, depend on $\boldsymbol{\theta}$, the problem is non-linear in $\boldsymbol{\theta}$. J_{LRPI} can be approximated to an expression linear in $\boldsymbol{\theta}$ in the neighbourhood of $\boldsymbol{\theta}_{\mathrm{k}}$ by defining $\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}$ as $\mathbf{X}_{\mathrm{a}}$ computed using $\boldsymbol{\theta}_{\mathrm{k}}$:

$$\begin{aligned}
\tilde{J}_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}) &= ||\mathbf{Y}_{\mathrm{a}} - \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})||_{\mathrm{F}}^2 \\
&= tr((\mathbf{Y}_{\mathrm{a}} - \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}))^{\mathrm{T}} (\mathbf{Y}_{\mathrm{a}} - \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}))) \\
&= tr(\mathbf{Y}_{\mathrm{a}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}}) - tr(\mathbf{Y}_{\mathrm{a}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})) \dots \\
&- tr((\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}})) + tr((\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}))
\end{aligned} \tag{7.9}$$

As in the Newton approach, the search direction is obtained by setting the derivative of the approximation of the cost function $\tilde{J}_{\mathrm{LRPI}}$ to zero:

$$\begin{aligned}
\frac{\partial \tilde{J}_{\mathrm{LRPI}}}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})} &= \frac{\partial tr(\mathbf{Y}_{\mathrm{a}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}})}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})} - \frac{\partial tr(\mathbf{Y}_{\mathrm{a}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}))}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})} - \frac{\partial tr((\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}})}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})} \dots \\
&+ \frac{\partial tr((\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}))}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})} \\
&= 0 - (\mathbf{Y}_{\mathrm{a}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}})^{\mathrm{T}} - \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}} + (\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} + (\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}})^{\mathrm{T}})(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}) \\
&= -2\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}} + 2\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p})
\end{aligned} \tag{7.10}$$

$$\begin{aligned}
\frac{\partial \tilde{J}_{\mathrm{LRPI}}}{\partial(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}_{\mathrm{k}})} = 0 \quad &= \quad -2\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}} + 2\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}} (\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}_{\mathrm{k}}) \\
(\boldsymbol{\theta}_{\mathrm{k}} + \mathbf{p}_{\mathrm{k}}) \quad &= \quad (\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}})^{-1} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}} \\
\mathbf{p}_{\mathrm{k}} \quad &= \quad (\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}})^{-1} \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}} \mathbf{Y}_{\mathrm{a}} - \boldsymbol{\theta}_{\mathrm{k}}
\end{aligned} \tag{7.11}$$

It can be seen from the previous equation that $\mathbf{p}_\mathrm{k}$ is equal to the closed form LS solution of a linear problem with $\mathbf{X}_\mathrm{a}|_\mathrm{k}$ and $\mathbf{Y}_\mathrm{a}$, minus the current point.

So far $\mathbf{p}_\mathrm{k}$ to be used in Equation (7.2) has been obtained. The next step is to compute the step length α_k:

$$\alpha_\mathrm{k} = \min_{\alpha} J_\mathrm{LRPI}(\boldsymbol{\theta}_\mathrm{k} + \alpha\, \mathbf{p}_\mathrm{k})$$

An exact line search of α_k is expensive, and merely requiring a decrease in J_LRPI does not ensure global convergence; hence, the interest of inexact line search to determine α_k. The terminating conditions for the inexact line search is as a trade-off between attaining a substantial reduction in J_LRPI and not spending too much time in making the choice. When the first derivative of the cost function is available, there are sets of generally accepted terminating conditions, such as Goldstein, or Wolfe conditions. However, there is no general agreement to stop the search when the derivative is unavailable [Fletcher 87].

The inexact line search adopted in this chapter is based on quadratic approximations of J_LRPI:

$$J_\mathrm{LRPI}(\boldsymbol{\theta}_\mathrm{k} + \alpha\, \mathbf{p}_\mathrm{k}) \approx a + b\alpha + c\alpha^2 \tag{7.12}$$

Note the range of valid values for α is $\alpha \in [0, 1]$ as proven in Proposition 7.2.1.

Inexact line search:

1. J_LRPI is evaluated for three values of $\alpha = [0, 0.5, 1]$.
2. a, b and c in Equation (7.12) are obtained. The three values of J_LRPI and α obtained in step 1 are used to form a system of three equations. The system of three equations in three unknowns is solved.
3. The value of α that minimizes Equation (7.12) is obtained. If the new α is not in the valid range $\alpha \in [0, 1]$, it is discarded. In that case α is taken

as the middle point between two previous values of α if the parabola moves up; or between a previous value of α and 0 or 1 otherwise. The reader is referred to the algorithm provided in Appendix B for further information on special cases in the minimization of Equation (7.12).

4. J_{LRPI} is evaluated for the value of α obtained in step 3.

5. The smallest two values of J_{LRPI} obtained in the previous iteration plus the one obtained in step 4 are used to estimate a, b and c and start again the search of α going back to step 3.

The above procedure is repeated while the new values of α obtained provide considerably different values of J_{LRPI}, which is verified with the following termination condition:

$$|J_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k}} + \alpha_{\mathrm{k-1}}\, \mathbf{p}_{\mathrm{k}}) - J_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k}} + \alpha_{\mathrm{k}}\, \mathbf{p}_{\mathrm{k}})| > 0.001 J_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k}}) \tag{7.13}$$

Using $\mathbf{p}_{\mathrm{k}}$ obtained in Equation (7.11), and α_{k} obtained in the above line search procedure, the new value $\boldsymbol{\theta}_{\mathrm{k+1}}$ is obtained in Equation (7.2). LS-PH iterates while the following termination condition holds:

$$(J_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k}}) - J_{\mathrm{LRPI}}(\boldsymbol{\theta}_{\mathrm{k+1}})) > 0.001 \tag{7.14}$$

Fig. 7.1 shows a two variable example. The J_{LRPI} level curves are in continuous line, and those of $\tilde{J}_{\mathrm{LRPI}}$ are in dashed dotted line. $\boldsymbol{\theta}_{\mathrm{k}}$ is represented by a circle and the actual minimum of J_{LRPI} by a cross. The line starting at $\boldsymbol{\theta}_{\mathrm{k}}$ represents $\mathbf{p}_{\mathrm{k}}$, and the thicker part of $\mathbf{p}_{\mathrm{k}}$ equals $\alpha_{\mathrm{k}}\, \mathbf{p}_{\mathrm{k}}$. The square represents $\boldsymbol{\theta}_{\mathrm{k+1}}$.

LS-PH approximates J_{LRPI} in the neighbourhood of $\boldsymbol{\theta}_{\mathrm{k}}$ and uses the approximation to define the search direction and move to the new point $\boldsymbol{\theta}_{\mathrm{k+1}}$. Provided the initial guess is close enough to a minimizer of the cost function, the algorithm converges to the minimizer. LS-PH is a line search numerical

optimization algorithm which presents the following benefits compared to the Newton and Quasi-Newton approaches:

- There is no need to explicitly obtain or approximate the derivatives of J_{LRPI}. This is specially important for MIMO problems in which $\boldsymbol{\theta}$ has many dimensions.
- The search direction —$\mathbf{p}_{\mathrm{k}}$— is obtained by solving a linear regression problem. So far the LS closed form solution has been used, but any other approach to linear regression problems may be used. In the following subsection, LS is replaced with PLS bringing the potential of LVMs to the minimization of J_{LRPI}.

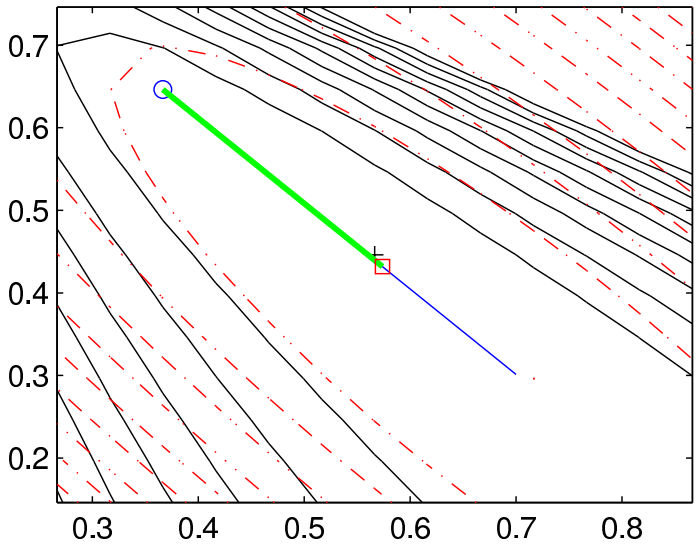


Figure 7.1: Level curves for J_{LRPI} and its approximation

Proposition 7.2.1 *The range of interest for α is $\alpha \in [0, 1]$*

Proof

- *α is to be greater than zero provided $\mathbf{p}_\mathrm{k}$ is a descendent direction*
- *Substituting $\mathbf{p}_\mathrm{k}$ form Equation (7.11) in Equation (7.2) and assuming $\alpha = 1$:*

$$\begin{aligned} \boldsymbol{\theta}_{\mathrm{k}+1} &= \boldsymbol{\theta}_\mathrm{k} + \alpha\, \mathbf{p}_\mathrm{k} \\ &= \boldsymbol{\theta}_\mathrm{k} + \alpha[(\mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{X}_\mathrm{a}|_\mathrm{k})^{-1}\, \mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{Y}_\mathrm{a} - \boldsymbol{\theta}_\mathrm{k}] \\ &= \boldsymbol{\theta}_\mathrm{k} + (\mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{X}_\mathrm{a}|_\mathrm{k})^{-1}\, \mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{Y}_\mathrm{a} - \boldsymbol{\theta}_\mathrm{k} \\ &= (\mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{X}_\mathrm{a}|_\mathrm{k})^{-1}\, \mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{Y}_\mathrm{a} \end{aligned}$$

then for $\alpha = 1$, $\boldsymbol{\theta}_{\mathrm{k}+1}$ obtained is the minimum of the approximation $\tilde{J}_\mathrm{LRPI}$. Then in the best case scenario, which is assuming $\tilde{J}_\mathrm{LRPI} = J_\mathrm{LRPI}$, the minimum is attained for $\alpha = 1$. □

7.2.3. PLS-PH

In the case of an ill-conditioned identification data set, line search methods yet introduced experience problems when inverting the matrix $\mathbf{H}_\mathrm{k}$; or in the case of LS-PH, when inverting $(\mathbf{X}_\mathrm{a}|_\mathrm{k}^\mathrm{T}\, \mathbf{X}_\mathrm{a}|_\mathrm{k})$ in Equation (7.11).

The solution proposed in this subsection is to replace the LS solution in Equation (7.11) with the PLS solution in Equations (4.6) and (4.4):

$$\begin{aligned} \mathbf{p}_{\mathrm{k}} &= (\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}}\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}})^{-1}\, \mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}^{\mathrm{T}}\, \mathbf{Y}_{\mathrm{a}} - \boldsymbol{\theta}_{\mathrm{k}} \\ &= \mathbf{Z}\mathbf{B}\mathbf{Q}^{T} - \boldsymbol{\theta}_{\mathrm{k}} \\ &= \underbrace{\mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}}_{\mathbf{Z}}\mathbf{B}\mathbf{Q}^{\mathrm{T}} - \boldsymbol{\theta}_{\mathrm{k}} \end{aligned} \tag{7.15}$$

Where $\mathbf{W}$, $\mathbf{P}$, $\mathbf{B}$ and $\mathbf{Q}$ are obtained applying PLS to $\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}$ and $\mathbf{Y}_{\mathrm{a}}$. The pseudo-code for the algorithm is introduced in Appendix B. The major advantage of this algorithm is the use of PLS to minimize J_{LRPI}, so parametric MRI is also possible in the case of ill-conditioned data.

7.3. Identification Examples

The predictive performance of the models obtained using three identification methods is compared. The three identification methods to compare are:

- **LS** is the easiest approach as it minimizes the J_{OSAPI} index. LS can justify the use of more complicated algorithms that minimize J_{LRPI} provided the model is to be used in an MPC framework.
- **LM** (Levenberg Marquardt) is regarded a trust region numerical optimization method. In this section the LM algorithm implemented in the Matlab® optimization toolbox is used to minimize J_{LRPI}.
- **PLS-PH** is the line search numerical optimization method presented in this chapter. PLS-PH minimizes J_{LRPI} resorting to PLS; thus, MRI can be done even in the case of ill-conditioned data.

Two identification examples are considered. First, a well-conditioned process is identified to compare LM and PLS-PH. In this example no reduction

in the number of variables is performed, thus, PLS-PH equals LS-PH. In the second example an ill-conditioned process is identified demonstrating the advantages of PLS-PH.

Two validation indicators are used to assess predictive performance of each of the models obtained. The validation indicators shown in this chapter are the mean of thirty Monte Carlo experiments.

- $J_{\mathrm{LRPI_{EV}}}$ is the value of the J_{LRPI} for external validation data. $J_{\mathrm{LRPI_{EV}}}$ is used to compare the performance of the models in terms of MRI.
- $\mathbf{R}^2$ is a $(n_{\mathrm{f}} \times n_{\mathrm{o}})$ matrix containing the values of the coefficient of determination for external validation data. Each element in $\mathbf{R}^2$ is obtained by using Equation (7.16). $\mathbf{R}^2$ is calculated to visualize the evolution of the performance within the prediction horizon. $\mathbf{R}^2$ has as many columns as there are process outputs, so $\mathbf{R}^2_{\mathrm{s}}$ is the column vector associated to output s.

$$\mathbf{R}^2_{\mathrm{s}}(j) = 1 - \frac{\mathbf{E}^{\mathrm{T}}_{\mathrm{a_{js}}} \mathbf{E}_{\mathrm{a_{js}}}}{(\mathbf{Y}_{\mathrm{a_{js}}} - E\{\mathbf{Y}_{\mathrm{a_{js}}}\})^{\mathrm{T}}(\mathbf{Y}_{\mathrm{a_{js}}} - E\{\mathbf{Y}_{\mathrm{a_{js}}}\})} \quad \forall j \in [1, n_{\mathrm{f}}], \forall s \in [1, n_{\mathrm{o}}] \tag{7.16}$$

Being: $\mathbf{E}_{\mathrm{a_{js}}}$, column s of $\mathbf{E}_{\mathrm{a_j}}$ in Equation (7.5); $\mathbf{Y}_{\mathrm{a_{js}}}$, column s of $\mathbf{Y}_{\mathrm{a_j}}$ in Equation (7.6); and $E\{.\}$ the mean operator.

7.3.1. Well-Conditioned Example

The process to be approximated is the non-linear Simulink® model of a PEMFC described in [Wang 05]. P_{anode}, P_{cathode}, and I_{load} are the process inputs. V_{stack} is the process output.

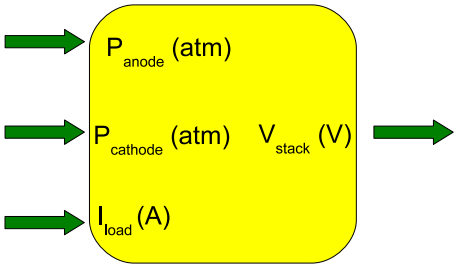


Figure 7.2: Block diagram of the PEMFC.

The non-linear process is excited in a region of the working point:

$$P_{\text{anode}} = 1.5 \pm 0.25 \text{ atm} \quad ; \quad P_{\text{cathode}} = 1 \pm 0.25 \text{ atm} \quad ; \quad I_{\text{load}} = 15 \pm 4 \text{ A}$$

The generated identification data set is shown in Figure 7.3. The sample time used is $T_{\text{s}} = 0.05$s. The identification data set is mean-centered and scaled prior to being used in the identification algorithms.

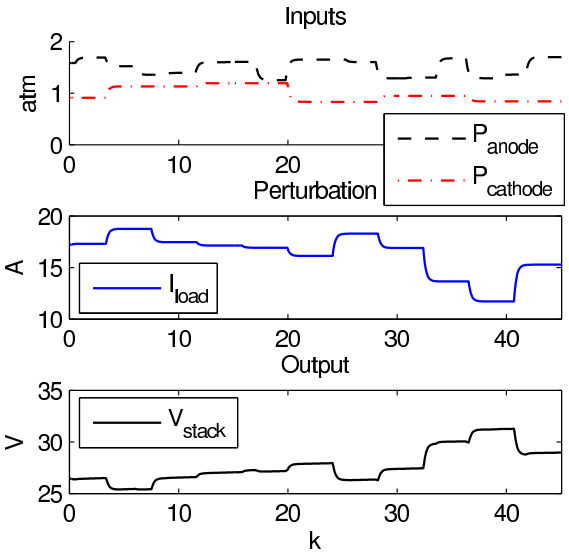


Figure 7.3: Identification data set for the PEMFC

The non-linear process is approximated by three linear models $n_a = n_b = \{1, 2, 3\}$. The prediction horizon is defined $n_f = 30$. The validation indicators, $J_{\mathrm{LRPI_{EV}}}$ and $\mathbf{R}^2_{(30\times1)}$, are shown in Fig. 7.4. Left plot is for the model $n_a = n_b = 1$, the middle plot is for $n_a = n_b = 2$, and the right plot is for $n_a = n_b = 3$. The value of $J_{\mathrm{LRPI_{EV}}}$ for each method is shown in the legend of each plot in Figure 7.4. $\mathbf{R}^2$ for the model obtained with each method is drawn as a line. The closer to 1 the $\mathbf{R}^2$ line is, and the smaller the value of $J_{\mathrm{LRPI_{EV}}}$, the better the model, and thus the algorithm used to obtain that model.

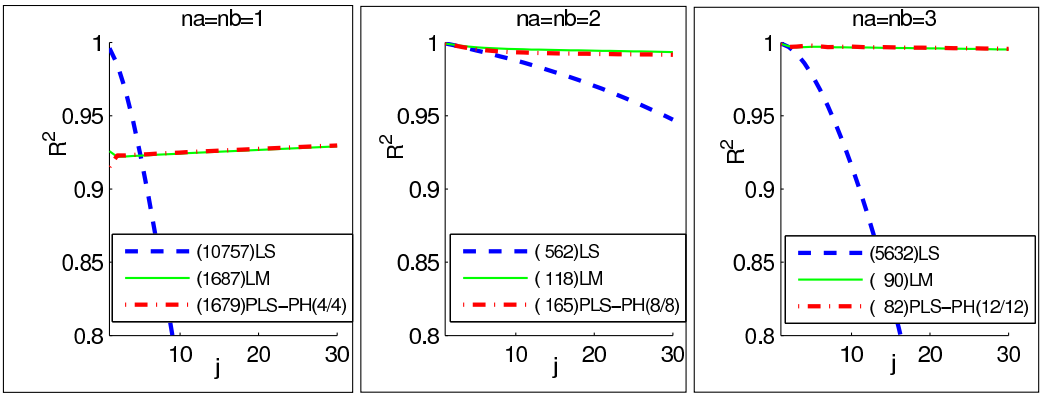


Figure 7.4: Validation results for the PEMFC example

As can be seen in Figure 7.4, the models obtained with LM and PLS-PH[1] are equivalent. LS provides worse models in terms of MRI as there is model-plant mismatch. It can be seen that the model obtained with LS performs much better for small values of j, so as predictions get further from k, the value of R^2 decreases. The models obtained using LM and PLS-PH however, are almost not affected by the value of j; thus, are good models to perform predictions inside the prediction window. All the latent variables have been used, $n_{\mathrm{lv}} = n_{\mathrm{x}}$, so PLS-PH equals LS-PH. For $n_a = n_b = \{2, 3\}$ some latent variables could have been removed due to collinearity between I_{load} and V_{stack}. This section however is intended to compare the proposed approach to LM

[1] n_{lv} is shown in the legend in the format PLS-PH($n_{\mathrm{lv}}/n_{\mathrm{x}}$), being n_{x} the number of columns in the regression matrix, then the maximum number of latent variables possible.

when variable reduction is not strictly necessary, thus, all the latent variables have been used.

7.3.2. Ill-Conditioned Example

The methanol/water distillation column model reported by Wood and Berry [Wood 73], is a typical MIMO plant with strong interaction between the controlled variables. The reflux (u_1) and the reboiler (u_2) steam flow rates are the process inputs. The compositions of the top (y_1) and bottom (y_2) products are the process outputs. The feed flow (ξ_1) and feed (ξ_2) composition rates are the disturbances.

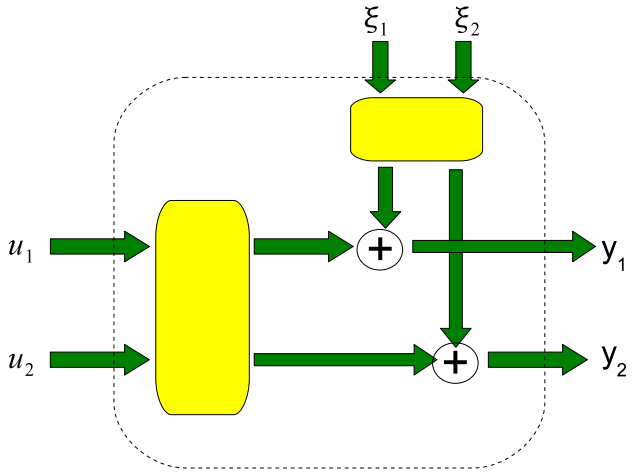


Figure 7.5: Block diagram of the distillation column.

The process delays are approximated by a second order Pade approximation. The discretized process, $T_s = 1$ minute, can be expressed:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \frac{0.06z^{-1}+0.63z^{-2}+0.072z^{-3}}{1-0.93z^{-1}-0.013z^{-2}-0.0023z^{-3}} & \frac{0.033z^{-1}+0.087z^{-2}-0.58z^{-3}}{1-1.57z^{-1}+0.72z^{-2}-0.13z^{-3}} \\ \frac{0.20z^{-1}-0.63z^{-2}+0.52z^{-3}}{1-2.18z^{-1}+1.58z^{-2}-0.39z^{-3}} & \frac{0.051z^{-1}+0.12z^{-2}-0.85z^{-3}}{1-1.55z^{-1}+0.71z^{-2}-0.13z^{-3}} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \begin{bmatrix} \frac{0.10z^{-1}-0.3z^{-2}+0.23z^{-3}}{1-2.29z^{-1}+1.74z^{-2}-0.45z^{-3}} & \frac{0.0037z^{-1}-0.011z^{-2}+0.0087z^{-3}}{1-2.28z^{-1}+1.72z^{-2}-0.44z^{-3}} \\ \frac{0.003z^{-1}-0.095z^{-2}+0.25z^{-3}}{1-1.65z^{-1}+0.84z^{-2}-0.16z^{-3}} & \frac{0.0052z^{-1}-0.014z^{-2}+0.01z^{-3}}{1-2.34z^{-1}+1.83z^{-2}-0.48z^{-3}} \end{bmatrix} \begin{bmatrix} \xi_1(k) \\ \xi_2(k) \end{bmatrix} \tag{7.17}$$

The signals to excite the process have been generated using the Matlab® idinput command with RGS, and BAND=$[0, 0.01]$. The amplitude of the white noise is such that the SNR (Signal to Noise Ratio) is approximately 10 DB for both outputs. The generated identification data set is shown in Fig. 7.6. The identification data set is mean-centered and scaled prior to being used in the identification algorithms.

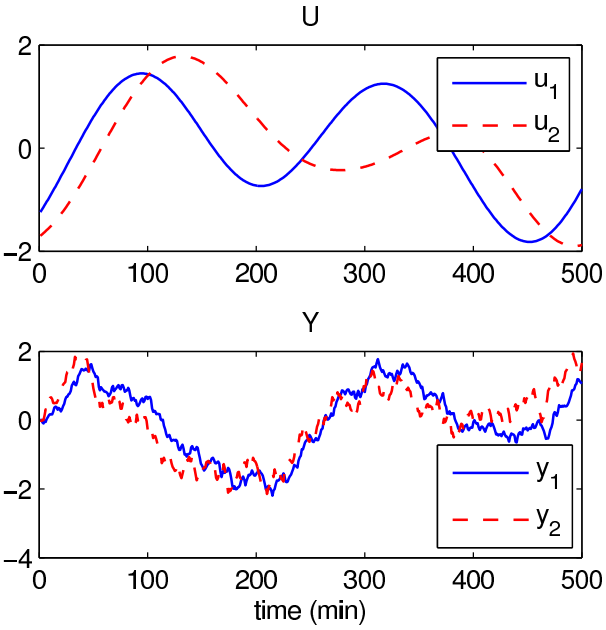


Figure 7.6: Identification data set for the Wood and Berry distillation column

Three MIMO ARX models, of orders $n_a = n_b = \{2, 3, 4\}$, are used to approximate the process. The prediction horizon is defined $n_f = 30$. The validation indicators, $J_{\text{LRPI}_{\text{EV}}}$ and $\mathbf{R}^2_{(30\times 2)}$, are shown in Fig. 7.7. The left plots are for the model $n_a = n_b = 2$, the middle plots are for $n_a = n_b = 3$, and the right plots are for $n_a = n_b = 4$. The graphs above show the vectors $\mathbf{R}^2_1$, and the graphs below show the vectors $\mathbf{R}^2_2$. The value of $J_{\text{LRPI}_{\text{EV}}}$ is shown in the legend.

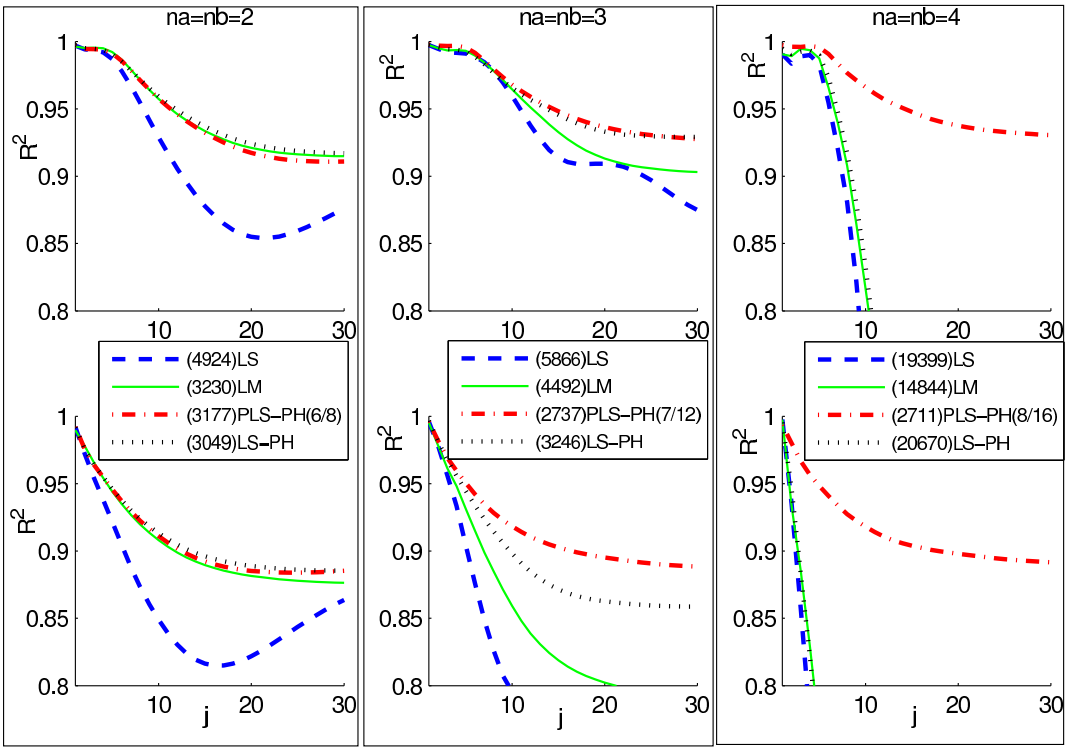


Figure 7.7: Validation results for the Wood and Berry distillation column

As can be seen in Fig. 7.7, a reduction in the number of variables has been performed in PLS-PH for the three model orders. Since $n_{\text{lv}} < n_x$, LS-PH and PLS-PH provide different models; the results for the models fitted with LS-PH are also shown in Fig. 7.7. Predictive performance of the PLS-PH model is

equivalent to the performance of the LM model for $n_\mathrm{a} = n_\mathrm{b} = 2$. PLS-PH clearly outperforms LM for $n_\mathrm{a} = n_\mathrm{b} = \{3, 4\}$. The number of latent variables used in PLS-PH differs more from n_x as the order of the model increases. For $n_\mathrm{a} = n_\mathrm{b} = 4$, $n_\mathrm{x} = 16$, $n_\mathrm{lv} = 8$, hence as the order of the model increases, there is more collinearity in the identification data set, and it is more important to reduce the number of variables.

For the three orders and identification algorithms compared, the best model in terms of $J_{\mathrm{LRPI}_\mathrm{EV}}$ is $n_\mathrm{a} = n_\mathrm{b} = 4$, fitted using PLS-PH with $n_\mathrm{lv} = 8$. LS is outperformed by all MRI methods, but it is included to see how the simplest method performs compared to the MRI approaches. It can be seen that PLS-PH is the only method that provides better models as the order of the model increases. The downwards trend in $J_{\mathrm{LRPI}_\mathrm{EV}}$ for PLS-PH for increasing orders of the model is expected to change for higher orders.

This example shows the importance of reducing the number of variables in the minimization of J_LRPI provided the identification data set is ill-conditioned. PLS-PH is presented as a minimization algorithm for J_LRPI when dealing with ill-conditioned identification data sets.

CHAPTER 8

CONCLUSIONS

This **Part II** of the thesis focuses on describing the ways to attain Model predictive control Relevant Identification (MRI). As explained at the beginning of this part, MRI can be implemented either with a parametric or a dynamic matrix approach. There are three main approaches for attaining parametric MRI: prefilter the identification data set; minimize a multi-step ahead cost function; or use subspace identification. This part provides the following two contributions for the parametric MRI approach:

- [Laurí 10c]: The MIMO and MISO identification approaches to model a MIMO plant have been compared both for traditional one-step ahead identification, and MRI. It is argued in this part of the thesis that, unlike in one-step ahead identification, the MIMO approach is preferable to the multiple MISO approach in MRI for a sufficiently large prediction horizon.

- [Laurí 10a]: A PLS line search numerical optimization approach to deal with parametric MRI has been proposed. The PLS-PH algorithm finds $\boldsymbol{\theta}$ which minimizes the multi-step ahead prediction error cost function even in the case of an ill-conditioned identification data set. PLS-PH is simpler than traditional numerical optimization methods provided derivatives of the non-linear cost function are not required. The PLS-PH approach outperforms LS in an MPC framework and also outperforms LM when the data is ill-conditioned, being equivalent to it otherwise. Then, PLS-PH is to be considered when identifying MIMO ARX models which are to be used to perform multi-step ahead predictions.

Summing-up, if a model to be used in MPC is needed: MIMO identification is preferable assuming the prediction horizon is large; and if excitation in the identification data set is limited either PLS-PH or the dynamic matrix approach fitted in the space of the latent variables are to be used. The following part of this thesis defines an MPC approach which uses MIMO identification and the dynamic matrix approach fitted with PLS.

Part III

LV-MPC: Latent Variable Model Predictive Control

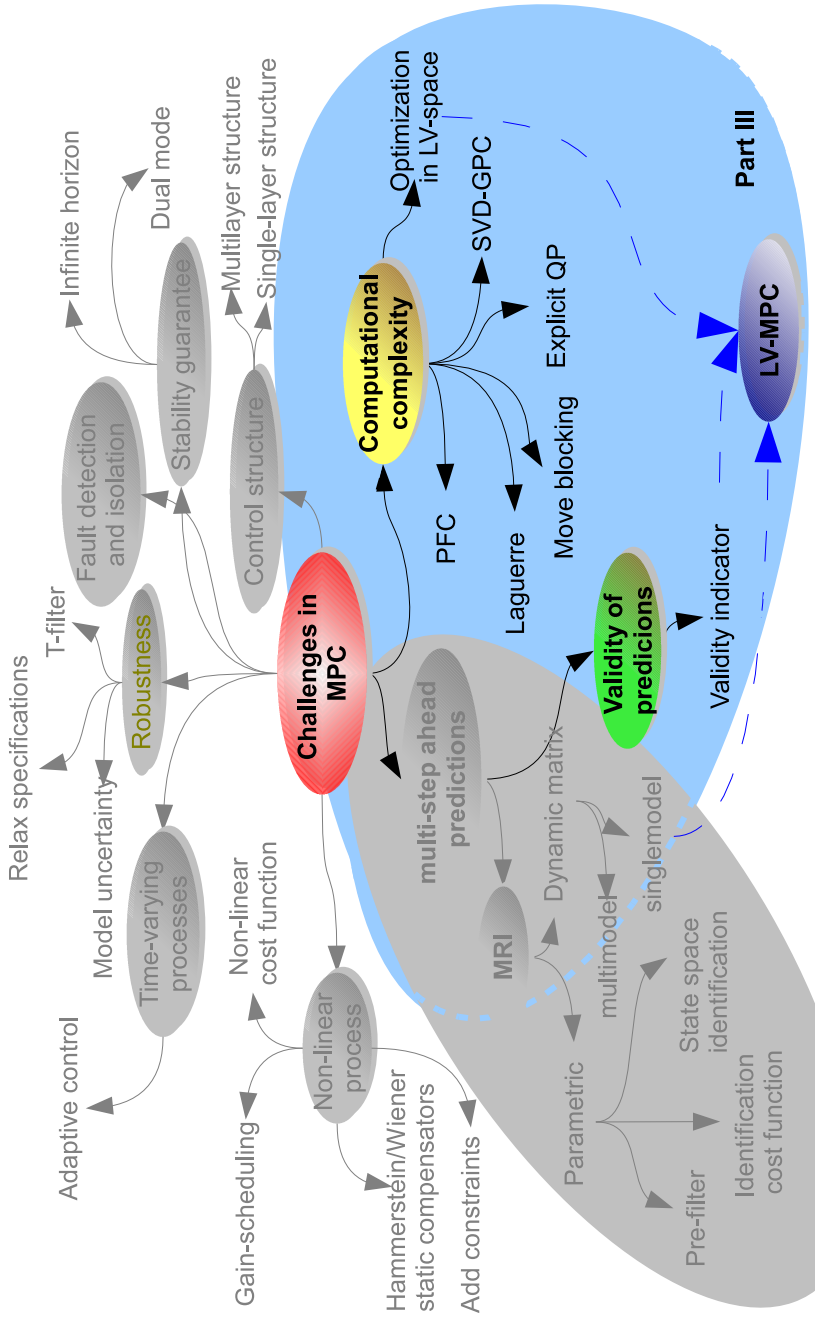


Figure 8.1: Location of PartIII in the general mindmap in Figure 1.4.

CHAPTER 9

LV-MPC METHODOLOGY

9.1. Introduction

This part introduces data-driven LV-MPC (Latent Variable Model Predictive Control) for continuous processes; in the sequel LV-MPC [Laurí 10b]. In LV-MPC the dynamic matrix used to perform multi-step ahead predictions is directly identified from the identification data set by means of PLS (Partial Least Squares), then the identification approach is MRI (Model predictive control Relevant Identification). The further predictive control is implemented in the reduced space of the latent variables. The features that motivate implementing the minimization in the space of the latent variables are:

- Computational complexity can be reduced provided decisions are taken in a reduced dimensional space.

- Indicators can be included in the controller to ensure validity of predictions, hence improving closed-loop performance.
- It can be more robust than traditional MPC provided projecting input data onto the latent variable space reduces the effect of noise.

This chapter is divided into 7 sections. In Section 9.2 the basic LV-MPC methodology is defined, this basic methodology is based on [Flores-Cerrillo 05] but deals with continuous processes instead of batch processes, and considers control and prediction horizons can be different. Sections 9.3 to 9.7 are devoted to add functionalities to the basic methodology yielding an enhanced LV-MPC control methodology. Finally, stability analysis is performed in Section 9.8.

9.2. Basic methodology

The PLS dynamic matrix predictor defined in Section 5.3 is to be used in the LV-MPC controller:

$$\hat{\mathbf{y}}_{\mathrm{f}}(k) = \overbrace{[\underbrace{\mathbf{u}_{\mathrm{p}}(k) \quad \mathbf{y}_{\mathrm{p}}(k)}_{\triangleq \mathbf{x}_{\mathrm{p}}(k)} \quad \mathbf{x}_{\mathrm{f}}(k) \quad \mathbf{x}_{\mathrm{dof}}(k)]}^{\mathbf{x}(k)} \underbrace{\underbrace{\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}}_{\mathbf{Z}}\mathbf{B}\mathbf{Q}^T}_{\boldsymbol{\theta}}$$

where $\boldsymbol{\theta}$ is the dynamic matrix with appropriate dimensions; $\mathbf{P}$, $\mathbf{W}$, $\mathbf{Q}$, and $\mathbf{B}$ are the matrices that define the PLS model, and

$$\begin{aligned}
\hat{\mathbf{y}}_{\mathrm{f}}(k) &= [\mathbf{y}_{\mathrm{k+1}} \ \cdots \ \mathbf{y}_{\mathrm{k+n_f}}] \\
\mathbf{u}_{\mathrm{p}}(k) &= [\mathbf{u}_{\mathrm{k-1}} \quad \mathbf{u}_{\mathrm{k-n_b+1}}] \\
\mathbf{y}_{\mathrm{p}}(k) &= [\mathbf{y}_{\mathrm{k-1}} \quad \mathbf{y}_{\mathrm{k-n_a}}] \\
\mathbf{x}_{\mathrm{f}}(k) &= [\mathbf{u}_{\mathrm{k+n_f-1}} \ \cdots \ \mathbf{u}_{\mathrm{k+n_u}}] \\
\mathbf{x}_{\mathrm{dof}}(k) &= [\mathbf{u}_{\mathrm{k+n_u-1}} \ \cdots \ \mathbf{u}_{\mathrm{k}}]
\end{aligned}$$

The LV-MPC cost function J_{C} in Equation (3.4) can be transformed to implement the minimization in the space of the latent variables:

$$\begin{aligned} J_{\mathrm{C}} &= ||[\mathbf{r}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)]\mathbf{W}_{\mathrm{y}}||_F^2 + \lambda_u||\Delta\mathbf{u}_{\mathrm{dof}}(k)\mathbf{W}_{\mathrm{u}}||_F^2 \\ &\Downarrow \\ J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}}) &= ||[\mathbf{r}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)]\mathbf{W}_{\mathrm{y}}||_F^2 + \lambda_u||\mathbf{x}_{\mathrm{dof}}(k)\mathbf{W}_{\mathrm{u}}||_F^2 \end{aligned} \tag{9.1}$$

where $\mathbf{t}_{\mathrm{d}}$ is the decision variable in the LV-space, and $\mathbf{x}_{\mathrm{dof}}$ contains the control sequence.

To perform the minimization of $J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})$; $\hat{\mathbf{y}}_{\mathrm{f}}(k)$ and $\mathbf{x}_{\mathrm{dof}}(k)$ are expressed in terms of $\mathbf{t}_{\mathrm{d}}$ as shown in propositions 9.2.2 and 9.2.1:[Hereafter the argument k is omitted for the sake of clarity.]

$$\begin{aligned} \mathbf{x}_{\mathrm{dof}} &= \mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}} \\ \hat{\mathbf{y}}_{\mathrm{f}} &= \mathbf{x}_{\mathrm{p}}\mathbf{S}_{\mathrm{p}} + \mathbf{t}_{\mathrm{d}}\mathbf{S}_{\mathrm{d}} \end{aligned}$$

In the absence of constraints, the minimization of $J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})$ yields the following analytic expression obtained in proposition 9.2.3:

$$\mathbf{t}_{\mathrm{d}} = -\mathbf{f}^T\mathbf{H}^{-1}$$

where

$$\begin{aligned} \mathbf{H} &= \mathbf{S}_{\mathrm{d}}\mathbf{W}_{\mathrm{y}}\mathbf{W}_{\mathrm{y}}^T\mathbf{S}_{\mathrm{d}}^T + \lambda_u\mathbf{M}_{\mathrm{dof}}\mathbf{W}_{\mathrm{u}}\mathbf{W}_{\mathrm{u}}^T\mathbf{M}_{\mathrm{dof}}^T \\ \mathbf{f}^T &= (\mathbf{x}_{\mathrm{p}}\mathbf{S}_{\mathrm{p}} - \mathbf{r}_{\mathrm{f}})\mathbf{W}_{\mathrm{y}}\mathbf{W}_{\mathrm{y}}^T\mathbf{S}_{\mathrm{d}} \end{aligned}$$

From $\mathbf{t}_{\mathrm{d}}$, $\mathbf{x}_{\mathrm{dof}}$ can be obtained as shown in proposition 9.2.1. Since the receding horizon policy is used, only the last[1] n_i elements in $\mathbf{x}_{\mathrm{dof}}$, $\mathbf{u}_{\mathrm{k}}$, are eventually applied to the process.

[1]Note from the definition of $\mathbf{x}_{\mathrm{dof}}$ in Equation (5.21) that the last n_i elements in $\mathbf{x}_{\mathrm{dof}}$ are $\mathbf{u}_{\mathrm{k}}$.

Proposition 9.2.1 $\mathbf{x}_{\mathrm{dof}}$ *can be expressed as a function of* $\mathbf{t}_{\mathrm{d}}$ *as*

$$\mathbf{x}_{\mathrm{dof}} = \mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}}$$

Proof *Equation* (4.4) *can be expressed at instant* k *as:*

$$\mathbf{t}(k) = \mathbf{x}(k)\mathbf{Z}. \tag{9.2}$$

$\mathbf{Z}$ *can be decomposed accordingly to* $\mathbf{x}(k)$ *in equation* (5.16)*:[Hereafter the argument* k *is omitted for the sake of clarity.]*

$$\begin{aligned} \mathbf{t} &= [\mathbf{x}_{\mathrm{p}} \quad \mathbf{x}_{\mathrm{f}} \quad \mathbf{x}_{\mathrm{dof}}] \begin{bmatrix} \mathbf{Z}_{\mathrm{p}} \\ \mathbf{Z}_{\mathrm{f}} \\ \mathbf{Z}_{\mathrm{dof}} \end{bmatrix} && (9.3) \\ &= \underbrace{\mathbf{x}_{\mathrm{p}}\mathbf{Z}_{\mathrm{p}}}_{\triangleq \mathbf{t}_{\mathrm{p}}} + \underbrace{\mathbf{x}_{\mathrm{f}}\mathbf{Z}_{\mathrm{f}}}_{\triangleq \mathbf{t}_{\mathrm{f}}} + \underbrace{\mathbf{x}_{\mathrm{dof}}\mathbf{Z}_{\mathrm{dof}}}_{\triangleq \mathbf{t}_{\mathrm{dof}}} && (9.4) \end{aligned}$$

thus,

$$\mathbf{t} = \mathbf{t}_{\mathrm{p}} + \mathbf{t}_{\mathrm{f}} + \mathbf{t}_{\mathrm{dof}}. \tag{9.5}$$

Provided $\mathbf{u}_{\mathrm{k+i}}$ *is set to* $\mathbf{u}_{\mathrm{k+n_u-1}}$ *for* $i \in [n_u, n_f - 1]$*, it can be shown from equations* (5.16)*,* (5.21)*, and* (5.20) *that*

$$\mathbf{x}_{\mathrm{f}} = \mathbf{x}_{\mathrm{dof}} \underbrace{\begin{bmatrix} \mathbf{I}_{\mathrm{n_i}} & \dots & \mathbf{I}_{\mathrm{n_i}} \\ \mathbf{0}_{(\mathrm{n_u}-1)\mathrm{n_i}\times(\mathrm{n_f}-\mathrm{n_u})\mathrm{n_i}} \end{bmatrix}}_{\triangleq \mathbf{\Gamma}} \tag{9.6}$$

From Equation (9.4)*,* $\mathbf{t}_{\mathrm{p}}$ *depends on known past data; and from Equations* (9.4) *and* (9.6)*,* $\mathbf{t}_{\mathrm{f}}$ *depends on* $\mathbf{t}_{\mathrm{dof}}$*. Hence, the decision vector in the latent variable space can be set*

$$\mathbf{t}_{\mathrm{d}} \triangleq \mathbf{t}_{\mathrm{dof}} \tag{9.7}$$

And Equation (9.5) *can be expressed*

$$\mathbf{t} = \mathbf{t}_\mathrm{p} + \mathbf{t}_\mathrm{f} + \mathbf{t}_\mathrm{d}.$$

From Equations (9.4) *and* (9.7)

$$\mathbf{t}_\mathrm{d} = \mathbf{x}_\mathrm{dof}\mathbf{Z}_\mathrm{dof}$$

Clearing $\mathbf{x}_\mathrm{dof}$ *yields:*

$$\mathbf{x}_\mathrm{dof} = \mathbf{t}_\mathrm{d}\underbrace{(\mathbf{Z}_\mathrm{dof}^T\mathbf{Z}_\mathrm{dof})^{-1}\mathbf{Z}_\mathrm{dof}^T}_{\triangleq\mathbf{M}_\mathrm{dof}}. \tag{9.8}$$

One may think $\mathbf{x}_\mathrm{dof}$ *could be more easily obtained from the PLS Equation* (4.2) $\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}$. *It is proven in Appendix C however, that such formulation is not consistent with using* $\mathbf{t}$ *to perform predictions.* □

Proposition 9.2.2 $\hat{\mathbf{y}}_\mathrm{f}$ *can be expressed as a function of* $\mathbf{t}_\mathrm{d}$ *as*

$$\hat{\mathbf{y}}_\mathrm{f} = \mathbf{x}_\mathrm{p}\mathbf{S}_\mathrm{p} + \mathbf{t}_\mathrm{d}\mathbf{S}_\mathrm{d}$$

Proof *Equation* (4.5) *can be expressed at an instant* k

$$\hat{\mathbf{y}}_\mathrm{f} = \mathbf{t}\mathbf{B}\mathbf{Q}^T = (\mathbf{t}_\mathrm{p} + \underbrace{\mathbf{t}_\mathrm{f} + \mathbf{t}_\mathrm{dof}}_{\triangleq\mathbf{t}_\mathrm{fdof}})\mathbf{B}\mathbf{Q}^T \tag{9.9}$$

Substituting in $\mathbf{t}_\mathrm{p}$ *from* (9.4)

$$\hat{\mathbf{y}}_\mathrm{f} = (\mathbf{x}_\mathrm{p}\mathbf{Z}_\mathrm{p} + \mathbf{t}_\mathrm{fdof})\mathbf{B}\mathbf{Q}^T \tag{9.10}$$

From equation (9.6) *and proposition 9.2.1*

$$\mathbf{x}_{\mathrm{f}} = \mathbf{x}_{\mathrm{dof}}\mathbf{\Gamma} \quad \Rightarrow \quad \mathbf{x}_{\mathrm{f}} = \underbrace{\mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}}}_{\mathbf{x}_{\mathrm{dof}}}\mathbf{\Gamma}$$

substituting $\mathbf{x}_{\mathrm{f}}$ *in Equation* (9.4)

$$\mathbf{t}_{\mathrm{f}} = \mathbf{x}_{\mathrm{f}}\mathbf{Z}_{\mathrm{f}} \quad \Rightarrow \quad \mathbf{t}_{\mathrm{f}} = \underbrace{\mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}}\mathbf{\Gamma}}_{\mathbf{x}_{\mathrm{f}}}\mathbf{Z}_{\mathrm{f}}$$

Substituting in the definition of $\mathbf{t}_{\mathrm{fdof}}$ *in Equation* (9.9)

$$\begin{aligned} \mathbf{t}_{\mathrm{fdof}} &= \mathbf{t}_{\mathrm{f}} + \mathbf{t}_{\mathrm{dof}} \\ &= \underbrace{\mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}}\mathbf{\Gamma}\mathbf{Z}_{\mathrm{f}}}_{\mathbf{t}_{\mathrm{f}}} + \mathbf{t}_{\mathrm{d}} \\ &= \mathbf{t}_{\mathrm{d}}\underbrace{(\mathbf{M}_{\mathrm{dof}}\mathbf{\Gamma}\mathbf{Z}_{\mathrm{f}} + \mathbf{I}_{\mathrm{n_{lv}}})}_{\triangleq \mathbf{M}_{\mathrm{t}}}. \end{aligned} \tag{9.11}$$

Substituting equation (9.11) *in equation* (9.10)

$$\hat{\mathbf{y}}_{\mathrm{f}} = \mathbf{x}_{\mathrm{p}}\underbrace{\mathbf{Z}_{\mathrm{p}}\mathbf{B}\mathbf{Q}^{T}}_{\triangleq \mathbf{S}_{\mathrm{p}}} + \mathbf{t}_{\mathrm{d}}\underbrace{\mathbf{M}_{\mathrm{t}}\mathbf{B}\mathbf{Q}^{T}}_{\triangleq \mathbf{S}_{\mathrm{d}}}. \quad \square$$

Proposition 9.2.3 *The value of* $\mathbf{t}_{\mathrm{d}}$ *that minimizes* $J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})$ *is*

$$\mathbf{t}_{\mathrm{d}} = -\mathbf{f}^{T}\mathbf{H}^{-1}$$

Proof *Substituting* $\hat{\mathbf{y}}_{\mathrm{f}}$ *and* $\mathbf{x}_{\mathrm{dof}}$ *from propositions 9.2.2 and 9.2.1 in Equation* (9.1) *yields:*

$$J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}}) = |||[\mathbf{r}_{\mathrm{f}} - \underbrace{(\mathbf{x}_{\mathrm{p}}\mathbf{S}_{\mathrm{p}} + \mathbf{t}_{\mathrm{d}}\mathbf{S}_{\mathrm{d}})}_{\hat{\mathbf{y}}_{\mathrm{f}}}]\mathbf{W}_{\mathrm{y}}||_{F}^{2} + \lambda_{u}||\underbrace{\mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{dof}}}_{\mathbf{x}_{\mathrm{dof}}}\mathbf{W}_{\mathrm{u}}||_{F}^{2}$$

which operating yields

$$J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}}) = \mathbf{t}_{\mathrm{d}}\mathbf{H}\mathbf{t}_{\mathrm{d}}^{T} + 2\mathbf{f}^{T}\mathbf{t}_{\mathrm{d}}^{T} + C$$

where

$$\begin{aligned} \mathbf{H} &= \mathbf{S}_{\mathrm{d}}\mathbf{W}_{\mathrm{y}}\mathbf{W}_{\mathrm{y}}^{T}\mathbf{S}_{\mathrm{d}}^{T} + \lambda_{u}\mathbf{M}_{\mathrm{dof}}\mathbf{W}_{\mathrm{u}}\mathbf{W}_{\mathrm{u}}^{T}\mathbf{M}_{\mathrm{dof}}^{T} && (9.12)\\ \mathbf{f}^{T} &= (\mathbf{x}_{\mathrm{p}}\mathbf{S}_{\mathrm{p}} - \mathbf{r}_{\mathrm{f}})\mathbf{W}_{\mathrm{y}}\mathbf{W}_{\mathrm{y}}^{T}\mathbf{S}_{\mathrm{d}}^{T} && (9.13)\\ C &= ||(\mathbf{r}_{\mathrm{f}} - \mathbf{x}_{\mathrm{p}}\mathbf{S}_{\mathrm{p}})\mathbf{W}_{\mathrm{y}}||_{F}^{2} \end{aligned}$$

The minimum of $J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})$ is attained by equating its first derivative to zero:(Note the Hessian matrix $\mathbf{H}$ is symmetric, then $\mathbf{H} = \mathbf{H}^{T}$)

$$\begin{aligned} \frac{\partial J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})}{\partial \mathbf{t}_{\mathrm{d}}} &= 0\\ (\mathbf{H} + \mathbf{H}^{T})\mathbf{t}_{\mathrm{d}}^{T} + 2\mathbf{f} &= 0\\ 2\mathbf{H}\mathbf{t}_{\mathrm{d}}^{T} + 2\mathbf{f} &= 0\\ \mathbf{H}\mathbf{t}_{\mathrm{d}}^{T} + \mathbf{f} &= 0\\ \mathbf{t}_{\mathrm{d}}\mathbf{H} + \mathbf{f}^{T} &= 0\\ \mathbf{t}_{\mathrm{d}}\mathbf{H} &= -\mathbf{f}^{T}\\ \mathbf{t}_{\mathrm{d}} &= -\mathbf{f}^{T}\mathbf{H}^{-1}.\square && (9.14) \end{aligned}$$

9.3. Improving the Hessian conditioning

The conditioning number of the Hessian matrix $\mathbf{H}$ in Equation (9.12) required to minimise $J_{\mathrm{C}}(\mathbf{t}_{\mathrm{d}})$ of Equation (9.1) is large whenever the matrices $\mathbf{M}_{\mathrm{dof}}$ in proposition 9.2.1 and $\mathbf{M}_{\mathrm{t}}$ in Equation (9.11) have a large conditioning number. The larger the conditioning number of H, the more numerical roundoff errors

and the slower the convergence of iterative methods[2] used to minimise (9.1). In this section, matrices $\mathbf{M}_{\text{dof}}$ and $\mathbf{M}_{\text{t}}$ are obtained in a different manner to improve the Hessian conditioning.

- $\mathbf{M}_{\text{dof}}$ is the transformation matrix that projects from the decision variable $\mathbf{t}_{\text{d}}$ to the space of control moves $\mathbf{x}_{\text{dof}}$.
- $\mathbf{M}_{\text{t}}$ is the transformation matrix that moves from the decision variable $\mathbf{t}_{\text{d}}$ to the scores of future control moves $\mathbf{t}_{\text{fdof}} = \mathbf{t}_{\text{f}} + \mathbf{t}_{\text{dof}}$.

To illustrate the proposed solution, lets take an example with 1 variable in $\mathbf{x}_{\text{f}}$, $\mathbf{x}_{\text{dof}}$, and $\mathbf{t}$. Two options, A and B, may be considered. In each option $\mathbf{x}_{\text{dof}}$ is obtained from $\mathbf{t}_{\text{d}}$ in a different manner. Provided there is only one variable in $\mathbf{x}_{\text{f}}$ and $\mathbf{x}_{\text{dof}}$, $\mathbf{x}_{\text{f}}$ can be easily obtained from $\mathbf{x}_{\text{dof}}$ since $\mathbf{x}_{\text{f}} = \mathbf{x}_{\text{dof}}$. $\mathbf{x}_{\text{fdof}}$ is obtained as $\mathbf{x}_{\text{fdof}} = [\mathbf{x}_{\text{f}} \quad \mathbf{x}_{\text{dof}}]$, and $\mathbf{t}_{\text{fdof}}$ is obtained projecting $\mathbf{x}_{\text{fdof}}$ to the inner space. Different ways to obtain $\mathbf{x}_{\text{dof}}$ from $\mathbf{t}_{\text{d}}$ provide different matrices $\mathbf{M}_{\text{dof}}$ and $\mathbf{M}_{\text{t}}$ which provides different Hessian matrices $\mathbf{H}$.

- Option A is shown in Figure 9.1. $\mathbf{x}_{\text{dof}}$ can be obtained projecting $\mathbf{t}_{\text{d}}$ to the outer space[3]. Provided $\mathbf{x}_{\text{f}} = \mathbf{x}_{\text{dof}}$, $\mathbf{x}_{\text{fdof}} = [\mathbf{x}_{\text{f}} \quad \mathbf{x}_{\text{dof}}]$. This can be graphically seen in Figure 9.1, where $\mathbf{x}_{\text{fdof}}$ is obtained as the intersection between the lines $\mathbf{y} = \mathbf{x}$ and $\mathbf{y} = \mathbf{x}_{\text{dof}}$. $\mathbf{t}_{\text{fdof}}$ can be obtained projecting $\mathbf{x}_{\text{fdof}}$ onto the latent variable space.
- Option B is shown in Figure 9.2. Given $\mathbf{t}_{\text{d}}$, $\mathbf{x}_{\text{fdof}}$ can be obtained as the intersection between the line $\mathbf{y} = \mathbf{x}$, and a line normal to $\mathbf{t}$ that passes through the projection of $\mathbf{t}_{\text{d}}$ to the outer space. $\mathbf{x}_{\text{dof}}$ can be obtained as part of $\mathbf{x}_{\text{fdof}}$.

[2] If inequality constraints are added to the problem in Equation (9.1), as seen in Section 9.5, iterative methods need be used to solve it.

[3] The outer space is defined as the space of future manipulated variables. In this simple example the outer space is [x, y].

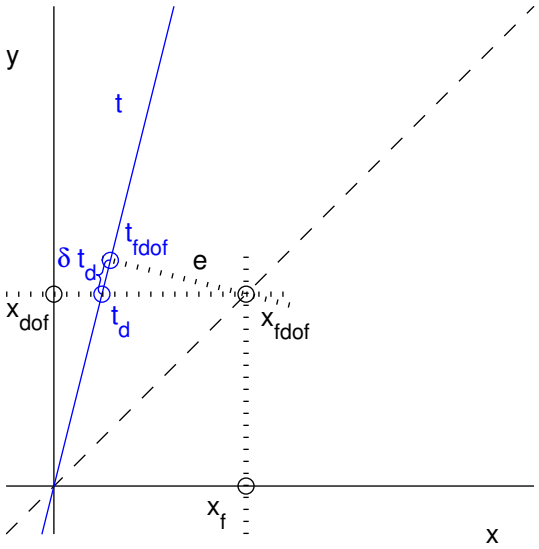


Figure 9.1: Option A

Both options may lead to an ill-conditioned Hessian matrix for a large number of dimensions; in option A because of $\delta\mathbf{t}_\mathrm{d}$, and in option B because of $\mathbf{e}$. The solution presented in this section settles in-between options A and B, see Figure 9.3. $\mathbf{x}_\mathrm{fdof}$ can be obtained as the intersection between the line $\mathbf{y} = \mathbf{x}$ and a line normal to $\mathbf{t}$. Provided there is an infinite number of lines normal to $\mathbf{t}$, there is an infinite number of solutions to this problem. The additional consideration is to minimize the squared 2-norm of vectors $\mathbf{e}$ and $\delta\mathbf{t}_\mathrm{d}$.

A more formal presentation of this intuitive concept is given next. For clarity, first the results are summarised and then the required propositions are given at the end of the section for completeness.

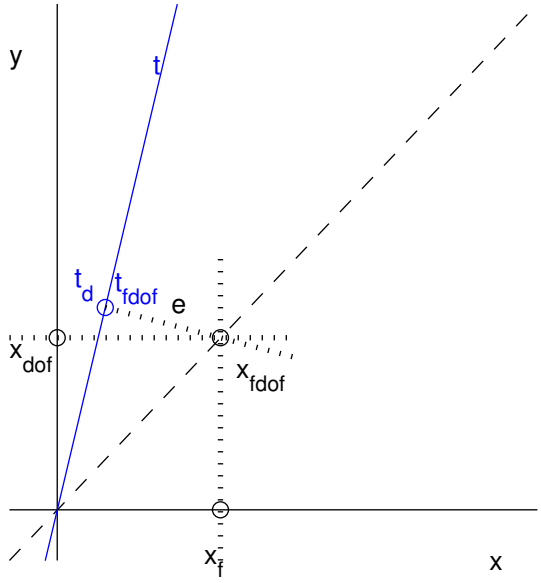


Figure 9.2: Option B

Vector γ in Figure 9.3 can be obtained from the minimisation:

$$\min_{\gamma} \quad \alpha||\mathbf{e}||_F^2 + \beta||\delta\mathbf{t}_\mathrm{d}||_F^2 \quad s.t. \quad \mathbf{x}_\mathrm{fdof}\mathbf{\Omega} = \mathbf{0} \tag{9.15}$$

where: $\mathbf{x}_\mathrm{fdof}$, $\delta\mathbf{t}_\mathrm{d}$, and $\mathbf{e}$ are expressed as functions of γ in propositions 9.3.1-9.3.3, and

$$\mathbf{\Omega} = \begin{bmatrix} -\mathbf{I}_{\mathrm{n_i}} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\mathbf{I}_{\mathrm{n_i}} \\ \mathbf{I}_{\mathrm{n_i}} & \dots & \mathbf{I}_{\mathrm{n_i}} \\ \mathbf{0}_{\mathrm{n_i(n_u-1)\times(n_f-n_u)n_i}} & & \end{bmatrix}$$

define the constraints such that Equation (9.6) holds.

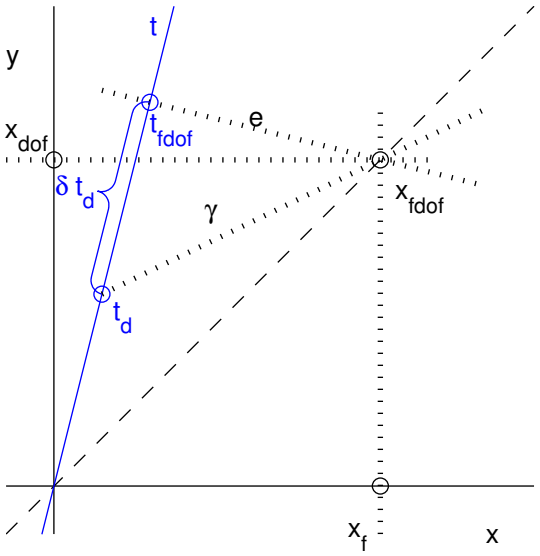


Figure 9.3: Option C

The problem in Equation (9.15) is a QP with equality constraints that can be solved using Lagrange multipliers [Wang 09]. Note the minimization problem has a unique solution provided γ has $n_{\mathrm{f}} n_{\mathrm{i}}$ elements, and $\mathbf{\Omega}$ adds $(n_{\mathrm{f}} - n_{\mathrm{u}})n_{\mathrm{i}}$ linearly independent constraints. From Proposition 9.3.4 the solution to the minimization problem is

$$\gamma = \mathbf{t}_{\mathrm{d}} \mathbf{M}_{\mathrm{QP}}. \tag{9.16}$$

$\mathbf{t}_{\mathrm{fdof}}$ equals the decision variable $\mathbf{t}_{\mathrm{d}}$, plus the modification due to setting future control actions passed instant $k + n_{\mathrm{u}} - 1$ equal to control actions at instant $k + n_{\mathrm{u}} - 1$

$$\mathbf{t}_{\mathrm{fdof}} = \mathbf{t}_{\mathrm{d}} + \delta \mathbf{t}_{\mathrm{d}} \tag{9.17}$$

substituting in $\delta\mathbf{t}_\mathrm{d}$ from propositions 9.3.2 and 9.3.4

$$\delta\mathbf{t}_\mathrm{d} = \underbrace{\mathbf{t}_\mathrm{d}\mathbf{M}_\mathrm{QP}}_{\gamma}\mathbf{Z}_\mathrm{fdof} \quad \Rightarrow \quad \mathbf{t}_\mathrm{fdof} = \mathbf{t}_\mathrm{d}\underbrace{(\mathbf{M}_\mathrm{QP}\mathbf{Z}_\mathrm{fdof} + \mathbf{I})}_{\bar{\mathbf{M}}_\mathrm{t}}. \tag{9.18}$$

From propositions 9.3.1 and 9.3.4

$$\mathbf{x}_\mathrm{fdof} = \mathbf{t}_\mathrm{d}(\mathbf{M}_\mathrm{QP} + \mathbf{Z}_\mathrm{fdof}^{-1*}) \tag{9.19}$$

$\mathbf{x}_\mathrm{dof}$ is a part of $\mathbf{x}_\mathrm{fdof}$ that can be expressed

$$\mathbf{x}_\mathrm{dof} = \underbrace{\mathbf{t}_\mathrm{d}(\mathbf{M}_\mathrm{QP} + \mathbf{Z}_\mathrm{fdof}^{-1*}) \begin{bmatrix} \mathbf{0}_{(\mathrm{n_f}-\mathrm{n_u})\mathrm{n_i}\times\mathrm{n_u}\mathrm{n_i}} \\ \mathbf{I}_{\mathrm{n_u}\mathrm{n_i}} \end{bmatrix}}_{\triangleq\bar{\mathbf{M}}_\mathrm{dof}}. \tag{9.20}$$

Summing up, to improve the Hessian conditioning in the basic LV-MPC methodology; matrices $\mathbf{M}_\mathrm{dof}$ and $\mathbf{M}_\mathrm{t}$ in Equations (9.8) and (9.11) need be replaced by $\bar{\mathbf{M}}_\mathrm{dof}$ and $\bar{\mathbf{M}}_\mathrm{t}$ in Equations (9.20) and (9.18). α and β are tuning parameters to improve Hessian conditioning. As proven in Appendix D, this methodology is consistent.

Proposition 9.3.1 $\mathbf{x}_\mathrm{fdof}$ *can be expressed as a function of* γ *as*

$$\mathbf{x}_\mathrm{fdof} = \gamma + \mathbf{t}_\mathrm{d}\mathbf{Z}_\mathrm{fdof}^{-1*}$$

Proof *Substituting* $\mathbf{t}$ *from equation* (9.5) *in equation* (9.3)

$$\mathbf{t}_\mathrm{p} + \underbrace{\mathbf{t}_\mathrm{f} + \mathbf{t}_\mathrm{dof}}_{\mathbf{t}_\mathrm{fdof}} = [\mathbf{x}_\mathrm{p} \quad \underbrace{\mathbf{x}_\mathrm{f} \quad \mathbf{x}_\mathrm{dof}}_{\mathbf{x}_\mathrm{fdof}}] \left.\begin{bmatrix} \mathbf{Z}_\mathrm{p} \\ \mathbf{Z}_\mathrm{f} \\ \mathbf{Z}_\mathrm{dof} \end{bmatrix}\right\} \triangleq \mathbf{Z}_\mathrm{fdof} \tag{9.21}$$

thus, given $\mathbf{t}_{\text{fdof}}$*,*

$$\mathbf{t}_{\text{fdof}} = \mathbf{x}_{\text{fdof}}\mathbf{Z}_{\text{fdof}} \quad \Rightarrow \quad \mathbf{x}_{\text{fdof}} = \mathbf{t}_{\text{fdof}}\underbrace{(\mathbf{Z}_{\text{fdof}}^T\mathbf{Z}_{\text{fdof}})^{-1}\mathbf{Z}_{\text{fdof}}^T}_{\triangleq\mathbf{Z}_{\text{fdof}}^{-1*}} \tag{9.22}$$

matrix $\mathbf{Z}_{\text{fdof}}$ *projects from the outer space to the inner space, and matrix* $\mathbf{Z}_{\text{fdof}}^{-1*}$ *projects from the inner space to the outer space.*

It can be seen from Figure 9.3 that vector $\mathbf{x}_{\text{fdof}}$ *equals the projection of* $\mathbf{t}_{\text{d}}$ *to the outer space plus* γ

$$\mathbf{x}_{\text{fdof}} = \gamma + \mathbf{t}_{\text{d}}\mathbf{Z}_{\text{fdof}}^{-1*}. \quad \square$$

Proposition 9.3.2 $\delta\mathbf{t}_{\text{d}}$ *can be expressed as a function of* γ *as*

$$\delta\mathbf{t}_{\text{d}} = \gamma\mathbf{Z}_{\text{fdof}}$$

Proof $\delta\mathbf{t}_{\text{d}}$ *is obtained projecting* γ *to the inner space (see Figure 9.3 and Equation* (9.21)*)*

$$\delta\mathbf{t}_{\text{d}} = \gamma\mathbf{Z}_{\text{fdof}}. \quad \square$$

Proposition 9.3.3 $\mathbf{e}$ *can be expressed as a function of* γ *as*

$$\mathbf{e} = \gamma(\mathbf{I} - \mathbf{Z}_{\text{fdof}}\mathbf{Z}_{\text{fdof}}^{-1*})$$

Proof *It can be seen from Figure 9.3 that the projection of* $\delta\mathbf{t}_{\mathrm{d}}$ *to the outer space plus* $\mathbf{e}$*, equals vector* γ

$$\begin{aligned}\delta\mathbf{t}_{\mathrm{d}}\mathbf{Z}_{\mathrm{fdof}}^{-1*}+\mathbf{e} &= \gamma\\ \mathbf{e} &= \gamma-\delta\mathbf{t}_{\mathrm{d}}\mathbf{Z}_{\mathrm{fdof}}^{-1*}\end{aligned}$$

Substituting in from proposition 9.3.2

$$\mathbf{e}=\gamma(\mathbf{I}-\mathbf{Z}_{\mathrm{fdof}}\mathbf{Z}_{\mathrm{fdof}}^{-1*}). \quad \square$$

Proposition 9.3.4 *The QP with equality constraints in Equation* (9.15) *yields the following value of* γ*:*

$$\gamma=\mathbf{t}_{\mathrm{d}}\mathbf{M}_{\mathrm{QP}}$$

Proof *Substituting in from propositions 9.3.3, 9.3.2, 9.3.1, the QP in Equation* (9.15) *yields*

$$\begin{aligned}&\min_{\gamma} \quad \alpha||\mathbf{e}||_F^2+\beta||\delta\mathbf{t}_{\mathrm{d}}||_F^2 \quad s.t. \quad \mathbf{x}_{\mathrm{fdof}}\mathbf{\Omega}=\mathbf{0}\\ \min_{\gamma} \quad &\alpha||\gamma(\mathbf{I}-\mathbf{Z}_{\mathrm{fdof}}\mathbf{Z}_{\mathrm{fdof}}^{-1*})||_F^2+\beta||\gamma\mathbf{Z}_{\mathrm{fdof}}||_F^2 \quad s.t. \quad [\gamma+\mathbf{t}_{\mathrm{d}}\mathbf{Z}_{\mathrm{fdof}}^{-1*}]\mathbf{\Omega}=\mathbf{0}\end{aligned}$$

which operating yields

$$\min_{\gamma} \quad \gamma\mathbf{L}\gamma^T \quad s.t. \quad \mathbf{M}\gamma^T=\mathbf{N}$$

where

$$\begin{aligned}\mathbf{L} &= \alpha(\mathbf{I}-\mathbf{Z}_{\mathrm{fdof}}\mathbf{Z}_{\mathrm{fdof}}^{-1*})(\mathbf{I}-\mathbf{Z}_{\mathrm{fdof}}\mathbf{Z}_{\mathrm{fdof}}^{-1*})^T+\beta\mathbf{Z}_{\mathrm{fdof}}\mathbf{Z}_{\mathrm{fdof}}^T && (9.23)\\ \mathbf{M} &= \mathbf{\Omega}^T && (9.24)\\ \mathbf{N}^T &= -\mathbf{t}_{\mathrm{d}}\mathbf{Z}_{\mathrm{fdof}}^{-1}\mathbf{\Omega} && (9.25)\end{aligned}$$

Applying Lagrange multipliers, the solution of the problem is [Wang 09]:

$$\gamma^T = \mathbf{L}^{-1}\mathbf{M}^T(\mathbf{M}\mathbf{L}^{-1}\mathbf{M}^T)^{-1}\mathbf{N} \tag{9.26}$$

thus

$$\begin{aligned}
\gamma &= \mathbf{N}^T(\mathbf{L}^{-1}\mathbf{M}^T(\mathbf{M}\mathbf{L}^{-1}\mathbf{M}^T)^{-1})^T \\
\gamma &= \underbrace{-\mathbf{t}_\mathrm{d}\mathbf{Z}_\mathrm{fdof}^{-1}\boldsymbol{\Omega}}_{\mathbf{N}^T}(\mathbf{L}^{-1}\mathbf{M}^T(\mathbf{M}\mathbf{L}^{-1}\mathbf{M}^T)^{-1})^T \\
\gamma &= \mathbf{t}_\mathrm{d}\underbrace{[-\mathbf{Z}_\mathrm{fdof}^{-1}\boldsymbol{\Omega}(\mathbf{L}^{-1}\mathbf{M}^T(\mathbf{M}\mathbf{L}^{-1}\mathbf{M}^T)^{-1})^T]}_{\triangleq\mathbf{M}_\mathrm{QP}}. \quad \square
\end{aligned} \tag{9.27}$$

9.4. Offset-free tracking

One requirement for offset-free tracking in MPC is that the MPC cost function in Equation (9.1) equals zero if and only if the system is at the correct steady state [Rossiter 03]. This has a knock on effect that, in steady state, the predictions of the CVs must be unbiased, and typically one uses $\Delta\mathbf{x}_\mathrm{dof}$ instead of $\mathbf{x}_\mathrm{dof}$ in Equation (9.1) although alternatives do exist for this term.

One means of ensuring unbiased predictions is through the appropriate inclusion of an integrated white noise disturbance model. As shown in proposition 9.4.1, the predictor form proposition 9.2.2 can be replaced by the following expression to attain unbiased predictions: [4]

$$\hat{\mathbf{y}}_\mathrm{f} = \mathbf{x}_\mathrm{p}^*\bar{\mathbf{S}}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\bar{\mathbf{S}}_\mathrm{d}. \tag{9.28}$$

[4] Note this predictor is based on a different model that predictor in Proposition 9.2.2, but the same nomenclature is used in the sake of simplicity of notation.

The cost function (or implied minimisation) of Equation (9.1) can be modified as follows to enable offset-free tracking:

$$\underset{\Delta \mathbf{t}_{\mathrm{d}}}{min} \;\; J_{\mathrm{C}}(\Delta \mathbf{t}_{\mathrm{d}}) = ||[\mathbf{r}_{\mathrm{f}} - \hat{\mathbf{y}}_{\mathrm{f}}]\mathbf{W}_{\mathrm{y}}||_F^2 + \lambda_u ||\Delta \mathbf{x}_{\mathrm{dof}} \mathbf{W}_{\mathrm{u}}||_F^2 \tag{9.29}$$

To minimize $J_{\mathrm{C}}(\Delta \mathbf{t}_{\mathrm{d}})$, $\hat{\mathbf{y}}_{\mathrm{f}}$ and $\Delta \mathbf{x}_{\mathrm{dof}}$ are expressed in terms of $\Delta \mathbf{t}_{\mathrm{d}}$ in proposition 9.4.1 and Equation (9.31). In the absence of constraints, the minimization of $J_{\mathrm{C}}(\Delta \mathbf{t}_{\mathrm{d}})$ yields the following analytic expression obtained in proposition 9.4.3:

$$\Delta \mathbf{t}_{\mathrm{d}} = -\mathbf{f}^T \mathbf{H}^{-1}$$

where

$$\begin{aligned} \mathbf{H} &= \bar{\mathbf{S}}_{\mathrm{d}} \mathbf{W}_{\mathrm{y}} \mathbf{W}_{\mathrm{y}}^T \bar{\mathbf{S}}_{\mathrm{d}}^T + \lambda_u \mathbf{M}_{\mathrm{dof}} \mathbf{W}_{\mathrm{u}} \mathbf{W}_{\mathrm{u}}^T \mathbf{M}_{\mathrm{dof}}^T \\ \mathbf{f}^T &= (\mathbf{x}_{\mathrm{p}}^* \bar{\mathbf{S}}_{\mathrm{p}} - \mathbf{r}_{\mathrm{f}}) \mathbf{W}_{\mathrm{y}} \mathbf{W}_{\mathrm{y}}^T \bar{\mathbf{S}}_{\mathrm{d}}^T \end{aligned}$$

$\mathbf{x}_{\mathrm{dof}}$ can be obtained form $\Delta \mathbf{t}_{\mathrm{d}}$ as shown in proposition 9.4.2. Provided the receding horizon policy is used, only the last n_i elements in $\mathbf{x}_{\mathrm{dof}}$, $\mathbf{u}_{\mathrm{k}}$, are eventually applied to the process. Note from the definition of $\mathbf{x}_{\mathrm{dof}}$ in Equation (5.21) that the last n_i elements in $\mathbf{x}_{\mathrm{dof}}$ are $\mathbf{u}_{\mathrm{k}}$.

Proposition 9.4.1 *The dependence of* $\hat{\mathbf{y}}_{\mathrm{f}}$ *on* $\Delta \mathbf{t}_{\mathrm{d}}$ *is given as*

$$\hat{\mathbf{y}}_{\mathrm{f}} = \mathbf{x}_{\mathrm{p}}^* \bar{\mathbf{S}}_{\mathrm{p}} + \Delta \mathbf{t}_{\mathrm{d}} \bar{\mathbf{S}}_{\mathrm{d}}$$

Proof *From [Huang 08] the model with integrated white noise can be expressed from the model in Equation* (9.9)*:*

$$\hat{\mathbf{y}}_{\mathrm{f}} = \Delta((\mathbf{t}_{\mathrm{p}} + \mathbf{t}_{\mathrm{f}} + \mathbf{t}_{\mathrm{dof}})\mathbf{B}\mathbf{Q}^T) \underbrace{\begin{bmatrix} \mathbf{I}_{\mathrm{n_o}} & \dots & \mathbf{I}_{\mathrm{n_o}} \\ & \ddots & \vdots \\ \mathbf{0} & & \mathbf{I}_{\mathrm{n_o}} \end{bmatrix}}_{\triangleq \mathbf{\Psi}_{\mathrm{y}}} + \mathbf{y}_{\mathrm{k}} \underbrace{[\mathbf{I}_{\mathrm{n_o}} \quad \dots \quad \mathbf{I}_{\mathrm{n_o}}]}_{\triangleq \mathbf{\Phi}_{\mathrm{y}}}$$

where $\Delta = 1 - z^{-1}$. *Substituting in from Equation* (9.4)

$$\hat{\mathbf{y}}_\mathrm{f} = (\Delta\mathbf{x}_\mathrm{p}\mathbf{Z}_\mathrm{p} + \Delta\mathbf{x}_\mathrm{f}\mathbf{Z}_\mathrm{f} + \Delta\mathbf{t}_\mathrm{dof})\mathbf{B}\mathbf{Q}^T\boldsymbol{\Psi}_\mathrm{y} + \mathbf{y}_\mathrm{k}\boldsymbol{\Phi}_\mathrm{y}$$

since the increments for the control law between $k + n_\mathrm{u}$ *and* $k + n_\mathrm{f}$ *are zero* $\Delta\mathbf{x}_\mathrm{f} = 0$, *then*

$$\hat{\mathbf{y}}_\mathrm{f} = (\Delta\mathbf{x}_\mathrm{p}\underbrace{\mathbf{Z}_\mathrm{p}\mathbf{B}\mathbf{Q}^T}_{\mathbf{S}_\mathrm{p}} + \Delta\mathbf{t}_\mathrm{dof}\mathbf{B}\mathbf{Q}^T)\boldsymbol{\Psi}_\mathrm{y} + \mathbf{y}_\mathrm{k}\boldsymbol{\Phi}_\mathrm{y}$$

Operating and substituting in from Equation (5.16)

$$\hat{\mathbf{y}}_\mathrm{f} = \mathbf{y}_\mathrm{k}\boldsymbol{\Phi}_\mathrm{y} + [\Delta\mathbf{u}_\mathrm{p} \quad \Delta\mathbf{y}_\mathrm{p}] \begin{bmatrix} \mathbf{S}_{\mathrm{p_u}} \\ \mathbf{S}_{\mathrm{p_y}} \end{bmatrix} \boldsymbol{\Psi}_\mathrm{y} + \Delta\mathbf{t}_\mathrm{d}\mathbf{B}\mathbf{Q}^T\boldsymbol{\Psi}_\mathrm{y}.$$

It can be shown from Equations (5.18) *and* (5.19) *that*

$$\Delta\mathbf{u}_\mathrm{p} = \underbrace{[\mathbf{u}_\mathrm{p} \quad \mathbf{u}_{\mathrm{k}-\mathrm{n_b}}]}_{\triangleq\mathbf{u}^*_\mathrm{p}} \underbrace{\begin{bmatrix} \mathbf{I}_{\mathrm{n_i}} & & \mathbf{0} \\ -\mathbf{I}_{\mathrm{n_i}} & \ddots & \\ & \ddots & \mathbf{I}_{\mathrm{n_i}} \\ \mathbf{0} & & -\mathbf{I}_{\mathrm{n_i}} \end{bmatrix}}_{\triangleq\boldsymbol{\Upsilon}_{\mathrm{u_p}}}$$

$$\Delta\mathbf{y}_\mathrm{p} = \underbrace{[\mathbf{y}_\mathrm{p} \quad \mathbf{y}_{\mathrm{k}-\mathrm{n_a}-1}]}_{\triangleq\mathbf{y}^*_\mathrm{p}} \underbrace{\begin{bmatrix} \mathbf{I}_{\mathrm{n_o}} & & \mathbf{0} \\ -\mathbf{I}_{\mathrm{n_o}} & \ddots & \\ & \ddots & \mathbf{I}_{\mathrm{n_o}} \\ \mathbf{0} & & -\mathbf{I}_{\mathrm{n_o}} \end{bmatrix}}_{\triangleq\boldsymbol{\Upsilon}_{\mathrm{y_p}}}$$

thus

$$\hat{\mathbf{y}}_\mathrm{f} = \mathbf{y}_\mathrm{k}\boldsymbol{\Phi}_\mathrm{y} + \mathbf{u}^*_\mathrm{p}\boldsymbol{\Upsilon}_{\mathrm{u_p}}\mathbf{S}_{\mathrm{p_u}}\boldsymbol{\Psi}_\mathrm{y} + \mathbf{y}^*_\mathrm{p}\boldsymbol{\Upsilon}_{\mathrm{y_p}}\mathbf{S}_{\mathrm{p_y}}\boldsymbol{\Psi}_\mathrm{y} + \Delta\mathbf{t}_\mathrm{d}\mathbf{B}\mathbf{Q}^T\boldsymbol{\Psi}_\mathrm{y}$$

Reorganizing terms

$$\hat{\mathbf{y}}_{\mathrm{f}} = \underbrace{[\mathbf{u}_{\mathrm{p}}^{*} \quad \mathbf{y}_{\mathrm{p}}^{*}]}_{\triangleq \mathbf{x}_{\mathrm{p}}^{*}} \underbrace{\begin{bmatrix} \boldsymbol{\Upsilon}_{\mathrm{u_p}} \mathbf{S}_{\mathrm{p_u}} \boldsymbol{\Psi}_{\mathrm{y}} \\ \begin{bmatrix} \mathbf{I}_{\mathrm{n_o}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\Phi}_{\mathrm{y}} + \boldsymbol{\Upsilon}_{\mathrm{y_p}} \mathbf{S}_{\mathrm{p_y}} \boldsymbol{\Psi}_{\mathrm{y}} \end{bmatrix}}_{\triangleq \bar{\mathbf{S}}_{\mathrm{p}}} + \Delta \mathbf{t}_{\mathrm{d}} \underbrace{\mathbf{B}\mathbf{Q}^{T} \boldsymbol{\Psi}_{\mathrm{y}}}_{\triangleq \bar{\mathbf{S}}_{\mathrm{d}}}. \quad \square \quad (9.30)$$

Proposition 9.4.2 *The dependence of* $\mathbf{x}_{\mathrm{dof}}$ *on* $\Delta \mathbf{t}_{\mathrm{d}}$ *is given as*

$$\mathbf{x}_{\mathrm{dof}} = \Delta \mathbf{t}_{\mathrm{d}} \mathbf{M}_{\mathrm{dof}} \boldsymbol{\Psi}_{\mathrm{u}} + \mathbf{u}_{\mathrm{k-1}} \boldsymbol{\Phi}_{\mathrm{u}}$$

Proof *Multiplying by the* Δ *operator in both sides of proposition 9.2.1*

$$\Delta \mathbf{x}_{\mathrm{dof}} = \Delta \mathbf{t}_{\mathrm{d}} \mathbf{M}_{\mathrm{dof}}. \quad (9.31)$$

It can be shown that

$$\mathbf{x}_{\mathrm{dof}} = \Delta \mathbf{x}_{\mathrm{dof}} \underbrace{\begin{bmatrix} \mathbf{I}_{\mathrm{n_i}} & & \mathbf{0} \\ \vdots & \ddots & \\ \mathbf{I}_{\mathrm{n_i}} & \dots & \mathbf{I}_{\mathrm{n_i}} \end{bmatrix}}_{\triangleq \boldsymbol{\Psi}_{\mathrm{u}}} + \mathbf{u}_{\mathrm{k-1}} \underbrace{[\mathbf{I}_{\mathrm{n_i}} \quad \dots \quad \mathbf{I}_{\mathrm{n_i}}]}_{\triangleq \boldsymbol{\Phi}_{\mathrm{u}}}$$

thus

$$\mathbf{x}_{\mathrm{dof}} = \underbrace{\Delta \mathbf{t}_{\mathrm{d}} \mathbf{M}_{\mathrm{dof}}}_{\Delta \mathbf{x}_{\mathrm{dof}}} \boldsymbol{\Psi}_{\mathrm{u}} + \mathbf{u}_{\mathrm{k-1}} \boldsymbol{\Phi}_{\mathrm{u}}. \quad \square$$

Proposition 9.4.3 *The value of* $\Delta\mathbf{t}_\mathrm{d}$ *that minimizes* $J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d})$ *is*

$$\Delta\mathbf{t}_\mathrm{d} = -\mathbf{f}^T\mathbf{H}^{-1}$$

Proof *Substituting* $\hat{\mathbf{y}}_\mathrm{f}$ *and* $\Delta\mathbf{x}_\mathrm{dof}$ *from proposition 9.4.1 and Equation* (9.31) *in Equation* (9.29) *yields:*

$$J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d}) = ||[\mathbf{r}_\mathrm{f} - \underbrace{(\mathbf{x}_\mathrm{p}^*\bar{\mathbf{S}}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\bar{\mathbf{S}}_\mathrm{d})}_{\hat{\mathbf{y}}_\mathrm{f}}]\mathbf{W}_\mathrm{y}||_F^2 + \lambda_u||\underbrace{\Delta\mathbf{t}_\mathrm{d}\mathbf{M}_\mathrm{dof}}_{\Delta\mathbf{x}_\mathrm{dof}}\mathbf{W}_\mathrm{u}||_F^2$$

which operating yields

$$J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d}) = \Delta\mathbf{t}_\mathrm{d}\mathbf{H}\Delta\mathbf{t}_\mathrm{d}^T + 2\mathbf{f}^T\Delta\mathbf{t}_\mathrm{d}^T + C$$

where

$$\begin{aligned}
\mathbf{H} &= \bar{\mathbf{S}}_\mathrm{d}\mathbf{W}_\mathrm{y}\mathbf{W}_\mathrm{y}^T\bar{\mathbf{S}}_\mathrm{d}^T + \lambda_u\mathbf{M}_\mathrm{dof}\mathbf{W}_\mathrm{u}\mathbf{W}_\mathrm{u}^T\mathbf{M}_\mathrm{dof}^T && (9.32)\\
\mathbf{f}^T &= (\mathbf{x}_\mathrm{p}^*\bar{\mathbf{S}}_\mathrm{p} - \mathbf{r}_\mathrm{f})\mathbf{W}_\mathrm{y}\mathbf{W}_\mathrm{y}^T\bar{\mathbf{S}}_\mathrm{d}^T && (9.33)\\
C &= ||(\mathbf{r}_\mathrm{f} - \mathbf{x}_\mathrm{p}^*\bar{\mathbf{S}}_\mathrm{p})\mathbf{W}_\mathrm{y}||_F^2
\end{aligned}$$

The minimum of $J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d})$ *is attained by equating its first derivative to zero:(Note the Hessian matrix* $\mathbf{H}$ *is symmetric, then* $\mathbf{H} = \mathbf{H}^T$*)*

$$\begin{aligned}
\frac{\partial J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d})}{\partial\Delta\mathbf{t}_\mathrm{d}} &= 0\\
(\mathbf{H} + \mathbf{H}^T)\Delta\mathbf{t}_\mathrm{d}^T + 2\mathbf{f} &= 0\\
2\mathbf{H}\Delta\mathbf{t}_\mathrm{d}^T + 2\mathbf{f} &= 0\\
\mathbf{H}\Delta\mathbf{t}_\mathrm{d}^T + \mathbf{f} &= 0\\
\Delta\mathbf{t}_\mathrm{d}\mathbf{H} + \mathbf{f}^T &= 0\\
\Delta\mathbf{t}_\mathrm{d}\mathbf{H} &= -\mathbf{f}^T\\
\Delta\mathbf{t}_\mathrm{d} &= -\mathbf{f}^T\mathbf{H}^{-1}.\square && (9.34)
\end{aligned}$$

9.5. Constraint handling

It is straightforward to add inequality constraints to the minimization problem in Equation (9.29).

$$\underset{\Delta \mathbf{t}_{\mathrm{d}}}{min} \; J(\Delta \mathbf{t}_{\mathrm{d}}) \quad s.t. \quad \mathbf{A}\Delta \mathbf{t}_{\mathrm{d}}^{T} \leq \mathbf{b} \tag{9.35}$$

$\mathbf{A}$ and $\mathbf{b}$ define constraints. Matrices $\mathbf{A}$ and $\mathbf{b}$ for constraints in the MVs rate are defined in Section 9.5.1, and for constraints in the MVs in Section 9.5.2.

The problem in Equation (9.35) cannot be solved analytically due to the inequality constraints. The common approach for this problems is to use QP (Quadratic programming). Among the many solvers available to solve the QP, qpas, revised 11 August 2009, from Adrian Wills, University of Newcastle, is the one used in this thesis.

9.5.1. MVs rate

Take upper and lower limits on $\Delta \mathbf{x}_{\mathrm{dof}}^{T}$ to be $\overline{\Delta \mathbf{x}_{\mathrm{dof}}}$, and $\underline{\Delta \mathbf{x}_{\mathrm{dof}}}$:

$$\underbrace{\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}}_{\triangleq \mathbf{H}_{\Delta}} \Delta \mathbf{x}_{\mathrm{dof}}^{T} \leq \underbrace{\begin{bmatrix} \overline{\Delta \mathbf{x}_{\mathrm{dof}}} \\ -\underline{\Delta \mathbf{x}_{\mathrm{dof}}} \end{bmatrix}}_{\triangleq \mathbf{k}_{\Delta}}$$

from Equation (9.31)

$$\underbrace{\mathbf{H}_{\Delta}\mathbf{M}_{\mathrm{dof}}^{T}}_{\triangleq \mathbf{A}_{\Delta}} \Delta \mathbf{t}_{\mathrm{d}}^{T} \leq \underbrace{\mathbf{k}_{\Delta}}_{\triangleq \mathbf{b}_{\Delta}}.$$

9.5.2. MVs

Take upper and lower limits on $\mathbf{x}_{\text{dof}}^T$ to be $\overline{\mathbf{x}_{\text{dof}}}$, and $\underline{\mathbf{x}_{\text{dof}}}$:

$$\underbrace{\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}}_{\triangleq \mathbf{H}_{\text{u}}} \mathbf{x}_{\text{dof}}^T \leq \underbrace{\begin{bmatrix} \overline{\mathbf{x}_{\text{dof}}} \\ -\underline{\mathbf{x}_{\text{dof}}} \end{bmatrix}}_{\triangleq \mathbf{k}_{\text{u}}}$$

from proposition 9.4.2

$$\mathbf{H}_{\text{u}}(\mathbf{\Psi}_{\text{u}}^T \mathbf{M}_{\text{dof}}^T \Delta \mathbf{t}_{\text{d}}^T + \mathbf{\Phi}_{\text{u}}^T \mathbf{u}_{\text{k}-1}^T) \leq \mathbf{k}_{\text{u}}$$

thus,

$$\underbrace{\mathbf{H}_{\text{u}} \mathbf{\Psi}_{\text{u}}^T \mathbf{M}_{\text{dof}}^T}_{\triangleq \mathbf{A}_{\text{u}}} \Delta \mathbf{t}_{\text{d}}^T \leq \underbrace{\mathbf{k}_{\text{u}} - \mathbf{H}_{\text{u}} \mathbf{\Phi}_{\text{u}}^T \mathbf{u}_{\text{k}-1}^T}_{\triangleq \mathbf{b}_{\text{u}}}.$$

9.6. Ensure validity of predictions

Assuming there is model-process mismatch, a model should be used in the region in which it has been identified. Using a model outside such region would imply extrapolation which is a bad practice if good predictions are sought. In this section firstly two indicators on how close the use of the model is to where it has been identified are defined, and secondly such indicators are taken into account in the controller by means of two different methodologies.

9.6.1. Validity indicators for predictions

Validity indicators for predictions can be defined in terms of: scores $\mathbf{t}$, and residuals $\mathbf{e}$. The former yields the Hotelling's T^2 index:

$$J_{\text{t}} = \mathbf{t} \mathbf{S}_{\text{a}}^{-1} \mathbf{t}^T \tag{9.36}$$

where $\mathbf{S^2}_{\mathrm{a}}$ is a diagonal matrix such that element i is the variance of the score $\mathbf{t}_{\mathrm{i}}$ in the identification data set. Extrapolation happens if J_{t} is outside the region spanned by the observations in the identification data set. The latter can be expressed

$$J_{\mathrm{e}} = ||\mathbf{e}||_2^2 = \mathbf{e}\mathbf{e}^T \tag{9.37}$$

where $\mathbf{e}$ are the residuals in the $\mathbf{X}$ space at a given instant (from Equation (4.2)):

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \quad \Rightarrow \quad \mathbf{e} = \mathbf{x} - \mathbf{t}\mathbf{P}^T.$$

J_{e} being outside the region spanned by the observations in the identification data set implies the error of projections to the latent variable space is larger than that obtained in the identification stage, then predictions may not be accurate. These two indices are often also used in control loop monitoring to detect controller abnormal operation [AlGhazzawi 09].

Figures 9.4 to 9.6 ease interpretation of J_{t} and J_{e} by means of a two-dimensional example for the input space and one latent variable. The two-dimensional input space is projected onto t_1, and this one-dimensional space –t_1– is used in the PLS model to predict the output. Lets say $x_1 := y_{\mathrm{k}-1}$, and $x_2 := u_{\mathrm{k}}$, then x_1 is known provided it belongs to the past, and the controller is to decide the control action x_2. The identification points represented using plus symbols in Figure 9.4 are used in the identification stage to define the orientation of the vector[5] that contains t_1.

The region in blue vertical lines in Figure 9.5 contains the points of the input space whose projections onto the latent variable space –t_1– lay in the region spanned by the observations in the identification data set. J_{t} indicates how much a point belongs to this blue region; large values of J_{t} compared to those obtained for the identification data set, indicate the point is outside the blue region. For a given $y_{\mathrm{k}-1}$, if the controller decides u_{k} such that the star point is obtained, J_{t} will be larger than the maximum value of J_{t} evaluated for the identification data set provided the star is outside the blue region. In other

[5]The vector of scores $\mathbf{p}_1$ define the orientation of the the first latent variable in the input space.

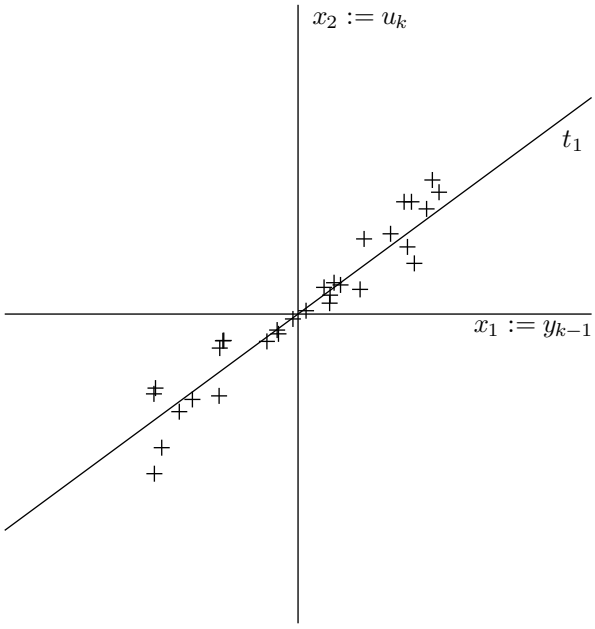


Figure 9.4: Ensure validity of the model: Identification data set

words, if the controller sets x_2 such that the point lays outside the blue region, the controller is using the model in extrapolation and predictions may not be accurate. Extrapolation can be avoided by considering J_t when deciding the control action, i.e. accounting for J_t in the controller.

Lets assume the controller accounts for J_t and the circle point in Figure 9.5 is obtained. The circle point lays in the blue region, however, the projection onto the latent variable presents a larger error than all the identification data points. Figure 9.6 adds a region in red horizontal lines which encloses the points of the input space whose errors of projection onto the latent variable space lay in the region spanned by the observations in the identification data

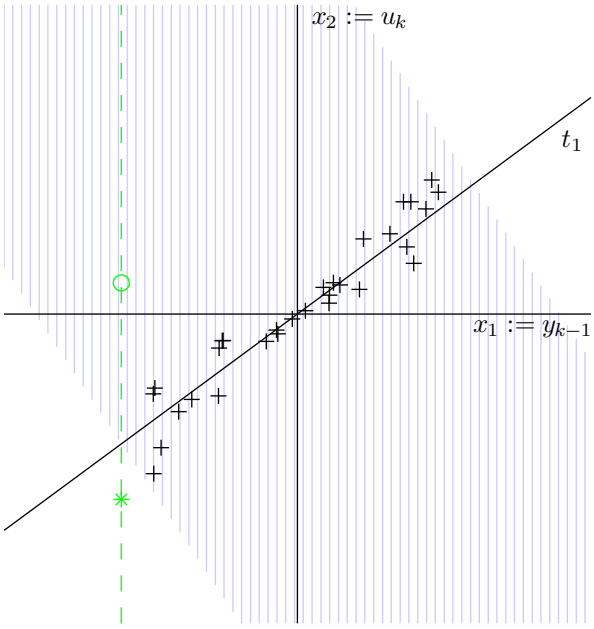


Figure 9.5: Ensure validity of the model: J_{t}

set. Index J_{e} indicates how much a point belongs to this red region; large values of J_{e} compared to those obtained for identification indicate the point is outside the red region. If the circle is outside the red region, then predictions are not reliable. If the controller accounts for J_{t} and J_{e} the triangle point may be obtained; provided the triangle point is inside both regions, the predictor in the controller has been used in the region in which it has been identified.

These indices can be normalized to the identification data set:

$$\bar{J}_{\mathrm{t}} = \frac{1}{J_{\mathrm{t}max}} J_{\mathrm{t}} \quad ; \quad \bar{J}_{\mathrm{e}} = \frac{1}{J_{\mathrm{e}max}} J_{\mathrm{e}} \tag{9.38}$$

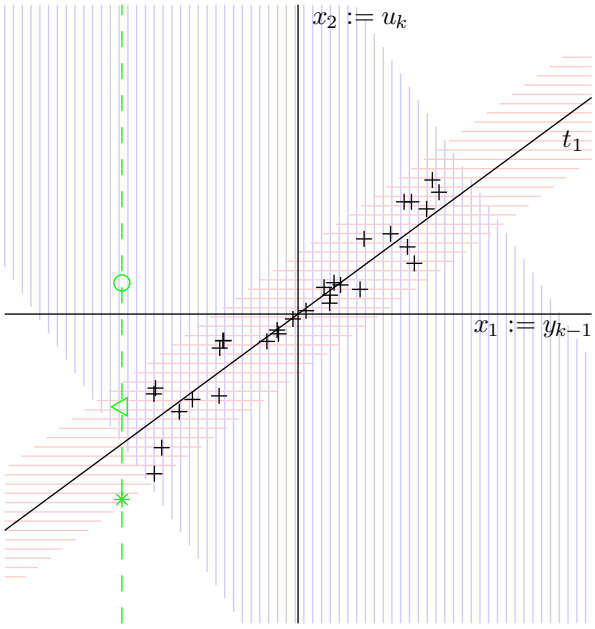


Figure 9.6: Ensure validity of the model: J_t and J_e

where: $J_{\mathrm{t}max}$, is the value of J_t that includes 95% of the observations in the identification data set; and J_{emax} is defined accordingly. Given this normalization, a value of $\bar{J}_\mathrm{t}$ or $\bar{J}_\mathrm{e}$ below 1 implies the controller uses the model in the region in which it has been identified. Note $\bar{J}_\mathrm{t} = 1$ defines the boundaries of the blue region in Figure 9.6 and $\bar{J}_\mathrm{e} = 1$ defines the boundaries of the red region.

Two approaches to account for $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ in the minimization of the controller are detailed in the following subsections. In the first approach $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ are weighted in the quadratic cost function of the controller; the weighting factor determines how tightly the solution is to be constrained to the region

of the identification scores and residuals. In the second approach $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ appear as constraints in the minimization problem.

Both approaches need $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ be expressed in terms of $\Delta\mathbf{t}_{\mathrm{d}}$. The expression $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}})$ can be derived from its definition in Equations (9.36) and (9.38), and the expression for $\mathbf{t}$ in terms of $\Delta\mathbf{t}_{\mathrm{d}}$ provided in proposition 9.6.1

$$\begin{aligned}
\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) &= \frac{1}{J_{\mathrm{t}max}}\mathbf{t}\mathbf{S}_{\mathrm{a}}^{\mathbf{2}\,-1}\mathbf{t}^T \\
&= \frac{1}{J_{\mathrm{t}max}}\underbrace{(\mathbf{x}_{\mathrm{p}}^*\mathbf{N}_{\mathrm{p}}\mathbf{Z} + \Delta\mathbf{t}_{\mathrm{d}}\mathbf{N}_{\mathrm{d}}\mathbf{Z})}_{\mathbf{t}}\mathbf{S}_{\mathrm{a}}^{\mathbf{2}\,-1}\underbrace{(\mathbf{x}_{\mathrm{p}}^*\mathbf{N}_{\mathrm{p}}\mathbf{Z} + \Delta\mathbf{t}_{\mathrm{d}}\mathbf{N}_{\mathrm{d}}\mathbf{Z})}_{\mathbf{t}}{}^T \\
&= \Delta\mathbf{t}_{\mathrm{d}}\mathbf{H}_{\mathrm{t}}\Delta\mathbf{t}_{\mathrm{d}}^T + 2\mathbf{f}_{\mathrm{t}}^T\Delta\mathbf{t}_{\mathrm{d}}^T + \mathbf{x}_{\mathrm{p}}^*\frac{\mathbf{N}_{\mathrm{p}}\mathbf{Z}\mathbf{S}_{\mathbf{a}}^{\mathbf{2}\,-1}\mathbf{Z}^T\mathbf{N}_{\mathrm{p}}^T}{J_{\mathrm{t}max}}\mathbf{x}_{\mathrm{p}}^{*T} \qquad (9.39)
\end{aligned}$$

where

$$\mathbf{H}_{\mathrm{t}} \triangleq \frac{\mathbf{N}_{\mathrm{d}}\mathbf{Z}\mathbf{S}_{\mathbf{a}}^{\mathbf{2}\,-1}\mathbf{Z}^T\mathbf{N}_{\mathrm{d}}^T}{J_{\mathrm{t}max}} \qquad (9.40)$$

$$\mathbf{f}_{\mathrm{t}}^T \triangleq \frac{\mathbf{x}_{\mathrm{p}}^*\mathbf{N}_{\mathrm{p}}\mathbf{Z}\mathbf{S}_{\mathbf{a}}^{\mathbf{2}\,-1}\mathbf{Z}^T\mathbf{N}_{\mathrm{d}}^T}{J_{\mathrm{t}max}} \qquad (9.41)$$

The expression $\bar{J}_{\mathrm{e}}(\Delta\mathbf{t}_{\mathrm{d}})$ can be derived from its definition in Equations (9.37) and (9.38), and the expression for $\mathbf{e}$ in terms of $\Delta\mathbf{t}_{\mathrm{d}}$ provided in proposition 9.6.2

$$\begin{aligned}
\bar{J}_{\mathrm{e}}(\Delta\mathbf{t}_{\mathrm{d}}) &= \frac{1}{J_{\mathrm{e}max}}\mathbf{e}\mathbf{e}^T \\
&= \frac{1}{J_{\mathrm{e}max}}\underbrace{(\mathbf{x}_{\mathrm{p}}^*\mathbf{E}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d}}\mathbf{E}_{\mathrm{d}})}_{\mathbf{e}}\underbrace{(\mathbf{x}_{\mathrm{p}}^*\mathbf{E}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d}}\mathbf{E}_{\mathrm{d}})}_{\mathbf{e}}{}^T \\
&= \Delta\mathbf{t}_{\mathrm{d}}\mathbf{H}_{\mathrm{e}}\Delta\mathbf{t}_{\mathrm{d}}^T + 2\mathbf{f}_{\mathrm{e}}^T\Delta\mathbf{t}_{\mathrm{d}}^T + \mathbf{x}_{\mathrm{p}}^*\frac{\mathbf{E}_{\mathrm{p}}\mathbf{E}_{\mathrm{p}}^T}{J_{\mathrm{e}max}}\mathbf{x}_{\mathrm{p}}^{*T} \qquad (9.42)
\end{aligned}$$

where

$$\mathbf{H}_{\mathrm{e}} \triangleq \frac{\mathbf{E}_{\mathrm{d}}\mathbf{E}_{\mathrm{d}}^T}{J_{\mathrm{e}max}} \qquad (9.43)$$

$$\mathbf{f}_{\mathrm{e}}^T \triangleq \frac{\mathbf{x}_{\mathrm{p}}^*\mathbf{E}_{\mathrm{p}}\mathbf{E}_{\mathrm{d}}^T}{J_{\mathrm{e}max}} \qquad (9.44)$$

Proposition 9.6.1 $\mathbf{t}$ *can be expressed in terms of* $\Delta\mathbf{t}_\mathrm{d}$ *as*

$$\mathbf{t} = \mathbf{x}_\mathrm{p}^*\mathbf{N}_\mathrm{p}\mathbf{Z} + \Delta\mathbf{t}_\mathrm{d}\mathbf{N}_\mathrm{d}\mathbf{Z}$$

Proof *From Equation* (4.4)

$$\mathbf{t} = \mathbf{x}\mathbf{Z} \tag{9.45}$$

From Equation (5.16)

$$\mathbf{x} = [\mathbf{x}_\mathrm{p}\ \mathbf{x}_\mathrm{f}\ \mathbf{x}_\mathrm{dof}]$$

From equation (9.6)*, and taking matrices* $\mathbf{0}$ *and* $\mathbf{I}$ *of appropriate dimensions,*

$$\mathbf{x} = \mathbf{x}_\mathrm{p}[\mathbf{I}\ \mathbf{0}\ \mathbf{0}] + \mathbf{x}_\mathrm{dof}[\mathbf{0}\ \Gamma\ \mathbf{I}]$$

Substituting in $\mathbf{x}_\mathrm{dof}$ *from proposition 9.4.2, and considering the definition of* $\mathbf{x}_\mathrm{p}^*$ *in proposition 9.4.1:*

$$\begin{aligned}
\mathbf{x} &= \mathbf{x}_\mathrm{p}[\mathbf{I}\ \mathbf{0}\ \mathbf{0}] + \underbrace{(\Delta\mathbf{t}_\mathrm{d}\mathbf{M}_\mathrm{dof}\boldsymbol{\Psi}_\mathrm{u} + \mathbf{u}_{\mathrm{k}-1}\boldsymbol{\Phi}_\mathrm{u})}_{\mathbf{x}_\mathrm{dof}}[\mathbf{0}\ \Gamma\ \mathbf{I}] \\
&= \mathbf{x}_\mathrm{p}[\mathbf{I}\ \mathbf{0}\ \mathbf{0}] + \mathbf{u}_{\mathrm{k}-1}\boldsymbol{\Phi}_\mathrm{u}[\mathbf{0}\ \Gamma\ \mathbf{I}] + \Delta\mathbf{t}_\mathrm{d}\mathbf{M}_\mathrm{dof}\boldsymbol{\Psi}_\mathrm{u}[\mathbf{0}\ \Gamma\ \mathbf{I}] \\
&= \mathbf{x}_\mathrm{p}^*\mathbf{N}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\mathbf{N}_\mathrm{d}
\end{aligned} \tag{9.46}$$

where

$$\mathbf{N}_\mathrm{p} \triangleq \left[\left[\left[\begin{array}{c} \mathbf{I}_{(\mathrm{n_b}-1)\mathrm{n_i}} \\ \mathbf{0}_{\mathrm{n_i}\times(\mathrm{n_b}-1)\mathrm{n_i}} \end{array}\right] \quad \mathbf{0} \quad \left[\begin{array}{c} \mathbf{I}_{\mathrm{n_a}\mathrm{n_o}} \\ \mathbf{0}_{\mathrm{n_o}\times\mathrm{n_a}\mathrm{n_o}} \end{array}\right]\right] \mathbf{0}\ \mathbf{0}\right] + \left[\begin{array}{c} \boldsymbol{\Phi}_\mathrm{u}[\mathbf{0}\ \Gamma\ \mathbf{I}] \\ \mathbf{0} \end{array}\right]$$

$$\mathbf{N}_\mathrm{d} \triangleq \mathbf{M}_\mathrm{dof}\boldsymbol{\Psi}_\mathrm{u}[\mathbf{0}\ \Gamma\ \mathbf{I}]$$

Substituting $\mathbf{x}$ *in Equation* (9.45)*:*

$$\mathbf{t} = \underbrace{\mathbf{x}_\mathrm{p}^*\mathbf{N}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\mathbf{N}_\mathrm{d}}_{\mathbf{x}}\mathbf{Z} = \mathbf{x}_\mathrm{p}^*\mathbf{N}_\mathrm{p}\mathbf{Z} + \Delta\mathbf{t}_\mathrm{d}\mathbf{N}_\mathrm{d}\mathbf{Z}. \quad \square$$

Proposition 9.6.2 $\mathbf{e}$ *can be expressed in terms of* $\Delta\mathbf{t}_\mathrm{d}$ *as*

$$\mathbf{e} = \mathbf{x}_\mathrm{p}^* \mathbf{E}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\mathbf{E}_\mathrm{d}$$

Proof *From Equation* (4.2)

$$\mathbf{x} = \mathbf{t}\mathbf{P}^T + \mathbf{e} \quad \Rightarrow \quad \mathbf{e} = \mathbf{x} - \mathbf{t}\mathbf{P}^T$$

Substituting $\mathbf{t}$ *in from Equation* (4.4)

$$\mathbf{e} = \mathbf{x} - \underbrace{\mathbf{x}\mathbf{Z}}_{\mathbf{t}}\mathbf{P}^T = \mathbf{x}(\mathbf{I} - \mathbf{Z}\mathbf{P}^T)$$

Substituting in $\mathbf{x}$ *from Equation* (9.46)

$$\mathbf{e} = \mathbf{x}_\mathrm{p}^* \underbrace{\mathbf{N}_\mathrm{p}(\mathbf{I} - \mathbf{Z}\mathbf{P}^T)}_{\triangleq \mathbf{E}_\mathrm{p}} + \Delta\mathbf{t}_\mathrm{d} \underbrace{\mathbf{N}_\mathrm{d}(\mathbf{I} - \mathbf{Z}\mathbf{P}^T)}_{\triangleq \mathbf{E}_\mathrm{d}}$$

thus

$$\mathbf{e} = \mathbf{x}_\mathrm{p}^* \mathbf{E}_\mathrm{p} + \Delta\mathbf{t}_\mathrm{d}\mathbf{E}_\mathrm{d}. \quad \square$$

9.6.2. Validity indicators for predictions neglecting past data

As defined in the previous subsection, $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ indicate if a model has been used in the region spanned by the identification data to perform a prediction. Such indicators may be used in the controller to make decisions that avoid

using the model outside the identification region. However, $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ not only depend on the degrees of freedom of the controller, but also on past measured data.

Past measured data may not lay in the region spanned by the identification data set probably because of an error in measuring one of the CVs of the process at a given instant, or simply because the process is steady at a point not included in the identification data set. If past measured data lays outside the region spanned by the identification data set: hard constraints on $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ lead to infeasibility; and soft constraints on $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ alter the decision trying to force the process to stay inside the region defined in the identification data set. However, since the decision space of the controller is the future control sequence, and cannot change the past, these constraints may cause an undesirable effect and the resulting control may be biased. One mean to cope with this fact is to neglect past data and define validity indicators only in terms of the actual degrees of freedom of the controller.

The following two indicators neglect past data and can be defined accordingly to $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$:

$$\check{J}_{\mathrm{t}} = \frac{1}{\check{J}_{\mathrm{t}max}} \check{\mathbf{t}} \check{\mathbf{S}}_a^{2^{-1}} \check{\mathbf{t}}^T \tag{9.47}$$

$$\check{J}_{\mathrm{e}} = \frac{1}{\check{J}_{\mathrm{e}max}} \check{\mathbf{e}} \check{\mathbf{e}}^T \tag{9.48}$$

where: $\check{\mathbf{t}}$ represents the projection of the input space to the latent variable space neglecting past values; $\check{\mathbf{e}}$ represents the error of projecting the input space to the latent variable space neglecting past values; $\check{\mathbf{S}}_a^2$ is a diagonal matrix such that element i is the variance of the score $\check{\mathbf{t}}_{\mathrm{i}}$ in the identification data set neglecting past values; $\check{J}_{\mathrm{t}max}$ is the value of the expression $\check{\mathbf{t}} \check{\mathbf{S}}_a^{2^{-1}} \check{\mathbf{t}}^T$ that includes 95% of the observations in the identification data set neglecting past values; and $\check{J}_{\mathrm{e}max}$ is defined accordingly for the expression $\check{\mathbf{e}} \check{\mathbf{e}}^T$.

To include these indices in the controller, both indices need be expressed as a function of $\Delta \mathbf{t}_{\mathrm{d}}$. $\check{J}_{\mathrm{t}}(\Delta \mathbf{t}_{\mathrm{d}})$ can be derived from its definition in Equation

(9.47), and the expression for $\check{\mathbf{t}}$ in terms of $\Delta\mathbf{t}_\mathrm{d}$ provided in proposition 9.6.3

$$\begin{aligned}
\check{J}_\mathrm{t}(\Delta\mathbf{t}_\mathrm{d}) &= \Delta\mathbf{t}_\mathrm{d}\check{\mathbf{H}}_\mathrm{t}\Delta\mathbf{t}_\mathrm{d}^T + 2\check{\mathbf{f}}_\mathrm{t}^T\Delta\mathbf{t}_\mathrm{d}^T + \mathbf{x}_\mathrm{p}^*\frac{\check{\mathbf{N}}_\mathrm{p}\mathbf{Z}\check{\mathbf{S}}_a^{2^{-1}}\mathbf{Z}^T\check{\mathbf{N}}_\mathrm{p}^T}{\check{J}_{\mathrm{t}max}}\mathbf{x}_\mathrm{p}^{*^T} \qquad (9.49)\\
\check{\mathbf{H}}_\mathrm{t} &\triangleq \frac{\check{\mathbf{N}}_\mathrm{d}\mathbf{Z}\check{\mathbf{S}}_a^{2^{-1}}\mathbf{Z}^T\check{\mathbf{N}}_\mathrm{d}^T}{\check{J}_{\mathrm{t}max}}\\
\check{\mathbf{f}}_\mathrm{t}^T &\triangleq \frac{\mathbf{x}_\mathrm{p}^*\check{\mathbf{N}}_\mathrm{p}\mathbf{Z}\check{\mathbf{S}}_a^{2^{-1}}\mathbf{Z}^T\check{\mathbf{N}}_\mathrm{d}^T}{\check{J}_{\mathrm{t}max}}
\end{aligned}$$

$\check{J}_\mathrm{e}(\Delta\mathbf{t}_\mathrm{d})$ can be derived from its definition in Equation (9.48), and the expression for $\check{\mathbf{e}}$ in terms of $\Delta\mathbf{t}_\mathrm{d}$ provided in proposition 9.6.4

$$\begin{aligned}
\check{J}_\mathrm{e}(\Delta\mathbf{t}_\mathrm{d}) &= \Delta\mathbf{t}_\mathrm{d}\check{\mathbf{H}}_\mathrm{e}\Delta\mathbf{t}_\mathrm{d}^T + 2\check{\mathbf{f}}_\mathrm{e}^T\Delta\mathbf{t}_\mathrm{d}^T + \mathbf{x}_\mathrm{p}^*\frac{\check{\mathbf{E}}_\mathrm{p}\check{\mathbf{E}}_\mathrm{p}^T}{\check{J}_{\mathrm{e}max}}\mathbf{x}_\mathrm{p}^{*T} \qquad (9.50)\\
\check{\mathbf{H}}_\mathrm{e} &\triangleq \frac{\check{\mathbf{E}}_\mathrm{d}\check{\mathbf{E}}_\mathrm{d}^T}{\check{J}_{\mathrm{e}max}}\\
\check{\mathbf{f}}_\mathrm{e}^T &\triangleq \frac{\mathbf{x}_\mathrm{p}^*\check{\mathbf{E}}_\mathrm{p}\check{\mathbf{E}}_\mathrm{d}^T}{\check{J}_{\mathrm{e}max}}
\end{aligned}$$

Proposition 9.6.3 $\check{\mathbf{t}} = \mathbf{x}_\mathrm{p}^*\check{\mathbf{N}}_\mathrm{p}\mathbf{Z} + \Delta\mathbf{t}_\mathrm{d}\check{\mathbf{N}}_\mathrm{d}\mathbf{Z}$

Proof *From Equations* (4.4) *and* (5.16)

$$\mathbf{t} = [\mathbf{x}_\mathrm{p} \quad \mathbf{x}_\mathrm{f} \quad \mathbf{x}_\mathrm{dof}]\mathbf{Z}$$

$\check{\mathbf{t}}$ *is defined forcing past data to be zero*

$$\check{\mathbf{t}} \triangleq \underbrace{[\mathbf{0} \quad \mathbf{x}_\mathrm{f} \quad \mathbf{x}_\mathrm{dof}]}_{\triangleq\check{\mathbf{x}}}\mathbf{Z} \qquad (9.51)$$

From equation (9.6)*, and taking matrices* $\mathbf{0}$ *and* $\mathbf{I}$ *of appropriate dimensions,*

$$\check{\mathbf{x}} = \mathbf{x}_\mathrm{dof}[\mathbf{0}\ \Gamma\ \mathbf{I}]$$

From the definition of $\mathbf{x}_{\mathrm{p}}^{*}$ *in proposition 9.4.1, and the expression for* $\mathbf{x}_{\mathrm{dof}}$ *in proposition 9.4.2:*

$$\check{\mathbf{x}} = \mathbf{x}_{\mathrm{p}}^{*}\check{\mathbf{N}}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d}}\check{\mathbf{N}}_{\mathrm{d}} \tag{9.52}$$

where

$$\begin{aligned}\check{\mathbf{N}}_{\mathrm{p}} &\triangleq \begin{bmatrix} \mathbf{\Phi}_{\mathrm{u}}[\mathbf{0}\ \Gamma\ \mathbf{I}] \\ \mathbf{0} \end{bmatrix} \\ \check{\mathbf{N}}_{\mathrm{d}} &\triangleq \mathbf{M}_{\mathrm{dof}}\mathbf{\Psi}_{\mathrm{u}}[\mathbf{0}\ \Gamma\ \mathbf{I}]\end{aligned}$$

thus,

$$\check{\mathbf{t}} = \mathbf{x}_{\mathrm{p}}^{*}\check{\mathbf{N}}_{\mathrm{p}}\mathbf{Z} + \Delta\mathbf{t}_{\mathrm{d}}\check{\mathbf{N}}_{\mathrm{d}}\mathbf{Z}. \quad \square$$

Proposition 9.6.4 $\check{\mathbf{e}} = \mathbf{x}_{\mathrm{p}}^{*}\check{\mathbf{E}}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d}}\check{\mathbf{E}}_{\mathrm{d}}$

Proof *From Equation* (4.2)

$$\check{\mathbf{e}} = \check{\mathbf{x}} - \check{\mathbf{t}}\mathbf{P}^{T}$$

Substituting in from Equation (9.51)

$$\check{\mathbf{e}} = \check{\mathbf{x}}(\mathbf{I} - \mathbf{Z}\mathbf{P}^{T})$$

Substituting in from Equation (9.52)

$$\check{\mathbf{e}} = \mathbf{x}_{\mathrm{p}}^{*}\underbrace{\check{\mathbf{N}}_{\mathrm{p}}(\mathbf{I} - \mathbf{Z}\mathbf{P}^{T})}_{\triangleq\check{\mathbf{E}}_{\mathrm{p}}} + \Delta\mathbf{t}_{\mathrm{d}}\underbrace{\check{\mathbf{N}}_{\mathrm{d}}(\mathbf{I} - \mathbf{Z}\mathbf{P}^{T})}_{\triangleq\check{\mathbf{E}}_{\mathrm{d}}}$$

thus

$$\check{\mathbf{e}} = \mathbf{x}_{\mathrm{p}}^{*}\check{\mathbf{E}}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d}}\check{\mathbf{E}}_{\mathrm{d}}. \quad \square$$

9.6.3. Weight validity indicators in the quadratic cost function

The minimization problem in Equation (9.35) can be reformulated to include the validity indicators by using Lagrange multipliers. The basic idea in Lagrange multipliers is to take the constraints into account by augmenting the objective function with a weighted sum of the constraint functions [Boyd 04].

In this subsection, the quadratic indicators, $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$, are weighted in the quadratic cost function in Equation (9.35). Note $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$ could be used instead if past data is to be neglected.

$$\min_{\Delta \mathbf{t}_{\mathrm{d}}} J_{\mathrm{C}}(\Delta \mathbf{t}_{\mathrm{d}}) + \lambda_t \bar{J}_{\mathrm{t}} + \lambda_e \bar{J}_{\mathrm{e}} \quad s.t. \quad \mathbf{A}\Delta \mathbf{t}_{\mathrm{d}}^T \leq \mathbf{b} \tag{9.53}$$

where $\lambda_t \geq 0$ and $\lambda_e \geq 0$ weight the indices added to the cost function.

Small values of λ_t and λ_e let $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ take values above 1, thus the model is used in extrapolation. Large values of λ_t and λ_e bound the decision space to the identification data set. Hence λ_t and λ_e should be tuned to indirectly tune how much exploration of new areas is allowed. In the example in [Laurí 10b], $\lambda_t = 10$ and $\lambda_e = 1$, provide $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ evaluated in closed loop around 1, then the model is being used in the region in which it has been identified. Another contribution that includes $\bar{J}_{\mathrm{t}}$ to the cost function is [Flores-Cerrillo 04].

Including these two indices (Equations (9.39) and (9.42)) to the optimization problem in Equation (9.35) adds two terms to $\mathbf{H}$ in Equation (9.32) and $\mathbf{f}^T$ in Equation (9.33):

$$\bar{\mathbf{H}} = \mathbf{H} + \lambda_t \mathbf{H}_{\mathrm{t}} + \lambda_e \mathbf{H}_{\mathrm{e}} \tag{9.54}$$

$$\bar{\mathbf{f}}^T = \mathbf{f}^T + \lambda_t \mathbf{f}_{\mathrm{t}}^T + \lambda_e \mathbf{f}_{\mathrm{e}}^T \tag{9.55}$$

where: $\mathbf{H}_{\mathrm{t}}$ and $\mathbf{f}_{\mathrm{t}}$ are defined in Equations 9.40 and 9.41. $\mathbf{H}_{\mathrm{e}}$ and $\mathbf{f}_{\mathrm{e}}$ are defined in Equations 9.43 and 9.44. λ_t and λ_e are tuned off-line, hence the resulting problem is a QP.

Alternatively one can search for the smallest λ_t and λ_e that yield an acceptable value for $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ (both below one if no extrapolation is to be allowed). An on-line search for λ_t and λ_e ensures, for a feasible problem, acceptable values of $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$, and minimizes alterations of the cost function $J(\Delta\mathbf{t}_\mathrm{d})$ to an increase in computational complexity provided the search for λ_t and λ_e implies solving a sequence of QP problems.

Summing up, to consider the validity indicators by weighting them in the cost function: $\mathbf{H}$ in Equation (9.32) is to be replaced by $\bar{\mathbf{H}}$ in Equation (9.54); $\mathbf{f}^T$ in Equation (9.33) is to be replaced by $\bar{\mathbf{f}}^T$ in Equation (9.55); and λ_t and λ_e need be tuned so that acceptable values of $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ are obtained in closed-loop. This approach is inexpensive on-line, but tuning of λ_t and λ_e to force validity indicators remain in a given area may alter considerably the shape of the cost function. An alternative solution is presented in the following subsection which includes validity indicators as quadratic constraints.

9.6.4. Add constraints on validity indicators to the controller

The minimization problem in Equation (9.35) can be augmented with constraints on $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$. Note $\check{J}_\mathrm{t}$ and $\check{J}_\mathrm{e}$ could be used instead if past data is to be neglected.[6]

$$\min_{\Delta\mathbf{t}_\mathrm{d}} J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d}) \quad s.t. \quad \begin{cases} \mathbf{A}\Delta\mathbf{t}_\mathrm{d}^T \leq \mathbf{b} \\ \bar{J}_\mathrm{t} \leq 1 \\ \bar{J}_\mathrm{e} \leq 1 \end{cases} \tag{9.56}$$

Note from Equations (9.39) and (9.42) that $\bar{J}_\mathrm{t}$ and $\bar{J}_\mathrm{e}$ depend quadratically on $\Delta\mathbf{t}_\mathrm{d}$, then the problem in Equation (9.56) is a quadratic constrained quadratic programming problem (QCQP).

[6] At the end of this subsection propositions are provided both for the validity indices used in this subsection, and also for validity indices neglecting past data.

In a QCQP we minimize a convex quadratic function over a feasible region that is the intersection of ellipsoids [Boyd 04]. In a QP however, we minimize a convex quadratic function over a feasible region that is the intersection of hyperplanes. A mean to simplify the QCQP is to transform it into a QP by bounding the ellipsoids by hyperplanes provided quadratic constraints are convex. Such hyperplanes are obtained by linearising the ellipsoids at some points of interest. Given the set of hyperplanes $\mathbf{A}_\mathrm{t}\Delta\mathbf{t}_\mathrm{d}^T \leq \mathbf{b}_\mathrm{t}$ that bounds quadratic constraint $\bar{J}_\mathrm{t}(\Delta\mathbf{t}_\mathrm{d}) \leq 1$, and the set of hyperplanes $\mathbf{A}_\mathrm{e}\Delta\mathbf{t}_\mathrm{d}^T \leq \mathbf{b}_\mathrm{e}$ that bounds quadratic constraint $\bar{J}_\mathrm{e}(\Delta\mathbf{t}_\mathrm{d}) \leq 1$, the QCQP in Equation (9.56) is reformulated:

$$\min_{\Delta\mathbf{t}_\mathrm{d}} J_\mathrm{C}(\Delta\mathbf{t}_\mathrm{d}) \quad s.t. \quad \begin{cases} \mathbf{A}\Delta\mathbf{t}_\mathrm{d}^T \leq \mathbf{b} \\ \begin{bmatrix} \mathbf{A}_\mathrm{t} \\ \mathbf{A}_\mathrm{e} \end{bmatrix} \Delta\mathbf{t}_\mathrm{d}^T \leq \begin{bmatrix} \mathbf{b}_\mathrm{t} \\ \mathbf{b}_\mathrm{e} \end{bmatrix} \end{cases} \tag{9.57}$$

$\mathbf{A}_\mathrm{t}$, $\mathbf{b}_\mathrm{t}$, $\mathbf{A}_\mathrm{e}$, and $\mathbf{b}_\mathrm{e}$ are needed to solve the problem in Equation (9.57), however, they are initially unknown. The problem in Equation (9.57) can be solved by means of the following iterative procedure, where $\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}$ stands for $\Delta\mathbf{t}_\mathrm{d}$ at iteration i.

1. $\mathbf{A}_\mathrm{t}$, $\mathbf{b}_\mathrm{t}$, $\mathbf{A}_\mathrm{e}$, and $\mathbf{b}_\mathrm{e}$ are initialized empty,
2. $\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}$ comes from solving the QP in Equation (9.57)
3. the algorithm finishes if $\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}$ satisfies both quadratic constraints: $\bar{J}_\mathrm{t}(\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}) \leq 1$ and $\bar{J}_\mathrm{e}(\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}) \leq 1$,
4. if $\bar{J}_\mathrm{t}(\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}) \leq 1$ does not hold; $\bar{J}_\mathrm{t}(\Delta\mathbf{t}_\mathrm{d})$ is linearised, and $\mathbf{A}_\mathrm{t}$, $\mathbf{b}_\mathrm{t}$ are augmented with the linearised constraint
5. if $\bar{J}_\mathrm{e}(\Delta\mathbf{t}_{\mathrm{d}_\mathrm{i}}) \leq 1$ does not hold; $\bar{J}_\mathrm{e}(\Delta\mathbf{t}_\mathrm{d})$ is linearised, and $\mathbf{A}_\mathrm{e}$, $\mathbf{b}_\mathrm{e}$ are augmented with the linearised constraint
6. go to step 2

Linearisation of the quadratic constraints on steps 4 and 5 in the above procedure are implemented taking the first order Taylor approximation of the quadratic constraints. Two alternative approaches depending on which point is used for linearisation are considered:[7]

- (I) Linearise $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ at the current solution $\Delta\mathbf{t}_{\mathrm{d_i}}$
- (II) Linearise $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$. $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ is defined such that $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d_{ti}}}) = 1$, and it is aligned with the current solution $\Delta\mathbf{t}_{\mathrm{d_i}}$ and $\Delta\mathbf{t}_{\mathrm{d_t}}$, where $\Delta\mathbf{t}_{\mathrm{d_t}}$ minimizes $\bar{J}_{\mathrm{t}}$. The expression for $\Delta\mathbf{t}_{\mathrm{d_t}}$ is derived in proposition 9.6.5 and for $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ in proposition 9.6.7.

The First order Taylor approximation of the quadratic constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ is derived in proposition 9.6.9

$$\mathbf{A}_{\mathrm{t_i}}\Delta\mathbf{t}_{\mathrm{d}}^{T} \leq \mathbf{B}_{\mathrm{t_i}}.$$

In the sake of clarity and to compare both linearisation approaches, the following two-dimensional example is considered:

$$J_{\mathrm{C}}(\Delta\mathbf{t}_{\mathrm{d}}) = \Delta\mathbf{t}_{\mathrm{d}} \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \Delta\mathbf{t}_{\mathrm{d}}^{T} + 2\,[-0.2 \quad 0]\Delta\mathbf{t}_{\mathrm{d}}^{T}$$

$$\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) = \Delta\mathbf{t}_{\mathrm{d}} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \Delta\mathbf{t}_{\mathrm{d}}^{T} + 2[0.05 \quad 0]\Delta\mathbf{t}_{\mathrm{d}}^{T} + 0.8$$

In Figures 9.7 and 9.8: the contour plot of the cost function $J(\Delta\mathbf{t}_{\mathrm{d}})$ is in grey; the area inside the ellipse satisfies constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$; $\Delta\mathbf{t}_{\mathrm{d_i}}$ is the solution of the QP problem with the current constraints; $\Delta\mathbf{t}_{\mathrm{d_t}}$ is the minimum of $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}})$; and $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ is aligned with $\Delta\mathbf{t}_{\mathrm{d_i}}$ and $\Delta\mathbf{t}_{\mathrm{d_t}}$ and intersects with the boundary of the quadratic constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) = 1$. Constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$

[7] Linearisation of the quadratic constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ is explained, but the same procedure applies to $\bar{J}_{\mathrm{e}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$.

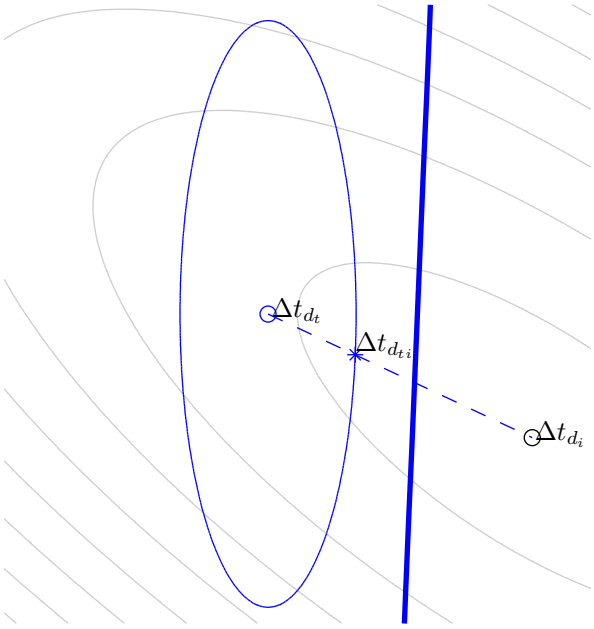


Figure 9.7: Linearise at $\Delta \mathbf{t}_{\mathrm{d_i}}$

is linearised at $\Delta \mathbf{t}_{\mathrm{d_i}}$ and represented in thick line in Figure 9.7, whereas in Figure 9.8 it is linearised at $\Delta \mathbf{t}_{\mathrm{d_{ti}}}$. It can be seen from those figures that the second approach linearises the constraint at the boundary which is the area of interest, thus the algorithm converges faster to a solution that satisfies constraints.

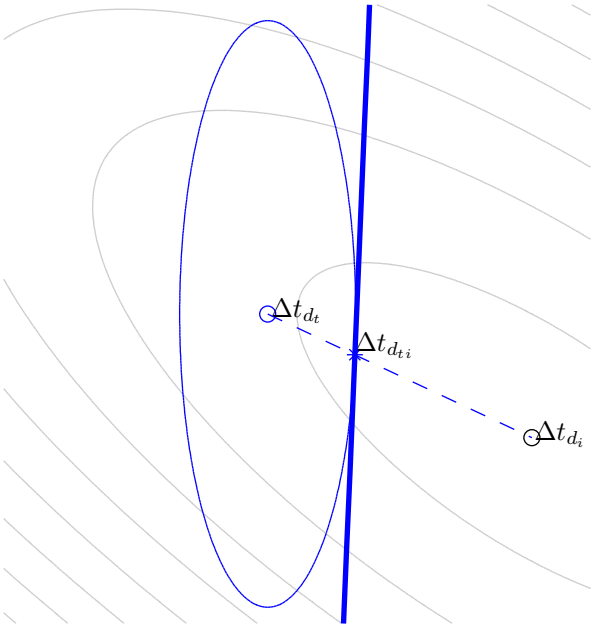


Figure 9.8: Linearise at $\Delta \mathbf{t}_{\mathrm{d_{ti}}}$

Soften linearised constraints

Recalling the example in Figure 9.6, the quadratic constraints added in this section represents the blue and red regions, and the intersection is the region which satisfies both constraints. Note that if the given value of x_1 in Figure 9.6 had been more to the left, the intersection between the vertical line defined by the given x_1 and the region that satisfies both constraints would have been null, then the problem would have been infeasible. In that case both constraints cannot be satisfied simultaneously because the situation the controller is about

to face is not contained in the identification data set. One possible solution is to treat both quadratic constraints as soft constraints.

Softening the linearised quadratic constraints in Equation (9.57) by means of slack variables

$$\min_{\Delta\mathbf{t}_{\mathrm{d}},\gamma} \quad \underbrace{J_{\mathrm{C}}(\Delta\mathbf{t}_{\mathrm{d}}) + k\gamma\gamma^{T}}_{\triangleq \bar{J}_{\mathrm{C}}} \quad s.t. \quad \begin{cases} \mathbf{A}\Delta\mathbf{t}_{\mathrm{d}}^{T} \leq \mathbf{b} \\ \begin{bmatrix} \mathbf{A}_{\mathrm{t}} \\ \mathbf{A}_{\mathrm{e}} \end{bmatrix} \Delta\mathbf{t}_{\mathrm{d}}^{T} \leq \begin{bmatrix} \mathbf{b}_{\mathrm{t}} \\ \mathbf{b}_{\mathrm{e}} \end{bmatrix} + \gamma^{T} \\ -\gamma^{T} \leq \mathbf{0} \end{cases} \tag{9.58}$$

where: γ the vector of slack variables; k, the weight of the slack variables which normally takes a large value.

Finally, if soft constraints want to be used instead of hard constraints, step 2 of the algorithm described in this subsection needs to be changed from

2. $\Delta\mathbf{t}_{\mathrm{d_i}}$ comes from solving the QP in Equation (9.57)

to

2. $\Delta\mathbf{t}_{\mathrm{d_i}}$ comes from solving the QP in Equation (9.58)

It is shown in proposition 9.6.11 that the problem in Equation (9.58) can be cast into a QP problem of an augmented vector of decisions.

Proposition 9.6.5 $\Delta\mathbf{t}_{\mathrm{d_t}}$ *which minimizes* $\bar{J}_{\mathrm{t}}$ *is obtained:*

$$\Delta\mathbf{t}_{\mathrm{d_t}} = -\mathbf{f}_{\mathrm{t}}^{T}\mathbf{H}_{\mathrm{t}}^{-1}$$

Proof *The minimum of the quadratic cost function* $\bar{J}_{\mathrm{t}}$ *in Equation* (9.39) *can be obtained equating its first derivative with respect to* $\Delta\mathbf{t}_{\mathrm{d}}$ *to* 0

$$\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta\mathbf{t}_{\mathrm{d}}} = 2\Delta\mathbf{t}_{\mathrm{d}}\mathbf{H}_{\mathrm{t}} + 2\mathbf{f}_{\mathrm{t}}^{T}$$

$$2\Delta\mathbf{t}_{\mathrm{d_t}}\mathbf{H}_{\mathrm{t}} + 2\mathbf{f}_{\mathrm{t}}^{T} = 0$$

$$\Delta\mathbf{t}_{\mathrm{d_t}} = -\mathbf{f}_{\mathrm{t}}^{T}\mathbf{H}_{\mathrm{t}}^{-1}. \quad \square$$

Proposition 9.6.6 $\Delta\mathbf{t}_{\mathrm{d_e}}$ *which minimizes* $\bar{J}_{\mathrm{e}}$ *is obtained:*

$$\Delta\mathbf{t}_{\mathrm{d_e}} = -\mathbf{f}_{\mathrm{e}}^{T}\mathbf{H}_{\mathrm{e}}^{-1}$$

Proof *The minimum of the quadratic cost function* $\bar{J}_{\mathrm{e}}$ *in Equation* (9.42) *can be obtained equating its first derivative with respect to* $\Delta\mathbf{t}_{\mathrm{d}}$ *to* 0

$$\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta\mathbf{t}_{\mathrm{d}}} = 2\Delta\mathbf{t}_{\mathrm{d}}\mathbf{H}_{\mathrm{e}} + 2\mathbf{f}_{\mathrm{e}}^{T}$$

$$2\Delta\mathbf{t}_{\mathrm{d_e}}\mathbf{H}_{\mathrm{e}} + 2\mathbf{f}_{\mathrm{e}}^{T} = 0$$

$$\Delta\mathbf{t}_{\mathrm{d_e}} = -\mathbf{f}_{\mathrm{e}}^{T}\mathbf{H}_{\mathrm{e}}^{-1}. \quad \square$$

Proposition 9.6.7 $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ *such that* $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d_{ti}}}) = 1$*, and it is aligned with the current solution* $\Delta\mathbf{t}_{\mathrm{d_i}}$ *and* $\Delta\mathbf{t}_{\mathrm{d_t}}$ *is obtained:*

$$\Delta\mathbf{t}_{\mathrm{d_{ti}}} = \Delta\mathbf{t}_{\mathrm{d_t}} + \bar{\gamma_t}(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_t}})$$

Proof $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ *is a point which satisfies:*

- $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ *is aligned with* $\Delta\mathbf{t}_{\mathrm{d_i}}$ *and* $\Delta\mathbf{t}_{\mathrm{d_t}}$
- $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ *is in-between* $\Delta\mathbf{t}_{\mathrm{d_i}}$ *and* $\Delta\mathbf{t}_{\mathrm{d_t}}$
- $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d_{ti}}}) = 1$

The first requirement can be expressed:

$$\Delta\mathbf{t}_{\mathrm{d}}(\gamma_t) = \Delta\mathbf{t}_{\mathrm{d_t}} + \gamma_t(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_t}})$$

where $\gamma_t \in \mathbb{R}$. *For the second requirement to hold* $\gamma_t \in [0, 1]$. *Substituting* $\Delta\mathbf{t}_{\mathrm{d}}$ *in proposition 9.6.1*

$$\mathbf{t}(\gamma_t) = \mathbf{x}^*_{\mathrm{p}}\mathbf{N}_{\mathrm{p}}\mathbf{Z} + \underbrace{(\Delta\mathbf{t}_{\mathrm{d_t}} + \gamma_t(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_t}}))}_{\Delta\mathbf{t}_{\mathrm{d}}}\mathbf{N}_{\mathrm{d}}\mathbf{Z}$$

which yields

$$\mathbf{t}(\gamma_t) = \underbrace{\mathbf{x}^*_{\mathrm{p}}\mathbf{N}_{\mathrm{p}}\mathbf{Z}_{+}\Delta\mathbf{t}_{\mathrm{d_t}}\mathbf{N}_{\mathrm{d}}\mathbf{Z}}_{\triangleq M_{\mathrm{t}}} + \gamma_t\underbrace{(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_t}})\mathbf{N}_{\mathrm{d}}\mathbf{Z}}_{\triangleq N_{\mathrm{t}}}$$

From Equations (9.36) *and* (9.38)

$$\bar{J}_{\mathrm{t}} = \frac{\mathbf{t}\mathbf{S}^{\mathbf{2}^{-1}}_{\mathrm{a}}\mathbf{t}^T}{J_{\mathrm{t}max}}$$

Substituting in $\mathbf{t}(\gamma_t)$ *yields*

$$\begin{aligned}\bar{J}_{\mathrm{t}}(\gamma_t) &= \frac{(M_{\mathrm{t}} + \gamma_t N_{\mathrm{t}})\mathbf{S}^{\mathbf{2}^{-1}}_{\mathrm{a}}(M^T_{\mathrm{t}} + \gamma_t N^T_{\mathrm{t}})}{J_{\mathrm{t}max}} \\ &= \underbrace{\frac{N_{\mathrm{t}}\mathbf{S}^{\mathbf{2}^{-1}}_{\mathrm{a}}N^T_{\mathrm{t}}}{J_{\mathrm{t}max}}}_{\triangleq a_{\mathrm{t}}}\gamma^2_t + \underbrace{\frac{2M_{\mathrm{t}}\mathbf{S}^{\mathbf{2}^{-1}}_{\mathrm{a}}N^T_{\mathrm{t}}}{J_{\mathrm{t}max}}}_{\triangleq b_{\mathrm{t}}}\gamma_t + \underbrace{\frac{M_{\mathrm{t}}\mathbf{S}^{\mathbf{2}^{-1}}_{\mathrm{a}}M^T_{\mathrm{t}}}{J_{\mathrm{t}max}}}_{\triangleq c_{\mathrm{t}}}\end{aligned}$$

The third requirement can be expressed

$$\bar{J}_{\mathrm{t}}(\bar{\gamma_t}) = 1$$

where $\bar{\gamma_t}$ comes from solving the above second order equation

$$\bar{\gamma_t} = \frac{-b_{\mathrm{t}} \pm \sqrt{b_{\mathrm{t}}^2 - 4a_{\mathrm{t}}(c_{\mathrm{t}} - 1)}}{2a_{\mathrm{t}}}.$$

Provided $\bar{J}_{\mathrm{t}}$ is symmetric to its minimum; and both $J_{\mathrm{t}}(\bar{\gamma_t})$ are aligned among them and with the point that minimizes $\bar{J}_{\mathrm{t}}$, the absolute value of the two solutions in the previous equation are equal. Since we are only interested in values of $\gamma_t \in [0, 1]$ we take the positive solution. Note that in case $\gamma_t > 1$, the current solution $\Delta\mathbf{t}_{\mathrm{d_i}}$ already satisfies the constraint and no linearisation of the quadratic constraint is needed. Consequently $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ need be computed only if $\bar{\gamma_t} \leq 1$.

$$\bar{\gamma_t} = \frac{-b_{\mathrm{t}} + \sqrt{b_{\mathrm{t}}^2 - 4a_{\mathrm{t}}(c_{\mathrm{t}} - 1)}}{2a_{\mathrm{t}}}.$$

And $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ can be expressed

$$\Delta\mathbf{t}_{\mathrm{d_{ti}}} = \Delta\mathbf{t}_{\mathrm{d_t}} + \bar{\gamma_t}(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_t}}). \quad \square$$

Proposition 9.6.8 *$\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ such that $\bar{J}_{\mathrm{e}}(\Delta\mathbf{t}_{\mathrm{d_{ei}}}) = 1$, and it is aligned with the current solution $\Delta\mathbf{t}_{\mathrm{d_i}}$ and $\Delta\mathbf{t}_{\mathrm{d_e}}$ is obtained:*

$$\Delta\mathbf{t}_{\mathrm{d_{ei}}} = \Delta\mathbf{t}_{\mathrm{d_e}} + \bar{\gamma_e}(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_e}})$$

Proof *$\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ is a point which satisfies:*

- *$\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ is aligned with $\Delta\mathbf{t}_{\mathrm{d_i}}$ and $\Delta\mathbf{t}_{\mathrm{d_e}}$*
- *$\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ is in-between $\Delta\mathbf{t}_{\mathrm{d_i}}$ and $\Delta\mathbf{t}_{\mathrm{d_e}}$*
- $\bar{J}_{\mathrm{e}}(\Delta\mathbf{t}_{\mathrm{d_{ei}}}) = 1$

The first requirement can be expressed:

$$\Delta\mathbf{t}_{\mathrm{d}}(\gamma_e) = \Delta\mathbf{t}_{\mathrm{d_e}} + \gamma_e(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_e}})$$

where $\gamma_e \in \mathbb{R}$. *For the second requirement to hold* $\gamma_e \in [0,1]$. *Substituting* $\Delta\mathbf{t}_{\mathrm{d}}$ *in proposition 9.6.2*

$$\mathbf{e}(\gamma_e) = \mathbf{x}_{\mathrm{p}}^{*}\mathbf{E}_{\mathrm{p}} + \underbrace{(\Delta\mathbf{t}_{\mathrm{d_e}} + \gamma_e(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_e}}))}_{\Delta\mathbf{t}_{\mathrm{d}}}\mathbf{E}_{\mathrm{d}}$$

which yields

$$\mathbf{e}(\gamma_e) = \underbrace{\mathbf{x}_{\mathrm{p}}^{*}\mathbf{E}_{\mathrm{p}} + \Delta\mathbf{t}_{\mathrm{d_e}}\mathbf{E}_{\mathrm{d}}}_{\triangleq M_{\mathrm{e}}} + \gamma_e\underbrace{(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_e}})\mathbf{E}_{\mathrm{d}}}_{\triangleq N_{\mathrm{e}}}$$

From Equations (9.37) *and* (9.38)

$$\bar{J}_{\mathrm{e}} = \frac{\mathbf{e}\mathbf{e}^T}{J_{\mathrm{e}max}}$$

Substituting in $\mathbf{e}(\gamma_e)$ *yields*

$$\bar{J}_{\mathrm{e}}(\gamma_e) = \frac{(M_{\mathrm{e}} + \gamma_e N_{\mathrm{e}})(M_{\mathrm{e}}^T + \gamma_e N_{\mathrm{e}}^T)}{J_{\mathrm{e}max}} \tag{9.59}$$

$$= \underbrace{\frac{N_{\mathrm{e}}N_{\mathrm{e}}^T}{J_{\mathrm{e}max}}}_{\triangleq a_{\mathrm{e}}}\gamma_e^2 + \underbrace{\frac{2M_{\mathrm{e}}N_{\mathrm{e}}^T}{J_{\mathrm{e}max}}}_{\triangleq b_{\mathrm{e}}}\gamma_e + \underbrace{\frac{M_{\mathrm{e}}M_{\mathrm{e}}^T}{J_{\mathrm{e}max}}}_{\triangleq c_{\mathrm{e}}} \tag{9.60}$$

The third requirement can be expressed

$$\bar{J}_{\mathrm{e}}(\bar{\gamma_e}) = 1$$

where $\bar{\gamma_e}$ *comes from solving the above second order equation*

$$\bar{\gamma_e} = \frac{-b_{\mathrm{e}} \pm \sqrt{b_{\mathrm{e}}^2 - 4a_{\mathrm{e}}(c_{\mathrm{e}} - 1)}}{2a_{\mathrm{e}}}.$$

Provided $\bar{J}_{\mathrm{e}}$ is symmetric to its minimum; and both $J_{\mathrm{e}}(\bar{\gamma_e})$ are aligned among them and with the point that minimizes $\bar{J}_{\mathrm{e}}$, the absolute value of the two solutions in the previous equation are equal. Since we are only interested in values of $\gamma_e \in [0,1]$ we take the positive solution. Note that in case $\gamma_e > 1$, the current solution $\Delta\mathbf{t}_{\mathrm{d_i}}$ already satisfies the constraint and no linearisation of the quadratic constraint is needed. Consequently $\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ need be computed only if $\bar{\gamma_e} \leq 1$.

$$\bar{\gamma_e} = \frac{-b_{\mathrm{e}} + \sqrt{b_{\mathrm{e}}^2 - 4a_{\mathrm{e}}(c_{\mathrm{e}} - 1)}}{2a_{\mathrm{e}}}.$$

And $\Delta\mathbf{t}_{\mathrm{d_{ei}}}$ can be expressed

$$\Delta\mathbf{t}_{\mathrm{d_{ei}}} = \Delta\mathbf{t}_{\mathrm{d_e}} + \bar{\gamma_e}(\Delta\mathbf{t}_{\mathrm{d_i}} - \Delta\mathbf{t}_{\mathrm{d_e}}). \quad \square$$

Proposition 9.6.9 *The first-order Taylor approximation of the quadratic constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ at a point β can be expressed*

$$\mathbf{A}_{\mathrm{t_i}}\Delta\mathbf{t}_{\mathrm{d}}^T \leq \mathbf{B}_{\mathrm{t_i}}$$

Proof *The first-order Taylor approximation of the quadratic constraint $\bar{J}_{\mathrm{t}}(\Delta\mathbf{t}_{\mathrm{d}}) \leq 1$ at a point β*

$$\left.\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta\mathbf{t}_{\mathrm{d}}}\right|_{\beta} (\Delta\mathbf{t}_{\mathrm{d}} - \beta) + \bar{J}_{\mathrm{t}}(\beta) \leq 1 \tag{9.61}$$

$\beta = \Delta\mathbf{t}_{\mathrm{d_i}}$ or $\beta = \Delta\mathbf{t}_{\mathrm{d_{ti}}}$ depending on which linearisation point is selected. The first derivative of $\bar{J}_{\mathrm{t}}$ in Equation (9.39) with respect to $\Delta\mathbf{t}_{\mathrm{d}}$

$$\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta\mathbf{t}_{\mathrm{d}}} = 2\Delta\mathbf{t}_{\mathrm{d}}\mathbf{H}_{\mathrm{t}} + 2\mathbf{f}_{\mathrm{t}}^T$$

then

$$\left.\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} = 2\beta \mathbf{H}_{\mathrm{t}} + 2\mathbf{f}_{\mathrm{t}}^{T}.$$

Reorganising terms in Equation (9.61)

$$\underbrace{\left.\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta}}_{\triangleq \mathbf{A}_{\mathrm{t_i}}} \Delta \mathbf{t}_{\mathrm{d}} \leq \underbrace{1 - \bar{J}_{\mathrm{t}}(\beta) + \left.\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} \beta}_{\triangleq \mathbf{B}_{\mathrm{t_i}}}$$

Note that for $\beta = \Delta \mathbf{t}_{\mathrm{d_{ti}}}$, $\bar{J}_{\mathrm{t}}(\beta) = \bar{J}_{\mathrm{t}}(\Delta \mathbf{t}_{\mathrm{d_{ti}}}) = 1$ *hence*

$$\mathbf{B}_{\mathrm{t_i}} = \left.\frac{\partial \bar{J}_{\mathrm{t}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} \beta. \quad \square$$

Proposition 9.6.10 *The first-order Taylor approximation of the quadratic constraint* $\bar{J}_{\mathrm{e}}(\Delta \mathbf{t}_{\mathrm{d}}) \leq 1$ *at a point* β *can be expressed*

$$\mathbf{A}_{\mathrm{e_i}} \Delta \mathbf{t}_{\mathrm{d}}^{T} \leq \mathbf{B}_{\mathrm{e_i}}$$

Proof *The first-order Taylor approximation of the quadratic constraint* $\bar{J}_{\mathrm{e}}(\Delta \mathbf{t}_{\mathrm{d}}) \leq 1$ *at a point* β *can be expressed*

$$\left.\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} (\Delta \mathbf{t}_{\mathrm{d}} - \beta) + \bar{J}_{\mathrm{e}}(\beta) \leq 1 \tag{9.62}$$

$\beta = \Delta \mathbf{t}_{\mathrm{d_i}}$ *or* $\beta = \Delta \mathbf{t}_{\mathrm{d_{ei}}}$ *depending on which linearisation point is selected. The first derivative of* $\bar{J}_{\mathrm{e}}$ *in Equation* (9.42) *with respect to* $\Delta \mathbf{t}_{\mathrm{d}}$

$$\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}} = 2\Delta \mathbf{t}_{\mathrm{d}} \mathbf{H}_{\mathrm{e}} + 2\mathbf{f}_{\mathrm{e}}^{T}$$

then

$$\left.\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} = 2\beta \mathbf{H}_{\mathrm{e}} + 2\mathbf{f}_{\mathrm{e}}^{T}.$$

Reorganising terms in Equation (9.62)

$$\underbrace{\left.\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta}}_{\triangleq \mathbf{A}_{\mathrm{e_i}}} \Delta \mathbf{t}_{\mathrm{d}} \leq \underbrace{1 - \bar{J}_{\mathrm{e}}(\beta) + \left.\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} \beta}_{\triangleq \mathbf{B}_{\mathrm{e_i}}}$$

Note that if $\beta = \Delta \mathbf{t}_{\mathrm{d_{ei}}}$, *then* $\bar{J}_{\mathrm{e}}(\beta) = 1$ *and*

$$\mathbf{B}_{\mathrm{e_i}} = \left.\frac{\partial \bar{J}_{\mathrm{e}}}{\partial \Delta \mathbf{t}_{\mathrm{d}}}\right|_{\beta} \beta. \quad \square$$

Proposition 9.6.11 *The minimization problem in Equation* (9.58) *can be expressed as a QP in terms of the augmented vector* $\mathbf{v}$.

Proof *Lets define the augmented vector*

$$\mathbf{v} \triangleq [\Delta \mathbf{t}_{\mathrm{d}} \quad \gamma].$$

Lets take the expanded expression for J_{C} *derived in proposition 9.4.3:*

$$J_{\mathrm{C}}(\Delta \mathbf{t}_{\mathrm{d}}) = \Delta \mathbf{t}_{\mathrm{d}} \mathbf{H} \Delta \mathbf{t}_{\mathrm{d}}^{T} + 2\mathbf{f}^{T} \Delta \mathbf{t}_{\mathrm{d}}^{T} + C$$

and transform it to tackle the minimization problem in Equation (9.58)*:*

$$\bar{J}_{\mathrm{C}}(\mathbf{v}) = \mathbf{v} \underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & k\mathbf{I} \end{bmatrix}}_{\triangleq \mathbf{H}_{\mathrm{v}}} \mathbf{v}^{T} + 2 \underbrace{\begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}^{T}}_{\triangleq \mathbf{f}_{\mathrm{v}}^{T}} \mathbf{v}^{T} + C \tag{9.63}$$

Finally the problem in Equation (9.58) *yields*

$$\min_{\mathbf{v}} \quad \mathbf{v}\mathbf{H}_{\mathrm{v}}\mathbf{v}^T + 2\mathbf{f}_{\mathrm{v}}^T\mathbf{v}^T \quad s.t \quad \begin{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{A}_{\mathrm{t}} \\ \mathbf{A}_{\mathrm{e}} \end{bmatrix} & \begin{matrix} \mathbf{0} \\ -\mathbf{I} \end{matrix} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \mathbf{v}^T \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{b}_{\mathrm{t}} \\ \mathbf{b}_{\mathrm{e}} \\ \mathbf{0} \end{bmatrix} \tag{9.64}$$

9.7. Systematic tuning

This Section defines how to systematically tune n_{u} and n_{f}. It is convenient in MPC to choose n_{u} and n_{f} such that $n_{\mathrm{f}} - n_{\mathrm{u}}$ is greater than or equal to the process settling time [Rossiter 03].

n_{u} is often chosen as a trade-off solution between computational complexity and achieved performance. As a rule of thumb n_{u} should be in-between a minimum and a maximum value [Shridhar 98]: the minimum value of n_{u} is such that the output at instant $k + n_{\mathrm{u}}$ has reached at least 60% of its steady state value; and the maximum value of n_{u} is such that increasing n_{u} has no further effect on the first move of the controller to a step change in the set point. Provided control in LV-MPC is implemented in the reduced space of the latent variables, large values of n_{u} are tenable in terms of computational complexity, then the rule of thumb in LV-MPC is to set n_{u} as half the process settling time:

$$n_{\mathrm{u}} = \frac{T_{\mathrm{ss}}}{2} = 0.5T_{\mathrm{ss}}$$

then assuming $n_{\mathrm{f}} - n_{\mathrm{u}} = T_{\mathrm{ss}}$, n_{f} yields

$$n_{\mathrm{f}} = T_{\mathrm{ss}} + n_{\mathrm{u}} = 1.5T_{\mathrm{ss}}.$$

Settling time is defined as the time required for the response curve to reach and stay within a range of certain percentage of the final value. The percentage is usually 5% or 2%, however, for oscillatory processes or processes with fast and slow dynamics, the settling time at 5% yields a large value for n_f then multi-step ahead predictions perform poorly, and consequently the resulting control. For processes with combined fast and slow or oscillatory dynamic behaviour, one can set T_ss to capture the most relevant dynamics.

9.8. Stability analysis

A priori stability guarantee of an MPC control law is established if the MPC cost function is a Lyapunov cost function. The MPC cost function in Equation (9.29) is a Lyapunov cost function if:

- It is positive-definite, then $\mathbf{W}_\mathrm{y}$ and $\mathbf{W}_\mathrm{u}$ need be positive-definite matrices
- Its time-derivative is negative semidefinite, which is attained if $n_\mathrm{u} = n_\mathrm{f} = \infty$ [Rossiter 03]

The use of infinite horizons make MPC intractable in the case of constraints. For this problem to be tractable one can use a dual mode implementation of MPC. In dual mode MPC the following two modes are commonly used:

- In the first mode there are n_c control moves and constrained on-line optimization is used to fix those degrees of freedom
- In the second mode, control moves are given by a feedback law

For constraints to hold in the second mode, the state of the process at the end of the first mode need be in a terminal set. The terminal set is defined by the

control law in the second mode such that: the more tightly tuned the control law in the second mode is, the smaller the terminal set will be.

The two tuning parameters in dual mode MPC are:

- The number of control moves n_c
- Terminal control law

The terminal control law can be defined such that the infinite horizon cost in the constraint free case is minimized, this is denoted LQMPC (Linear Quadratic optimal MPC). If the LQMPC is tightly tuned, then feasible regions may not be large and the control law may not be robust. Alternatively one can set the terminal control law to zero, this is denoted NTC (No Terminal Control). NTC is the most popular approach deployed in Industry, in fact DMC and GPC are NTC dual mode strategies [Rossiter 03]. For NTC to be appropriate, the system should be open-loop stable provided the open-loop behaviour of the system is obtained in the second mode.

For large values of n_c much of the transients are shaped in the first mode and the choice of the terminal control law will have negligible effect on control performance. Consequently, if n_c is large, one can either use a loosely tuned LQMPC, or NTC if the process is open-loop stable. A drawback in using a large n_c is an increase in computational complexity. However, the control methodology proposed in this thesis —LV-MPC— reduces computational complexity by performing the optimization in the latent variable space, thus making it possible to choose a large n_c.

LV-MPC implements NTC dual mode MPC with the following two modes:

- In the **first mode** there are n_u control moves and constrained on-line optimization is implemented. From Section 9.7 $n_u = 0.5T_{ss}$, and assuming closed-loop specifications at least halve the process settling time, the

dynamic behaviour is shaped in the first mode and the choice of the terminal control law has negligible effect on control performance.

- In the **second mode** the terminal control law is set to zero, then the open-loop response of the process is obtained. If n_f is defined such that $n_\mathrm{f} - n_\mathrm{u}$ is greater than the process settling time, then the resulting control with a finite n_f is equivalent to control with infinite horizon. Note constraints are defined just for MVs and increments on the MVs, then if constraints hold in the first mode, they also hold in the second mode provided all increments are zero in the second mode, and the last value for the MVs in the first mode is used in the second mode. Consequently there is no need for a terminal region in the first mode.

Summing up, LV-MPC implements the heuristic and commonly used approach for stability that applies for open-loop stable processes with $n_\mathrm{f} - n_\mathrm{u}$ greater than the process settling time.

9.9. LV-MPC Implementation

This section aims at clarifying how to implement LV-MPC. The sequence of steps to implement LV-MPC can be split into three stages: **(I)** obtain the predictor, **(II)** obtain the matrices of the controller, **(III)** evaluate the controller on-line.

(I) Obtain the predictor:

- Obtain the identification and validation data sets.
- Set sampling time T_s and control and prediction horizons n_u and n_f. The directions given in Section 9.7 can be followed to set this three parameters.

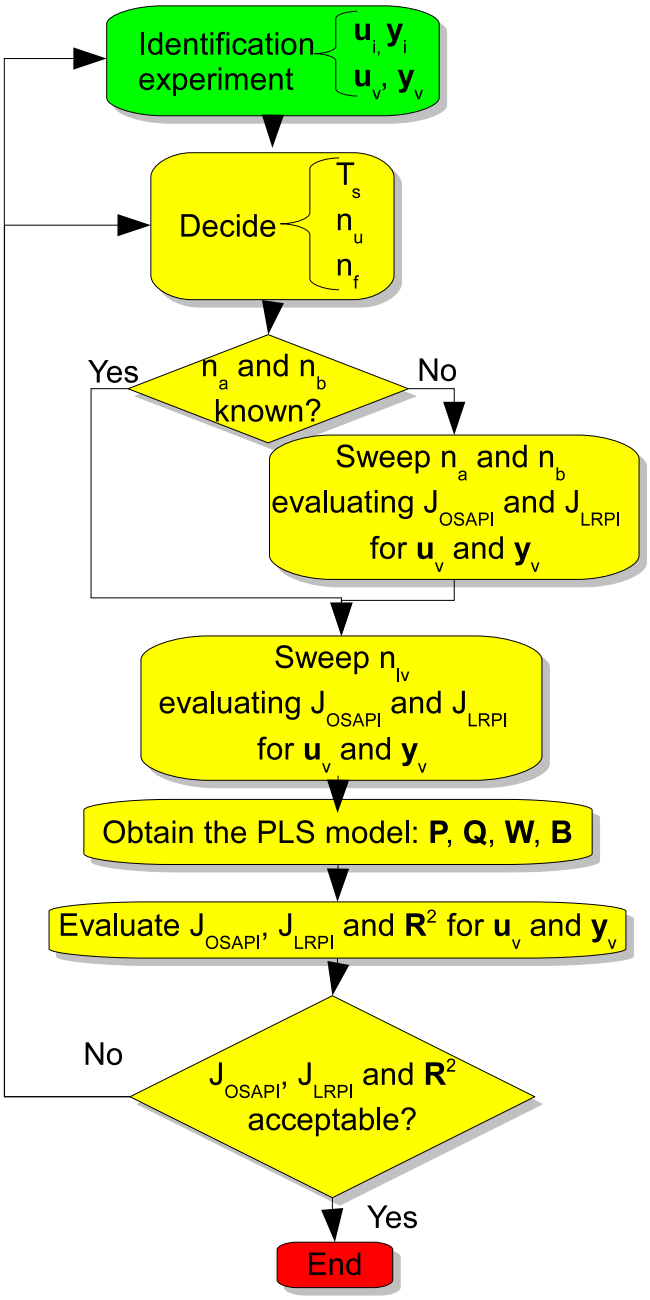


Figure 9.9: Pseudo-code for obtaining the predictor.

- If not available, decide the order of the linear model n_{a} and n_{b}. The order of the model can be swept and evaluate some indicators of the quality of predictions, such as J_{OSAPI} in Equation (5.13) and/or J_{LRPI} in Equation (5.14) for the validation data set . From the sweep, one can decide the order of the model as the order that minimizes the quality indicators evaluated.

- Sweep n_{lv} in the PLS model and calculate J_{OSAPI} and J_{LRPI} for the validation data set. Set n_{lv} as the value above which no significant improvement in terms of J_{OSAPI} and J_{LRPI} is attained.

- Obtain the matrices that define the PLS model: $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{B}$. (see Subsection 5.3.2)

- Evaluate the quality of predictions for the validation data set and decide whether it is enough, and hence stop, or needs improvement. If the predictor is to be improved, one can start over with the identification procedure either obtaining a new data set, or changing some of the decisions. The two previous quality indicators, J_{OSAPI} and J_{LRPI}, may be used along with the coefficient of determination $\mathbf{R}_2$ in Equation (7.16).

(II) Obtain the matrices of the controller:

- Decide some tuning parameters of the controller: λ_u, $\mathbf{W}_{\mathrm{u}}$ and $\mathbf{W}_{\mathrm{y}}$.

- If Hessian conditioning is to be improved, as explained in Section 9.3, projection matrices $\bar{\mathbf{M}}_{\mathrm{dof}}$ and $\bar{\mathbf{M}}_{\mathrm{t}}$ in Equations (9.20) and (9.18) are to be used. In the basic approach however, projection matrices $\mathbf{M}_{\mathrm{dof}}$ and $\mathbf{M}_{\mathrm{t}}$ in Equations (9.8) and (9.11) are used. α and β need be tuned to obtain $\bar{\mathbf{M}}_{\mathrm{dof}}$ and $\bar{\mathbf{M}}_{\mathrm{t}}$, one of them may be set to one, and the other swept to find the value that minimizes the conditioning of the resulting matrix $\mathbf{H}$.

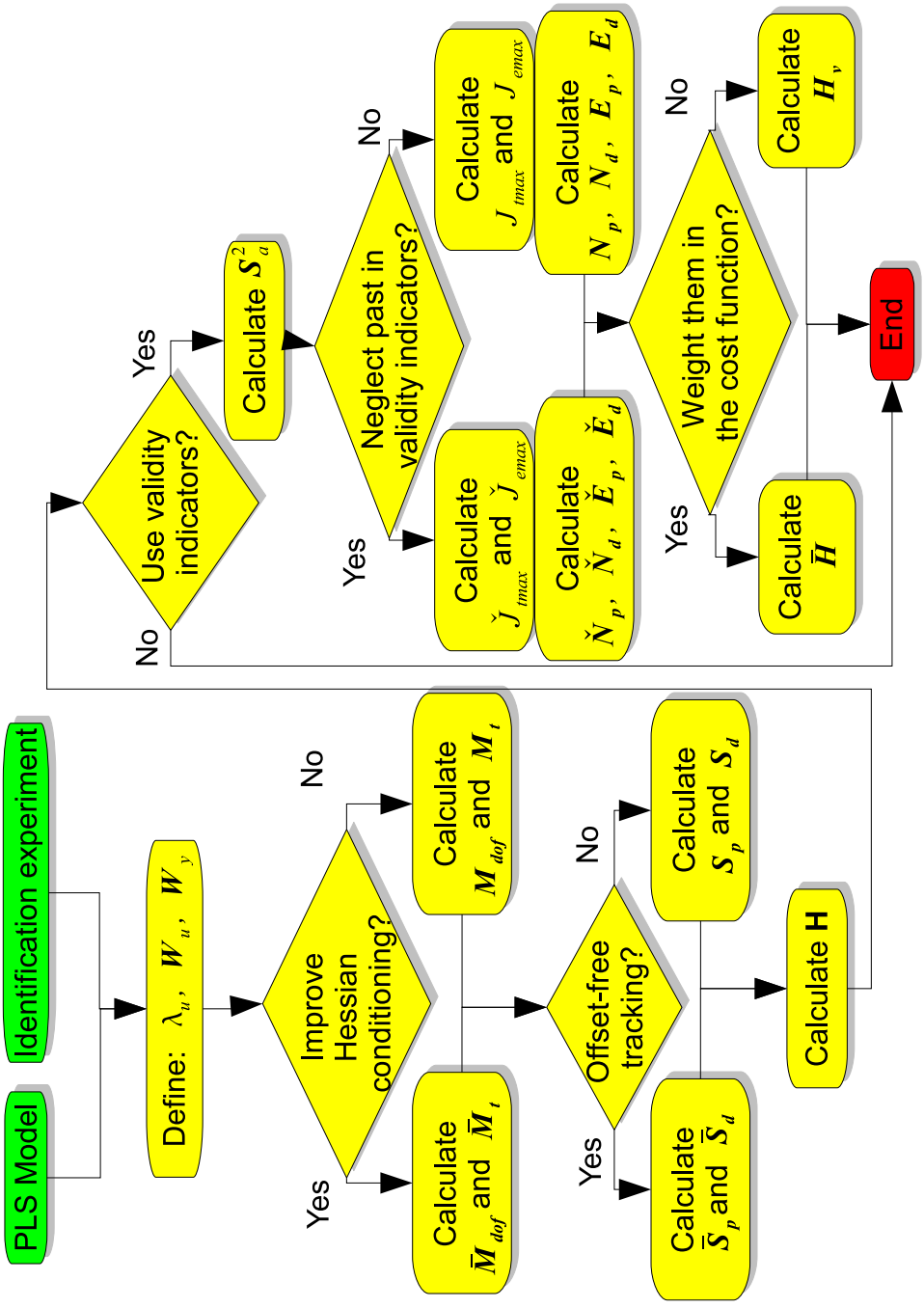


Figure 9.10: Pseudo-code for obtaining the matrices of the controller.

- Calculate the matrices of the predictor as a function of the degrees of freedom of the controller: $\Delta\mathbf{t}_\mathrm{d}$ if offset-free tracking is to be attained or $\mathbf{t}_\mathrm{d}$ otherwise. If offset-free tracking is desirable $\bar{\mathbf{S}}_\mathrm{p}$ and $\bar{\mathbf{S}}_\mathrm{d}$ in proposition 9.4.1 are obtained. In the basic control methodology however, offset-free tracking is not considered and $\mathbf{S}_\mathrm{p}$ and $\mathbf{S}_\mathrm{d}$ in proposition 9.2.2 are obtained.

- Calculate the Hessian matrix $\mathbf{H}$ using the expression in Equation (9.12) or in Equation (9.32), depending on whether offset-free tracking is needed or not.

- The controller can already be regarded as completely defined if indicators on validity of predictions are not to be considered in the controller. Otherwise, consider the following items:

  - Calculate $\mathbf{S^2}_\mathrm{a}$ in Equation 9.36.
  - If validity indicators should neglect past data, $\check{J}_\mathrm{tmax}$, $\check{J}_\mathrm{emax}$, $\check{\mathbf{N}}_\mathrm{p}$, $\check{\mathbf{N}}_\mathrm{d}$, $\check{\mathbf{E}}_\mathrm{p}$, and $\check{\mathbf{E}}_\mathrm{d}$ defined in Subsection 9.6.2 are obtained. However, if validity indicators should include past data, J_tmax, J_emax, $\mathbf{N}_\mathrm{p}$, $\mathbf{N}_\mathrm{d}$, $\mathbf{E}_\mathrm{p}$, and $\mathbf{E}_\mathrm{d}$ defined in Subsection 9.6.1 are to be obtained.
  - If indicators on validity of predictions are to be weighted in the cost function, a term is added to matrix $\mathbf{H}$ as shown in Equation (9.54) which yields $\bar{\mathbf{H}}$. λ_t and λ_e are needed to obtain $\bar{\mathbf{H}}$ and can be given an initial value of 1, and be readjusted in closed-loop operation to avoid constraints on validity take values outside the desired region.
  - Otherwise, if indicators on validity of predictions are to be added as soft constraints, the decision vector is expanded with slack variables and hence is $\mathbf{H}$ as shown in Equation (9.63) which yields $\mathbf{H}_\mathrm{v}$. k in Equation (9.63) weights slack variables and is often given a large value, i.e. $k = 1 \cdot 10^{10}$.

(III) Evaluate the controller on-line:

- Form row vector $\mathbf{x}_\mathrm{p}$ using Equation (5.16), or $\mathbf{x}_\mathrm{p}*$ in Equation (9.30) if offset-free tracking is to be attained.
- Obtain the matrices needed to solve the QP:
  - Calculate $\mathbf{f}$ using Equation (9.13), or (9.33) if offset-free tracking is to be attained. Then, if soft constraints for validity indicators are to be considered, obtain $\mathbf{f}_\mathrm{v}$ from $\mathbf{f}$ as shown in Equation (9.63).
  - Take $\mathbf{H}$ of the controller.
  - Calculate $\mathbf{A}$ and $\mathbf{b}$ that define constraints as in Section 9.5. Then Add $\mathbf{A}_\mathrm{t}$, $\mathbf{A}_\mathrm{e}$, $\mathbf{b}_\mathrm{t}$ and $\mathbf{b}_\mathrm{e}$ if validity constraints are to be linearised as explained in Subsection 9.6.4.
- Solve the QP.
- Obtain $\mathbf{x}_\mathrm{dof}$ form $\mathbf{t}_\mathrm{d}$ as shown in proposition 9.2.1, or from $\Delta\mathbf{t}_\mathrm{d}$ from proposition 9.4.2 in the offset-free case.
- Applying the receding horizon policy, $\mathbf{u}_\mathrm{k}$ is extracted from the control sequence ($\mathbf{x}_\mathrm{dof}$) and applied to the process.

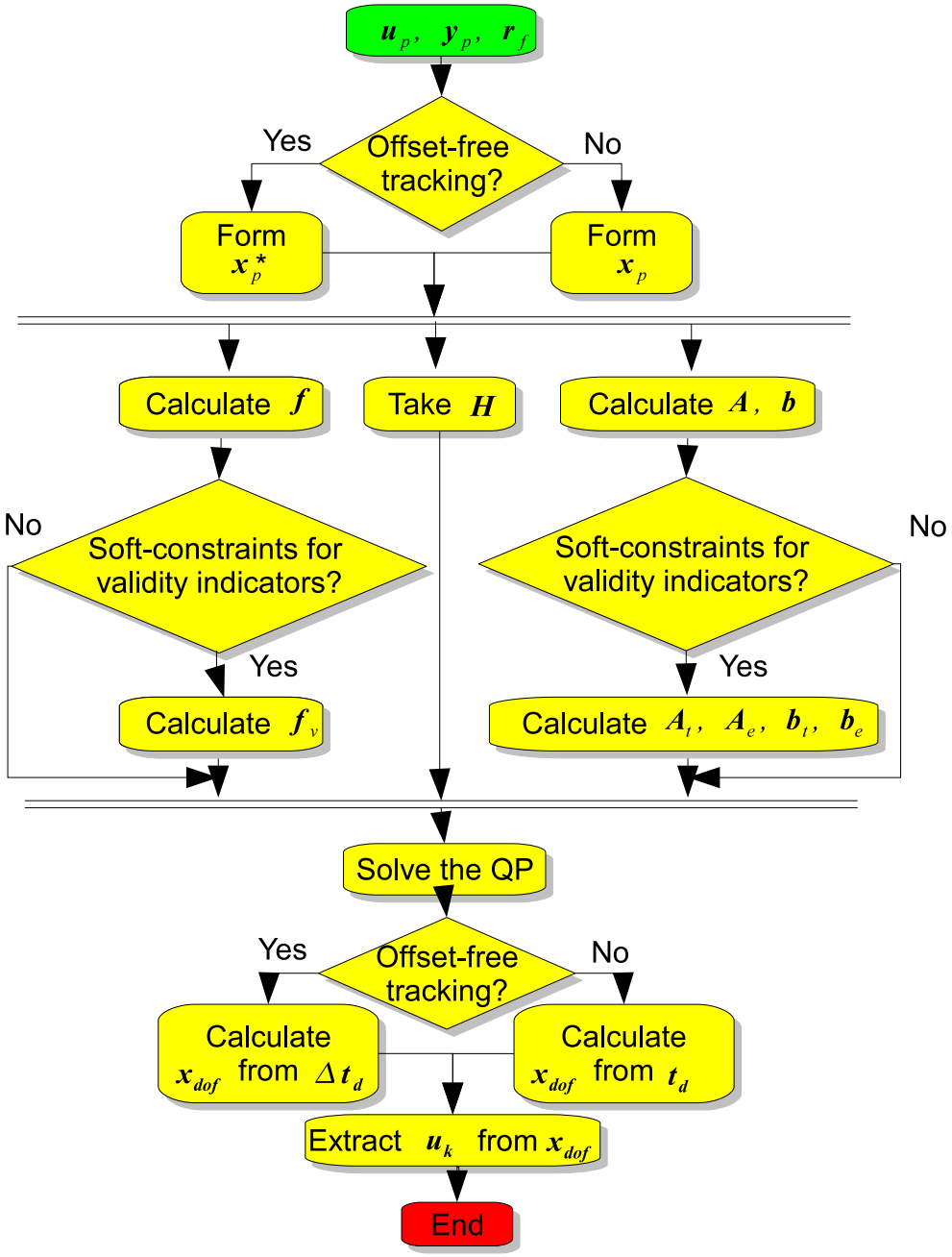


Figure 9.11: Pseudo-code for evaluating the controller.

CHAPTER 10

LV-MPC CASE STUDIES

The proposed LV-MPC methodology is evaluated in this chapter and compared to traditional MPC. The traditional MPC approach implemented uses the dynamic matrix model to account for the use of the model in a prediction window, and is denoted DM in this chapter. Four examples are used for this comparison. First, the model of an ill-conditioned distillation column is controlled proving LV-MPC can outperform DM in terms of computational complexity and closed-loop performance; also the need to implement the approach explained in Section 9.3 that improves Hessian conditioning is demonstrated. Second, a Twin-rotor non-linear simulink model is controlled proving the LV-MPC approach can outperform DM in the event of a failure provided the optimization is performed in the reduced latent variable space and model validity can be taken into account in the decision of the controller. Third, the twin-rotor model with an added perturbation is controller proving LV-MPC can outperform DM in the event of perturbation. Fourth, the model of a 2x2 Boiler is controlled comparing using traditional validity indicators and those neglecting past data.

10.1. Control of a Distillation column model

In this section a 2x2 MIMO process is defined, identified, and controlled by means of three control methodologies:

DM Traditional data-driven MPC approach. In DM, the dynamic matrix of the predictor is obtained directly from input-output data, hence it is MRI (Model predictive control Relevant Identification).

LV-MPC The methodology proposed in this part accounting for Hessian conditioning, offset-free tracking, and validity of the model by weighting validity indicators in the control cost function.

LV-MPC-basic The methodology proposed in this part accounting for offset-free tracking, and validity of the model by weighting validity indicators in the control cost function.

The process to control is a distillation column. Control of a distillation column is challenging provided it is ill-conditioned and multivariate with strong interactions between its outputs.

10.1.1. Process description

The process to control is the distillation column in [Skogestad 97]. The distillation column is running in the so-called LV-configuration, so the process is 2x2 in terms of control. The variables in the process and their working points are:

- MVs:
  - u_1: L, reflux flow (2.706 kmol/min)

  - u_2: V, boilup flow (3.206 kmol/min)

- CVs:
  - y_1: x_D, distilled product composition (0.99 mole fraction)
  - y_2: x_B, bottom product composition (0.01 mole fraction)

- Unmeasured disturbances:
  - F: feed rate (1 kmol/min)
  - zF: feed composition (0.5 mole fraction)

Sample time is set to $T_\mathrm{s} = 1\mathrm{min}$. Random walk disturbances are added to CVs and unmeasured disturbances. The added disturbances are obtained as low-pass filtered uncorrelated white noises

$$F(z) = \frac{0.02}{1 - 0.98z^{-1}}$$

and scaled so that the amplitude of the added disturbances are

- ± 0.01 for x_D and x_B (y_1, y_2)
- ± 0.1 for F
- ± 0.05 for zF

The block diagram of the process is depicted in Figure 10.1.

10.1.2. Control parameters

It is convenient in MPC to choose $\mathbf{n}_\mathrm{u}$ and $\mathbf{n}_\mathrm{f}$ such that $\mathbf{n}_\mathrm{f} - \mathbf{n}_\mathrm{u}$ is greater than or equal to the process settling time (see Section 9.7). From the response of the

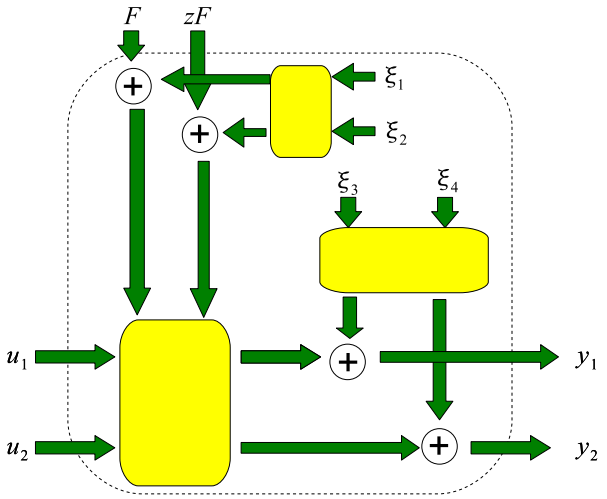


Figure 10.1: Block diagram of the Distillation column.

outputs to a simultaneous 1% step in the inputs, Figure 10.2, $T_{ss} \approx 500$min. If $\mathbf{n}_f - \mathbf{n}_u$ is set to 500, the prediction horizon will be too large, and provided there is modelling error, the predictor will perform poorly in the far horizon deteriorating the overall performance. In this example $\mathbf{n}_f - \mathbf{n}_u$ is set to 150min as a trade-off between accounting for most of the dynamic behaviour, and not having a too large prediction horizon.

$$\mathbf{n}_f = \mathbf{n}_u + 150$$

According to the directions given in Section 9.7, n_u yields

$$n_u = \frac{150}{2} = 75$$

It is noted that with normal MPC such large n_u would not be tenable, and users would make use of any of the strategies to reduce computational complexity commented in Chapter 1: move blocking, Laguerre functions, or the LVM approach presented in this part.

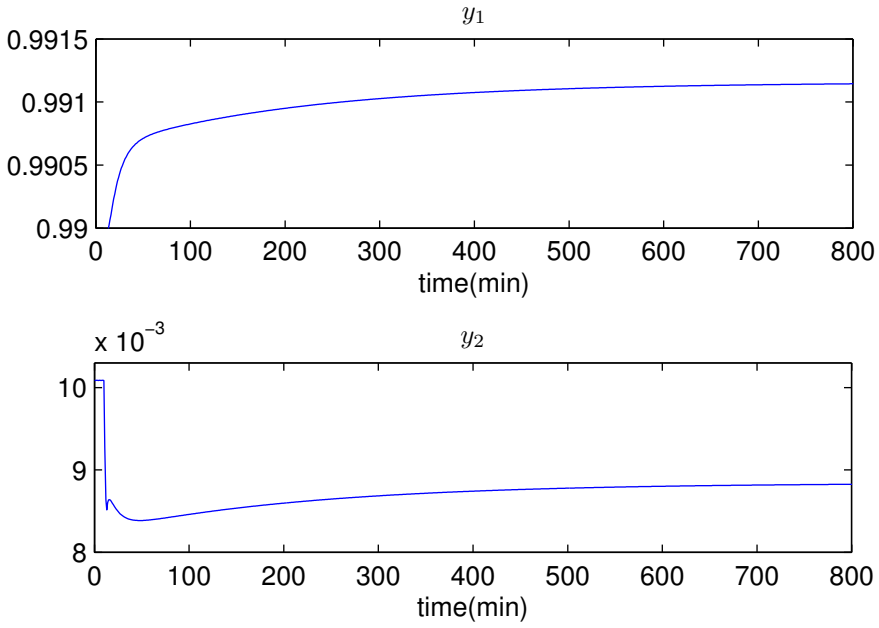


Figure 10.2: Step response of the process with no added noise.

Constraints are defined for the MVs $1 \leq \mathbf{u}_{\mathrm{k}} \leq 4$, and for the rate of the MVs $-0.5 \leq \Delta\mathbf{u}_{\mathrm{k}} \leq 0.5$. Future references are assumed unknown,

$$\mathbf{r}_{\mathrm{k+i}} = \mathbf{r}_{\mathrm{k+1}}, \quad \forall i \in [1, n_{\mathrm{f}}]$$

the weight of the control moves is set as 0.1 multiplied by a normalizing factor[1]

$$\lambda_u = 0.1\frac{n_{\mathrm{f}}}{n_{\mathrm{u}}}$$

λ_t and λ_e, which weight validity indicators $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ in the cost function, see Subsection 9.6.3, are tuned on-line such that $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ in closed loop operation stay within the region defined by the observations in the identification data set ($\bar{J}_{\mathrm{t}} \leq 1$ and $\bar{J}_{\mathrm{e}} \leq 1$). In this example $\lambda_t = 10$ and $\lambda_e = 1$.

[1]Provided there are n_{f} error terms and n_{u} increments on the control action in the problem in Equation (9.29), a scaling factor has been added to λ_u. The scaling factor reduces the influence of n_{u} in the trade-off between tracking error and control effort.

10.1.3. Identification

The identification data set is obtained in closed loop. To obtain the identification data set the process is controlled using two PID controllers [Skogestad 97]; steps are applied to the set points of the CVs; and low-pass filtered white noise is added to the MVs. The block diagram of the closed-loop scheme for closed-loop identification is shown in Figure 10.3, and the identification data set can be found in Figure 10.4.

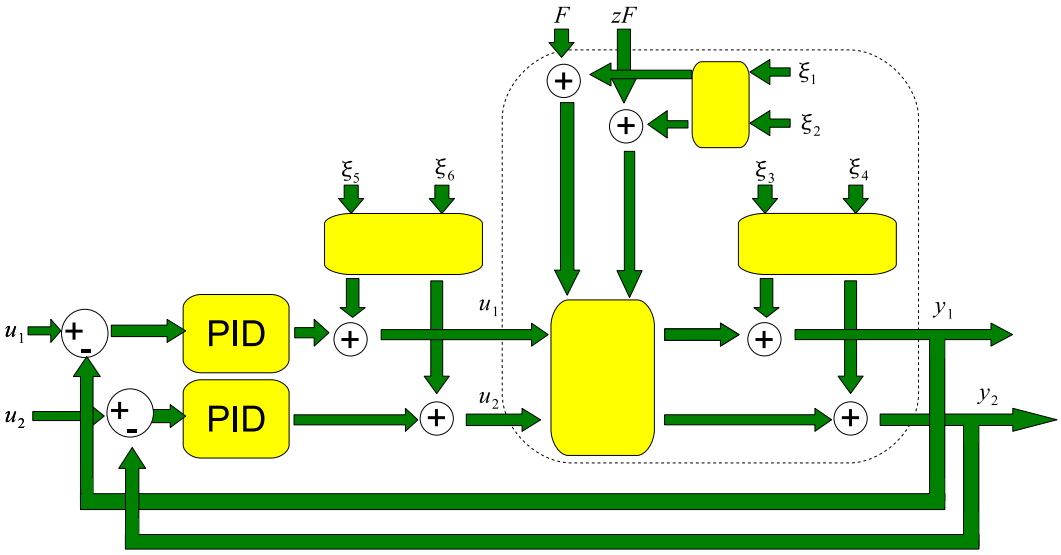


Figure 10.3: Block diagram for closed-loop identification.

Although the distillation column is non-linear, according to [Skogestad 97] the behaviour of the process is much less dependent on the operating point if the CVs are transformed:

$$\bar{y_i} = ln\left(\frac{y_i}{1-y_i}\right), \quad \forall i = \{1,2\}$$

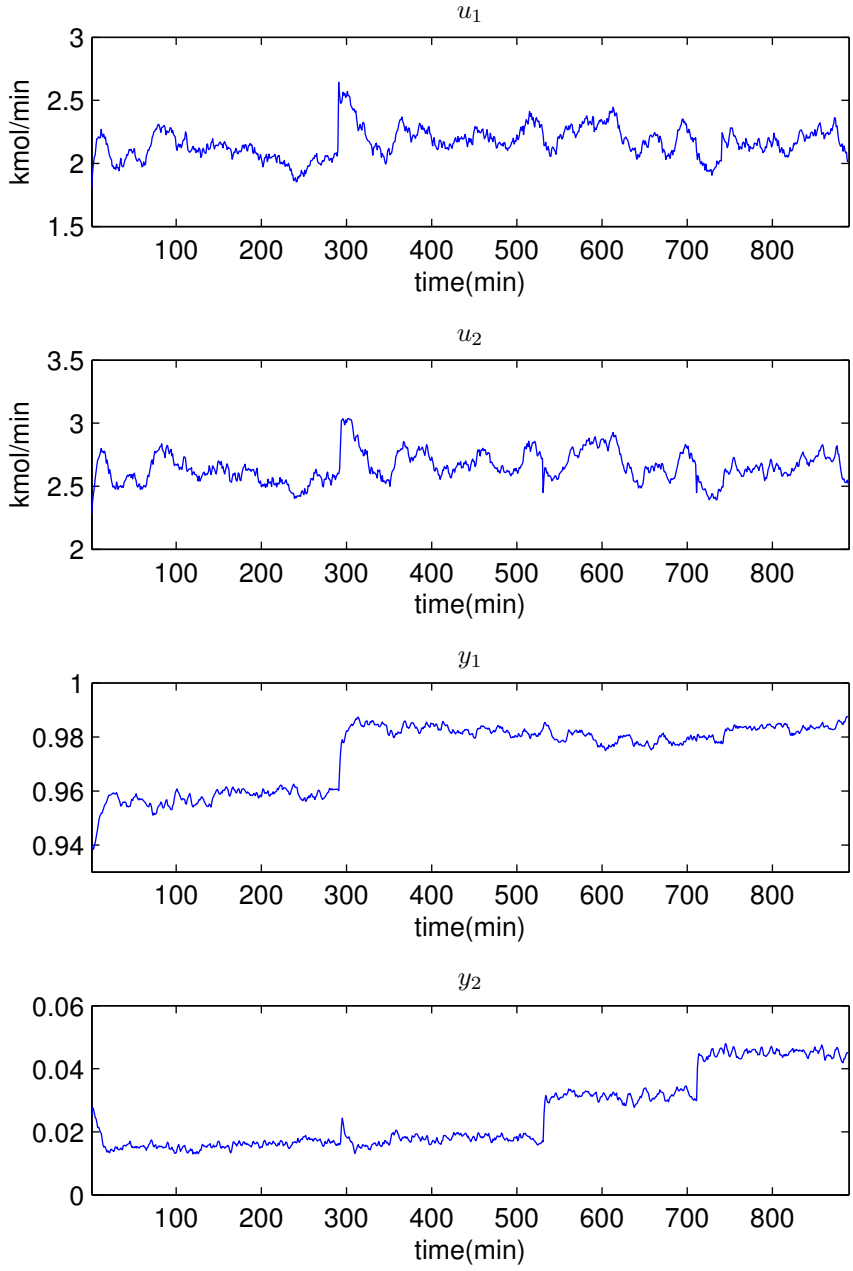


Figure 10.4: Identification data set

The CVs are first transformed using the logarithmic function, and then mean-centered and scaled previous to model identification. MVs are mean-centered and scaled as well. From [Skogestad 97] the process can be successfully approximated with a linear model of order 10, hence $n_\mathrm{a} = 10, \quad n_\mathrm{b} = 10$.

The numbers of columns in vectors $\hat{\mathbf{y}}_\mathrm{f}(k)$ and $\mathbf{x}(k)$ in Equation (5.16) are:

- $n_\mathrm{f} n_\mathrm{o} = 450$ columns in $\hat{\mathbf{y}}_\mathrm{f}(k)$.
- $(n_\mathrm{b} - 1)n_\mathrm{i} + n_\mathrm{a} n_\mathrm{o} + n_\mathrm{f} n_\mathrm{i} = 488$ columns in $\mathbf{x}(k)$.

thus the size of $\boldsymbol{\theta}$ in Equation (5.16) is 488×450. In DM, $\boldsymbol{\theta}$ is fitted to the identification data set using least squares. In LV-MPC however, PLS is used to obtain the predictor in the latent variable space as well as $\boldsymbol{\theta}$. Weighting is used in PLS identification, see Equation 5.24, with

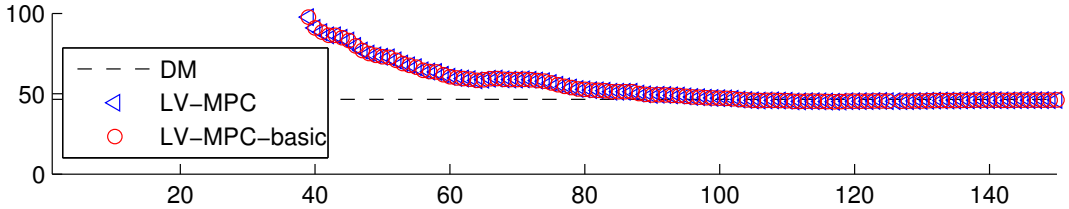
$$\boldsymbol{\Lambda} = diag([2^{n_\mathrm{f} n_\mathrm{o}}, \ldots, 2^1]).$$

The number of latent variables to include in the predictor is a decision to be made in the identification stage. Two indicators for a validation data set are evaluated for this purpose. The validation data set is obtained in the same conditions as the identification data set. The two indicators are the sum of squared prediction errors one-step ahead J_OSAPI from Equation (5.13), and multi-step ahead J_LRPI from Equation (5.12) which for a given data set can be expressed:

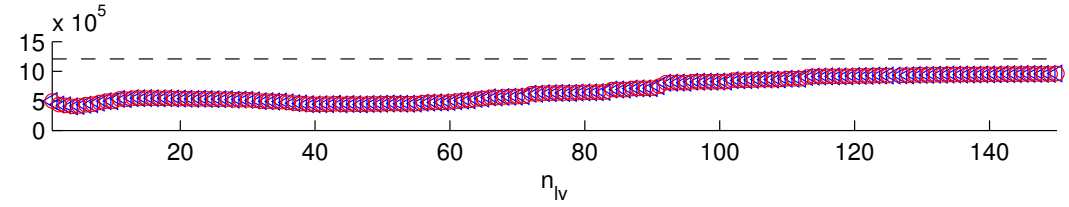
$$J_\mathrm{OSAPI} = \sum_{k=1}^{N} ||\mathbf{y}_\mathrm{k} - \hat{\mathbf{y}}_\mathrm{k}||_F^2; \quad J_\mathrm{LRPI} = \sum_{k=1}^{N} ||\mathbf{y}_\mathrm{f}(k) - \hat{\mathbf{y}}_\mathrm{f}(k)||_F^2$$

where N is the number of rows in the regression matrix. The CVs in the indicators are pre-treated as exposed above. Note J_OSAPI is the squared error of one-step ahead predictions, which is included as a term in J_LRPI, which is the squared error of multi-step ahead predictions.

These two indicators are evaluated for the model used in DM, and are swept for n_{lv} in LV-MPC[2]. Provided $\mathbf{x}(k)$ has 488 columns, n_{lv} can take any value in-between 1 and 488. Nevertheless, in DM, the MPC problem has $n_{\mathrm{u}}n_{\mathrm{i}} = 150$ d.o.f., thus there is no point in using LV-MPC with $n_{lv} > n_{\mathrm{u}}n_{\mathrm{i}}$. From J_{OSAPI} in Figure 10.5(a), one may chose $n_{lv} \geq 60$. However, from J_{LRPI} in Figure 10.5(b), multi-step ahead predictions deteriorate for large values of n_{lv}. As a trade-off between one-step ahead predictions and multi-step ahead predictions $60 < n_{lv} < 80$.



(a) J_{OSAPI}



(b) J_{LRPI}

Figure 10.5: Identification Indicators.(Note the model in LV-MPC-basic is the same as in LV-MPC, thus triangles and circles overlay.)

If $n_{lv} = 60$, the multi-step ahead predictor in LV-MPC performs better than that in DM, but one-step ahead predictions are slightly worse. If $n_{lv} = 80$, the multi-step ahead predictor in LV-MPC performs slightly worse than that obtained using $n_{lv} = 60$, but still better than that in DM; and one-step ahead predictions are as good as those in DM. Any n_{lv} in-between 60 and 80 seems reasonable.

[2]Note the model in LV-MPC-basic is the same as in LV-MPC, the difference between these two methodologies is in the form to compute the matrices of the controller.

10.1.4. Control results

To compare the three control methodologies in terms of control performance, the following indicators are evaluated for the three controllers to a change in the process set point:

- Hessian conditioning
- MSTE, mean squared tracking error
- MSDU, mean squared increments in the control action
- Number of floats to define the matrices of the controller
- Mean computation time to solve the QP with constraints.

In LV-MPC there are two additional tuning parameters, α and β (see Section 9.3). What is important is the ratio between those two parameters, thus α is set to 1, and β is given 26 values: $\beta = 10^i, \forall i = \{-5, -4, \ldots, 20\}$. The value of β which provides the best conditioned Hessian matrix is eventually used. Note this sweep of β is implemented off-line; given the model of the process, $\mathbf{H}$ is obtained for those 26 values of β, and the best conditioned $\mathbf{H}$ is eventually used. An example of this search for $n_{lv} = 60$ is provided in Figure 10.6. As shown in Figure 10.6, a good conditioning for $\beta \in [10^5, 10^7]$ is obtained. Conditioning in DM in logarithmic scale is around 10, thus for $10^2 < \beta < 10^15$, LV-MPC for $n_{lv} = 60$ outperforms DM in terms of Hessian conditioning. Note the search for β is performed off-line and no fine search is needed.

The control performance indicators sweeping[3] n_{lv} from 1 to 150 are shown in Figure 10.7:

[3]Note there is no n_{lv} in DM, thus the value of the indicators in DM is the same for any value of n_{lv}.

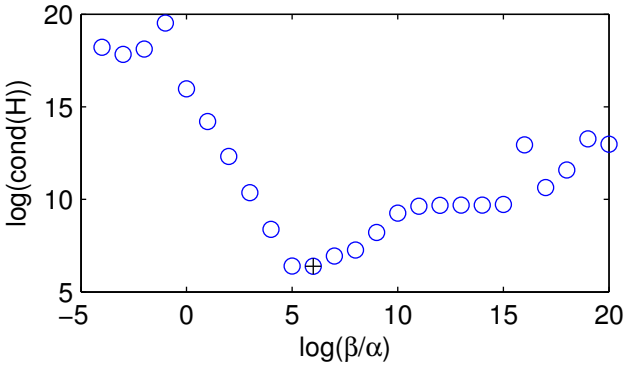
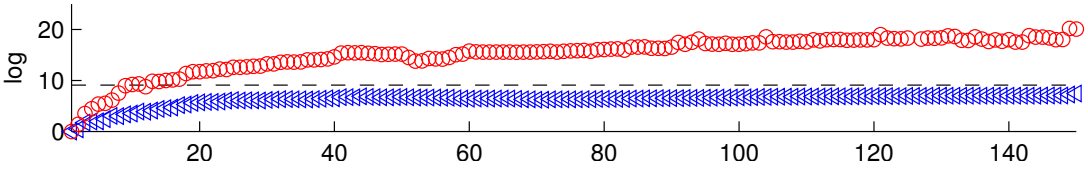
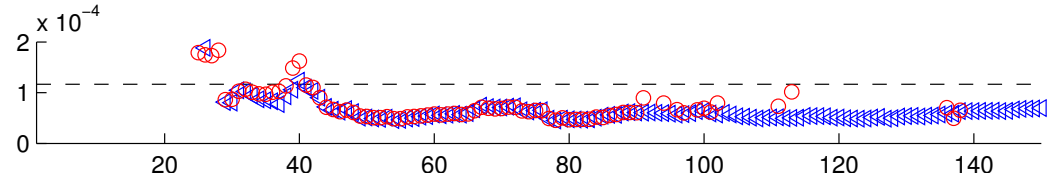


Figure 10.6: Tuning α and β

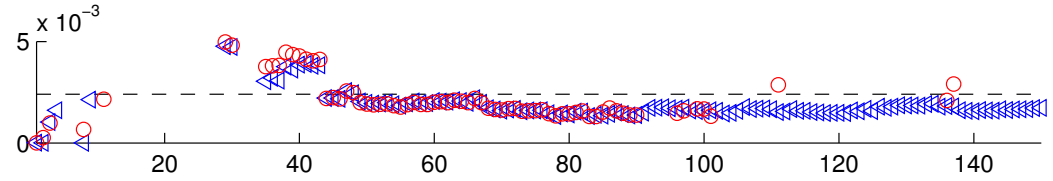
- Figure 10.7(a): Hessian conditioning in LV-MPC-basic is much larger than that in LV-MPC and DM. Hessian conditioning in LV-MPC is slightly smaller to that in DM.
- Figure 10.7(b): LV-MPC outperforms DM in terms of MSTE for $n_{\mathrm{lv}} > 45$. In LV-MPC-basic however, for $n_{\mathrm{lv}} > 90$, larger values of MSTE are obtained due to ill-conditioning.
- Figure 10.7(c): LV-MPC is equivalent to DM in terms of MSDU for $n_{\mathrm{lv}} > 45$. In LV-MPC-basic however, for $n_{\mathrm{lv}} > 90$, larger values of MSDU are obtained due to ill-conditioning.
- Figure 10.7(d): the number of floats is the same in LV-MPC and LV-MPC-basic. Both LV-MPC approaches outperform DM.
- Figure 10.7(e): LV-MPC outperforms DM in terms of computing time for $25 < n_{lv} < 100$. LV-MPC-basic needs far more time than LV-MPC and DM for $n_{lv} > 90$ due to ill-conditioning.



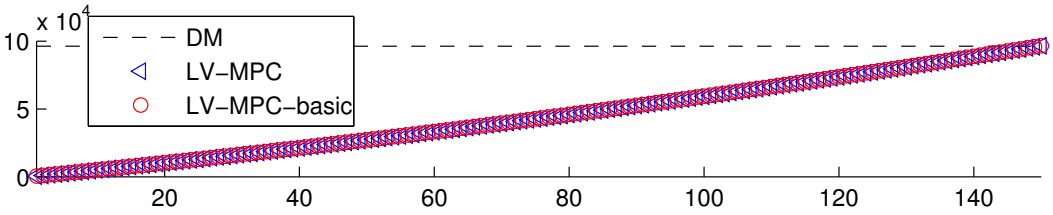
(a) Hessian conditioning



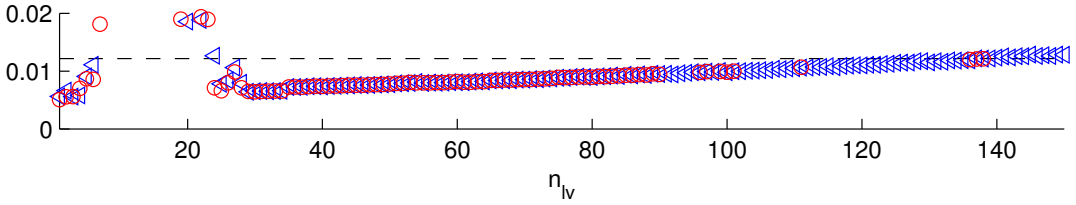
(b) MSTE



(c) MSDU



(d) Number of floats



(e) Computation time

Figure 10.7: Control indicators

Summing up, LV-MPC is preferable to LV-MPC-basic because:

- The Hessian matrix in LV-MPC is far better conditioned than that in LV-MPC-basic.
- The performance of both methodologies is equivalent whenever LV-MPC-basic is not ill-conditioned.
- Tuning the additional parameter β in LV-MPC is systematic, inexpensive, and is done off-line.

The MVs and CVs in the control experiment using DM, LV-MPC for $n_{lv} = 60$, and LV-MPC for $n_{lv} = 80$, are shown in Figure 10.8. Provided the two CVs are correlated, a change in set point in y_1 affects tracking of y_2 and vice versa. Such interaction is reduced when using the LV-MPC controller, either for $n_{lv} = 60$ or $n_{lv} = 80$. From Figure 10.8, a better control result is obtained for $n_{\mathrm{lv}} = 80$. From Figure 10.5, the predictor for $n_{\mathrm{lv}} = 80$ performs better than that using $n_{\mathrm{lv}} = 60$ at one-step ahead predictions, but worse at multi-step ahead predictions. Hence, it makes sense to use $\mathbf{\Lambda}$ in PLS to improve predictions in the near horizon for a reduced n_{lv}.

Monitoring of $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ during closed-loop operation for LV-MPC and $n_{lv} = 80$ is shown in Figure 10.9. $\bar{J}_{\mathrm{t}}$ is plot in continuous line, $\bar{J}_{\mathrm{e}}$ in dash-dotted line, and a dashed-line has been plot for a value of 1. Note 95% of the identification samples provided $\bar{J}_{\mathrm{t}} \leq 1$ and $\bar{J}_{\mathrm{e}} \leq 1$. During control-loop monitoring (Figure 10.9) most observations provide $\bar{J}_{\mathrm{t}} \leq 1$ and $\bar{J}_{\mathrm{e}} \leq 1$, hence the model is not beeing used in extrapolation mode.

Table 10.1 shows improvements of LV-MPC versus DM for $n_{lv} = 60$ and $n_{lv} = 80$. If n_{lv} is tuned as said in the identification stage, one would choose a value in-between 60 and 80. For $60 < n_{lv} < 80$, LV-MPC outperforms DM in terms of MSTE in about 55%, the size of the matrices of the controller in about 60% and computation time in about 30%.

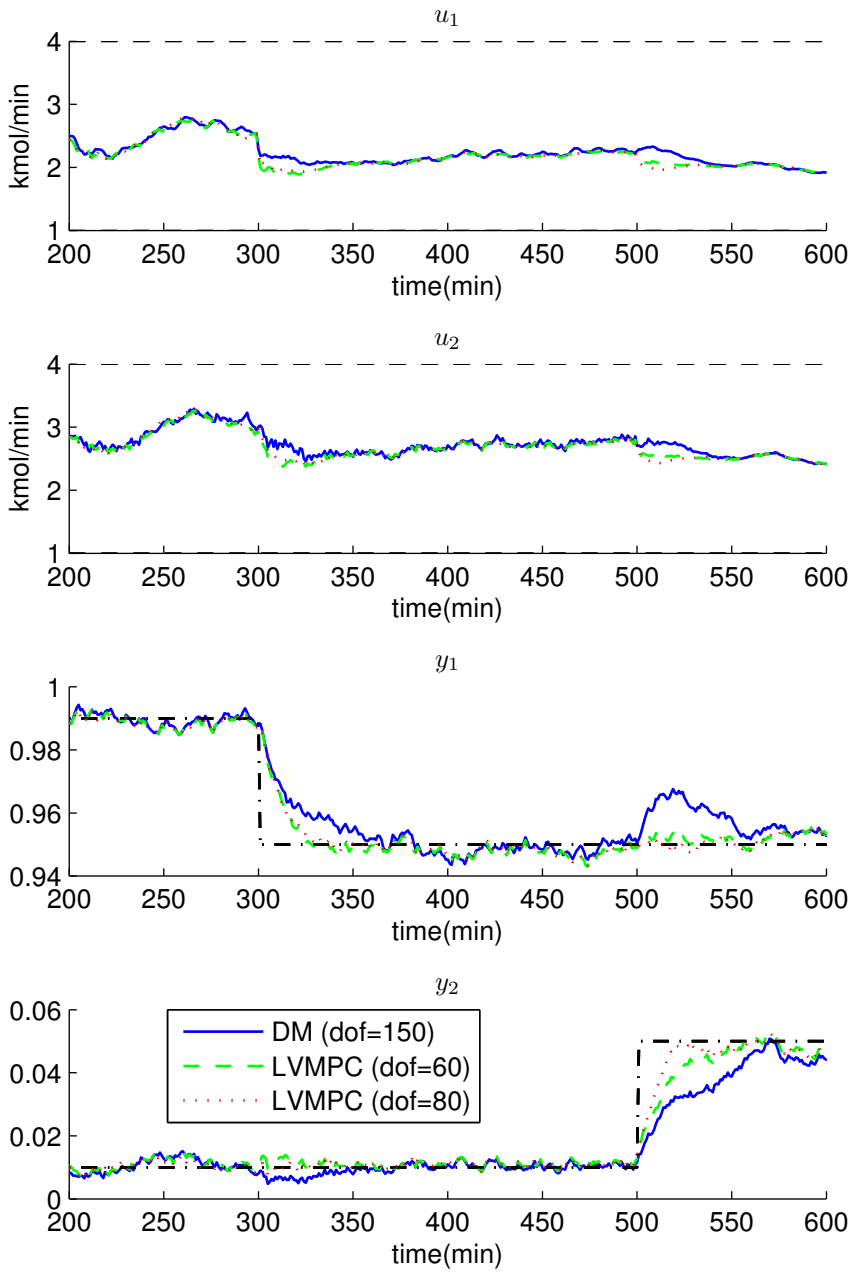


Figure 10.8: Time response of the controlled process

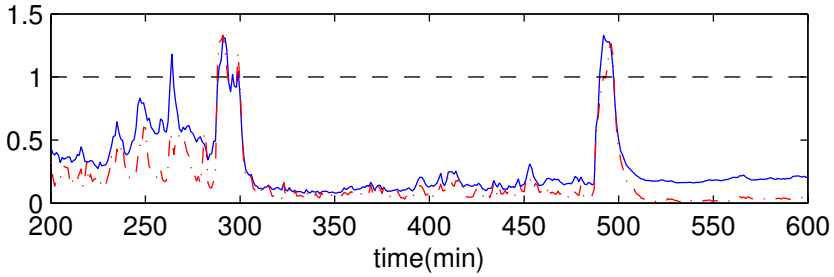


Figure 10.9: $\bar{J}_{\mathrm{t}}$, continuous line; and $\bar{J}_{\mathrm{e}}$, dash-dotted line.

	LV-MPC($n_{lv} = 60$)	LV-MPC($n_{lv} = 80$)
MSTE	51%	59%
Floats	65%	52%
Computation	33%	26%

Table 10.1: Improvement vs DM

10.2. Control of a Twin-rotor MIMO model

In this section a 2x2 MIMO simulink non-linear model is defined, identified, and controlled by means of two control methodologies:

DM Traditional data-driven MPC approach.

LV-MPC The methodology proposed in this part accounting for Hessian conditioning, offset-free tracking, and validity of the model. Two different strategies to ensure model validity are implemented and compared.

This example aims at comparing performance of these two control methodologies both in normal operation and in case of measurement error. The controllers are tuned so that a fast closed-loop response is obtained, then the controllers are more sensitive to errors.

10.2.1. Process description

The process to control is the non-linear simulink model of the Twin rotor MIMO system from Feedback. The non-linear model can be downloaded at http://sergarro.webs.upv.es/TRMSDown/TRMSDown.html. The variables in the process and their working points are:

- MVs:
  - u_1: Main rotor; range $[-1 \quad 1]$; operating point 0
  - u_2: Tail rotor; range $[-1 \quad 1]$; operating point 0
- CVs:
  - y_1: Vertical angle; range $[-\pi \quad \pi]$; operating point 0 (rad)
  - y_2: Horizontal angle; range $[-\pi \quad \pi]$; operating point 0 (rad)

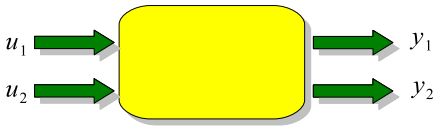


Figure 10.10: Block diagram of the TRMS.

10.2.2. Control parameters

In this Section some parameters of the controller are decided. First, the sampling time is defined from the step response of the process to a step of 0.1 amplitude applied simultaneously to both inputs, see Figure 10.11. The steepest slope takes about 1 second, and T_s is set so that 10 samples are taken during the fastest slope of the process, then $T_s = 0.1$ *sec*.

It is convenient in MPC to choose n_u and n_f such that $n_\mathrm{f} - n_\mathrm{u}$ is greater than or equal to the process settling time. As shown in Figure 10.11, settling time

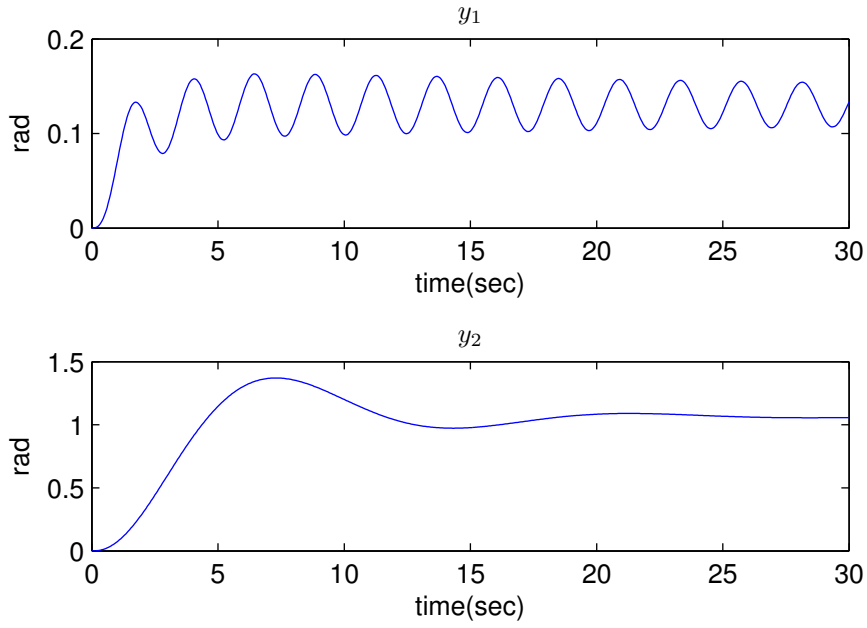


Figure 10.11: Step response of the process

for output y_1 is greater than 30 seconds, which would yield a considerably large prediction horizon. Provided the non-linear process is approximated with a linear model, there is process-model mismatch, hence large prediction horizons yield bad multi-step ahead predictions which in turn provide poor closed-loop performance. In this example $n_\mathrm{f} - n_\mathrm{u}$ is set to 20 seconds as a trade-off between accounting for most of the dynamic behaviour and having a reliable predictor. Provided $T_s = 0.1$

$$n_\mathrm{f} = n_\mathrm{u} + 200$$

According to the tuning directrices provided in Section 9.7,

$$n_\mathrm{u} = \frac{n_\mathrm{f} - n_\mathrm{u}}{2} = 100.$$

Constraints are defined for the MVs $-1 \leq \mathbf{u}_{\mathrm{k}} \leq 1$. Future references are assumed unknown,

$$\mathbf{r}_{\mathrm{k+i}} = \mathbf{r}_{\mathrm{k+1}}, \quad \forall i \in [1, n_{\mathrm{f}}]$$

The weight of the control moves is set so that a fast response is obtained

$$\lambda_u = 10$$

10.2.3. Identification

The identification data set in Figure 10.12 has been obtained in open-loop. The MVs have been excited with low-pass filtered white noise

$$F(z) = \frac{0.05}{1 - 0.95z^{-1}}$$

and the amplitude of u_1 doubles the amplitude of u_2 so that y_1 can be better identified.

To decide the order of the model, n_a and n_b are swept and two indicators are evaluated for a validation data set using the DM model[4]. The two indicators are the sum of squared prediction errors one-step ahead $\bar{J}_{\mathrm{OSAPI}}$ from Equation (5.13), and multi-step ahead $\bar{J}_{\mathrm{LRPI}}$ from Equation (5.12), both of them normalized to the number of quadratic terms to sum:

$$\bar{J}_{\mathrm{OSAPI}} = \frac{1}{N n_{\mathrm{o}}} \sum_{k=1}^{N} ||\mathbf{y}_{\mathrm{k}} - \hat{\mathbf{y}}_{\mathrm{k}}||_F^2; \quad \bar{J}_{\mathrm{LRPI}} = \frac{1}{N n_{\mathrm{o}} n_{\mathrm{f}}} \sum_{k=1}^{N} ||\mathbf{y}_{\mathrm{f}}(k) - \hat{\mathbf{y}}_{\mathrm{f}}(k)||_F^2$$

N is the number of samples in the data set. Note $\bar{J}_{\mathrm{OSAPI}}$ is an indicator on squared errors for one-step ahead predictions, and $\bar{J}_{\mathrm{LRPI}}$ is an indicator on squared errors of multi-step ahead predictions. Figure 10.13 plots the values of $\bar{J}_{\mathrm{OSAPI}}$ and $\bar{J}_{\mathrm{LRPI}}$ for different values of n, being $n_a = n$ and $n_b = n$.

[4]The validation data set is obtained in the same conditions as the identification data set.

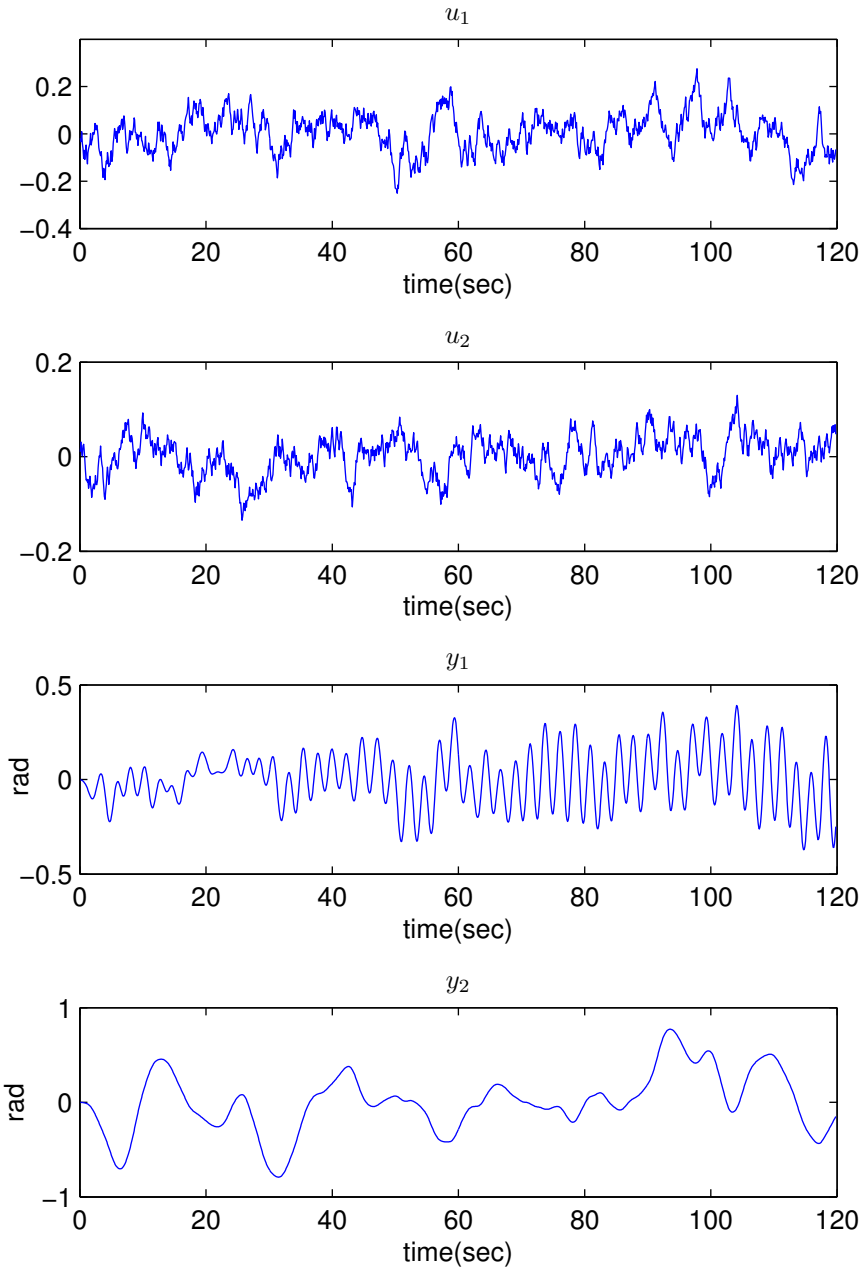
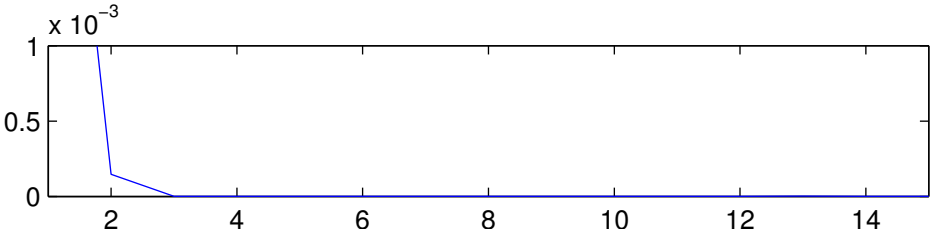
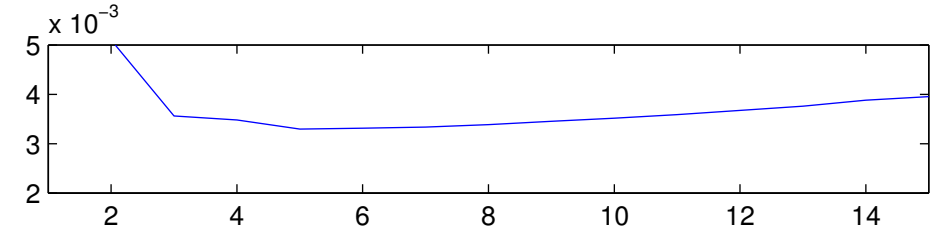


Figure 10.12: Identification data set



(a) $\bar{J}_{\text{OSAPI}}(n)$



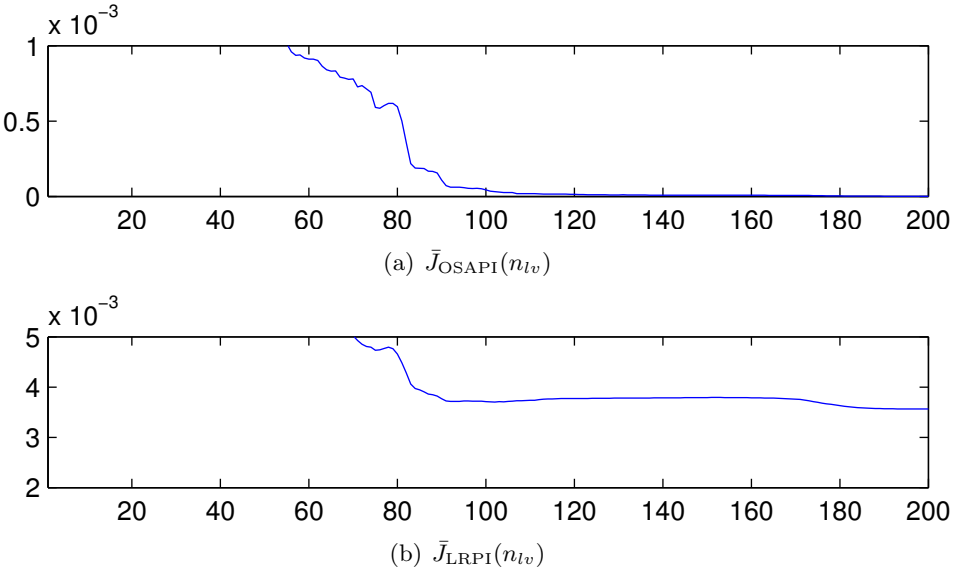
(b) $\bar{J}_{\text{LRPI}}(n)$

Figure 10.13: Identification Indicators versus n.

From Figure 10.13(a) $n \geq 3$, and from Figure 10.13(b) $n = 5$, then a fifth order linear model is used to approximate the nonlinear process: $n_a = n_b = 5$.

The number of latent variables to include in the predictor is a decision to be made in the identification stage. n_{lv} can take any value in-between 1 and the number of columns in $\mathbf{x}(k)$, which in this example yields $(n_{\text{b}} - 1)n_{\text{i}} + n_{\text{a}}n_{\text{o}} + n_{\text{f}}n_{\text{i}} = 618$ columns in $\mathbf{x}(k)$. Consequently n_{lv} can take any value in-between 1 and 618, however, the controller has $n_{\text{u}}n_{\text{i}} = 200$ d.o.f., then it makes sense to sweep n_{lv} in-between 1 and 200. As shown in Figure 10.14, $\bar{J}_{\text{OSAPI}}$ and $\bar{J}_{\text{LRPI}}$ decrease as n_{lv} increases, then in this example $n_{lv} = 200$.

Predictive performance of the two models obtained is tested performing predictions for a validation data set. Figures 10.15 and 10.16 contain one-step ahead predictions, predictions in the far prediction horizon, and the index R^2 evaluated for predictions from $k + 1$ up to $k + n_{\text{f}}$. Note the LV-MPC model contains 200 latent variables out of the 618 columns in the input vector of the model, but still performs as the DM model.



(a) $\bar{J}_{\mathrm{OSAPI}}(n_{lv})$

(b) $\bar{J}_{\mathrm{LRPI}}(n_{lv})$

Figure 10.14: Identification Indicators versus n_{lv}.

10.2.4. Control results under normal operation

The DM and LV-MPC controllers are tested under normal operation for a sequence of step changes in the set points of the CVs. Two control scenarios are considered for LV-MPC in terms of quadratic constraints to ensure validity of the model, in the first scenario $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ are considered for quadratic constraints and in the second one $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$.

As shown in Figure 10.19, the LV-MPC controller that uses $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ does not reach the references in steady state when the set point is at the limit of the identification data set 0.2 (see Figure 10.12). The reason is quadratic constraints depend on past data, which cannot be changed, and the identification data set does not contain steady operation at those points, then quadratic constraints bound the decision space in an effort to maintain the process in the region where it has been identified. Such effort prevents the LV-MPC cost function to reach 0 resulting in biased control for static points on the boundary

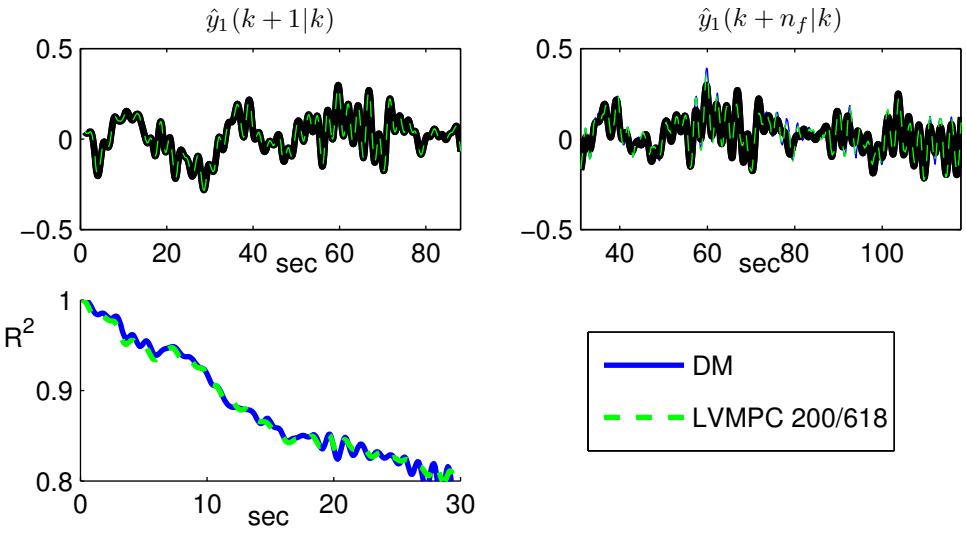


Figure 10.15: Validation results for y_1.

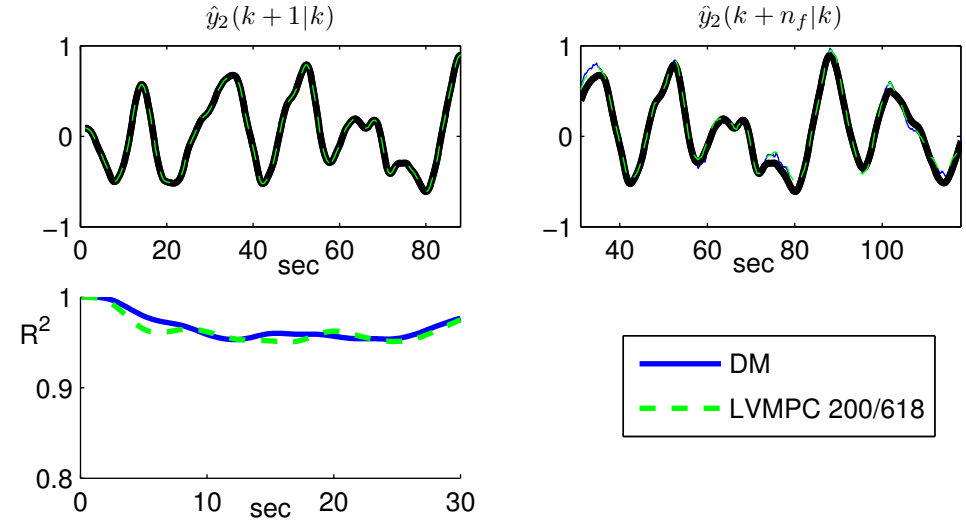


Figure 10.16: Validation results for y_2.

of the identification data set. On the contrary, LV-MPC that uses $\check{J}_t$ and $\check{J}_e$ for the quadratic constraints can reach the references at steady state because it neglects past data. $\check{J}_t$ and $\check{J}_e$ neglect the past, then only constrain the control moves to be in a valid range compared to those used for identification, then these quadratic constraints have an influence on the closed-loop dynamic response, but eventually the process will reach the point if it can be reached. The value of $\check{J}_t$ and $\check{J}_e$ obtained for the control experiment is plot in Figure 10.17, it can be seen from the Figure that quadratic constraints hold since $\check{J}_t \leq 1$ and $\check{J}_e \leq 1$ for the whole experiment. Figure 10.18 plots the number of iterations needed to linearise quadratic constraints.

The DM controller presents similar settling time to that obtained with the LV-MPC controllers, but a larger overshoot. This can be an argument to chose LV-MPC instead of DM in this case, however, overall the three controllers provide similar closed-loop response.

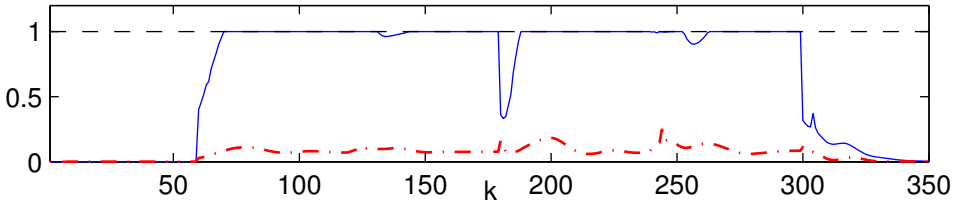


Figure 10.17: $\check{J}_t$, continuos blue line; $\check{J}_e$, dash-dotted red line.

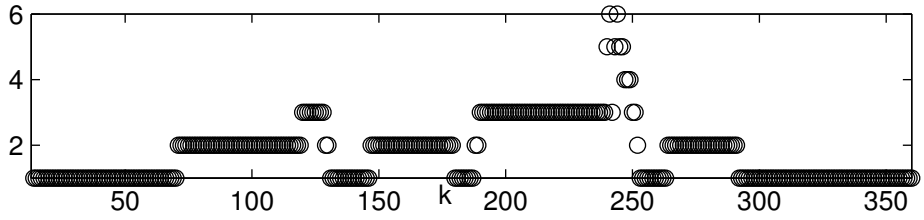


Figure 10.18: Number of iterations to linearise quadratic constraints.

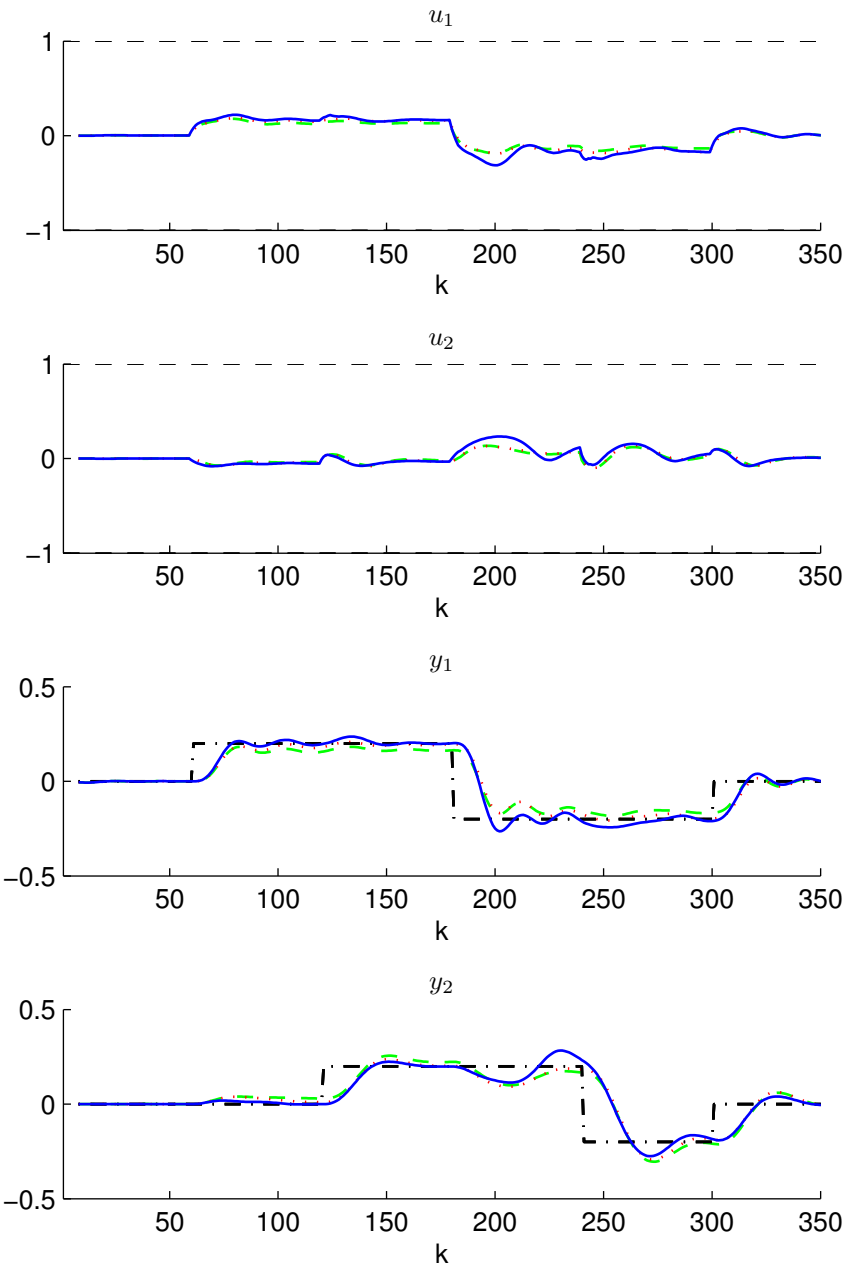


Figure 10.19: Closed-loop time response in normal operation. DM, continuous blue line; LV-MPC evaluating $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$, dashed green line; LV-MPC evaluating $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$,dotted red line.

10.2.5. Control results in the event of a measurement error

The DM and LV-MPC controllers are tested controlling the process to a sequence of step changes in the set points of the CVs. The measurement of the CVs is assumed not to be available for $k = [90, 91, 92]$ and they are assumed to be equal to the measurements at $k = 89$. This measurement error has not much effect in case of loose control, but in this example $\lambda_u = 10$ which yields a settling time of about 2 seconds then the control is tight. In LV-MPC quadratic constraints to ensure validity of the model are formulated in terms of $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$.

Figure 10.22 plots the closed-loop time response of both control strategies. As seen in the Figure, DM cannot cope with the measurement error whilst LV-MPC remains almost unaltered versus the control without measurement error in Figure 10.19. There are two reasons that justify the more robust behaviour of LV-MPC versus DM:

- **Decide in the latent variable space**: LV-MPC performs the minimization in the latent variable space and projecting the error to the latent variable space can reduce its effect. In fact, latent variable methods emerged as a solution to identify processes with short and fat matrices of data in which correlation, lack of data, and measurement error makes identification in the original space impossible. In latent variable methods, data is projected onto the latent variable space in which identification can be successfully performed. LV-MPC deals with making the decision of the control moves in a time window, but such decision is taken in the inner, and more robust, latent variable space.
- **Ensure model validity**: In the event of a failure, LV-MPC with quadratic constraints will make decisions trying to ensure model validity, then avoiding too aggressive moves which can lead to instability. If the process remains in a safe region during the failure, the controller will be able to lead the process back to track the set points.

The value of $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$ obtained for the control experiment is plot in Figure 10.20, the figure shows that quadratic constraints hold for the whole experiment. Figure 10.21 plots the number of iterations needed to linearise quadratic constraints. When the error happens, $k = 90$, the number of iterations reaches its maximum. Comparing the number of iterations around $k = 90$ between the two control scenarios, Figures 10.18 and 10.21, the number of iterations around $k = 90$ to satisfy quadratic constraints increases from 3 to 7 due to the measurement error.

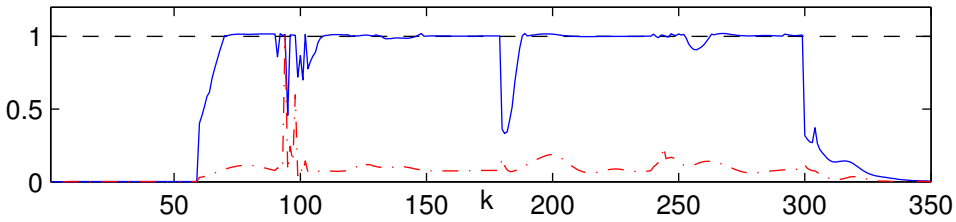


Figure 10.20: $\check{J}_{\mathrm{t}}$, continuos blue line; $\check{J}_{\mathrm{e}}$, dash-dotted red line.

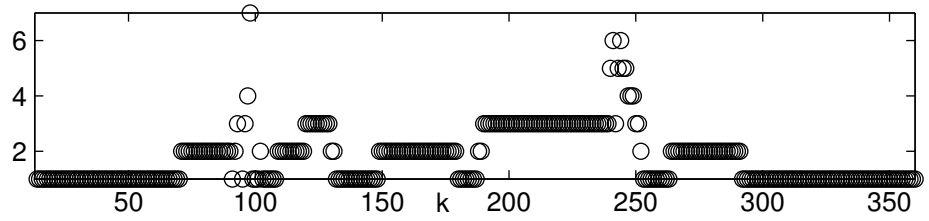


Figure 10.21: Number of iterations to linearise quadratic constraints.

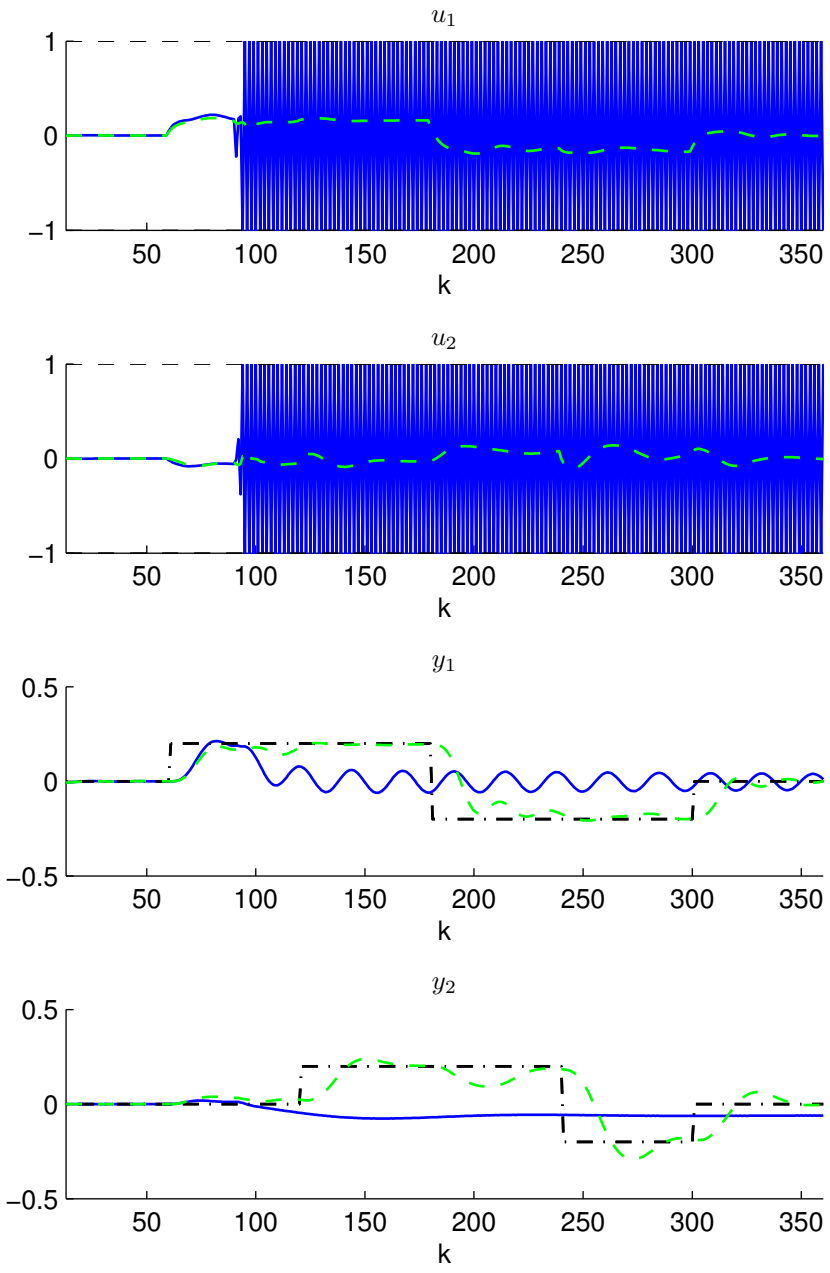


Figure 10.22: Closed-loop time response in the event of measurement error. DM, continuous blue line; LV-MPC evaluating $\check{J}_\mathrm{t}$ and $\check{J}_\mathrm{e}$, dashed green line.

10.3. Control of a Twin-rotor MIMO model with perturbation

10.3.1. Process description

The process is the same used in Section 10.2 adding filtered white noise to the CVs as a perturbation. The low-pass filter to obtain the perturbation is

$$F(z) = \frac{0.05}{1 - 0.95z^{-1}}$$

and the range of the resulting perturbation is 10% the range of the closed-loop experiment used to compare the different control strategies [±0.02].

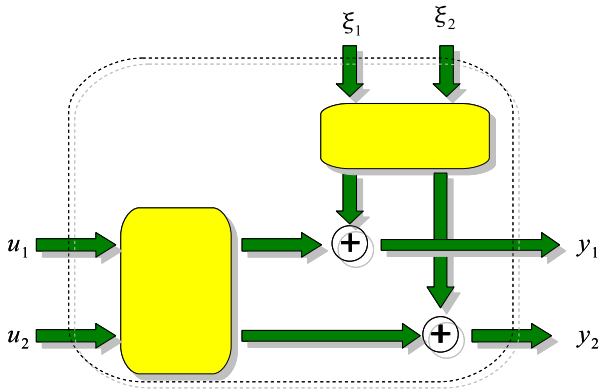


Figure 10.23: Block diagram of the TRMS with the added perturbation.

10.3.2. Control parameters

T_s, n_f, and n_u take the same values used in Subsection 10.2.2. Provided there is a perturbation in the process λ_u is increased compared to that in Subsection 10.2.2.

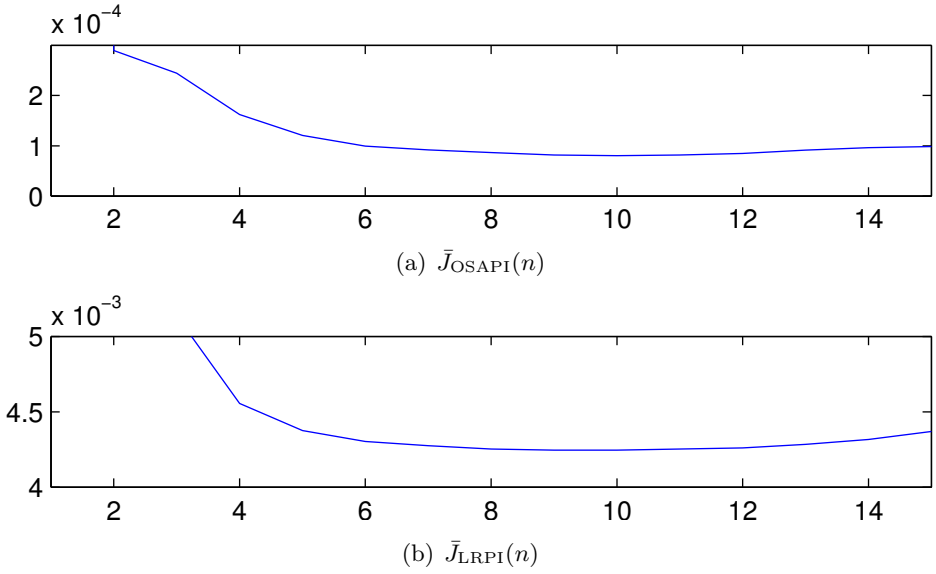
$$\lambda_u = 100.$$

For $\lambda_u = 10$, LV-MPC provides similar closed-loop response to that obtained for $\lambda_u = 100$; closed-loop response in DM however, deteriorates considerably for $\lambda_u = 10$. Then, in the sake of comparability among the two control strategies, λ_u is set to 100 in this example.

10.3.3. Identification

The identification data set is generated adding the perturbation to the CVs in the data set in Figure 10.12. Figure 10.24 plots the values of $\bar{J}_{\text{OSAPI}}$ and $\bar{J}_{\text{LRPI}}$ for different values of n, being $n_a = n$ and $n_b = n$. The minimum of both indicators happens for

$$n_a = n_b = 10.$$

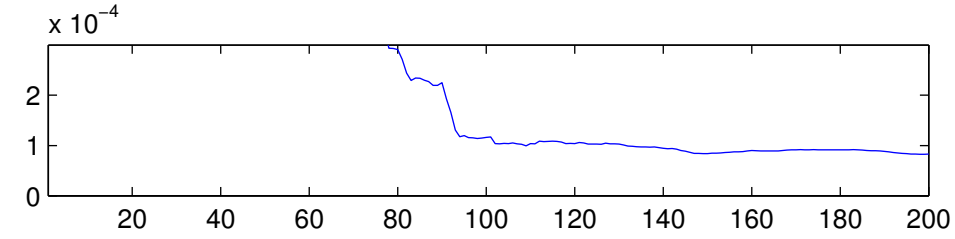


(a) $\bar{J}_{\text{OSAPI}}(n)$

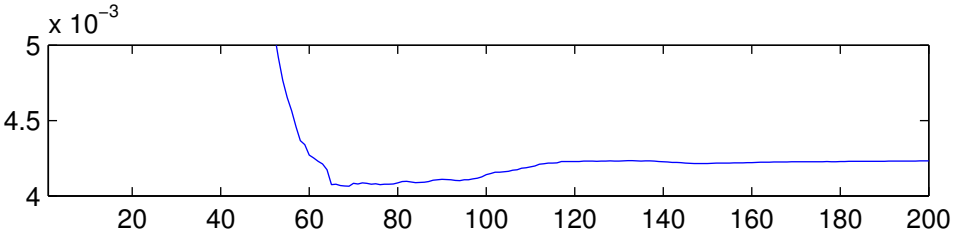
(b) $\bar{J}_{\text{LRPI}}(n)$

Figure 10.24: Identification Indicators versus n.

Figure 10.25 plots the values of $\bar{J}_{\text{OSAPI}}$ and $\bar{J}_{\text{LRPI}}$ for different values of n_{lv}. From Figure 10.25(a) $n_{lv} \geq 90$, and from Figure 10.25(b) $70 \leq n_{lv} \leq 100$. In this example n_{lv} is set to 90.



(a) $\bar{J}_{\text{OSAPI}}(n_{lv})$



(b) $\bar{J}_{\text{LRPI}}(n_{lv})$

Figure 10.25: Identification Indicators versus n_{lv}.

Figures 10.26 and 10.27 contain one-step ahead predictions, predictions in the far prediction horizon, and the index R^2 evaluated for predictions from $k+1$ up to $k+n_{\text{f}}$. Note the LV-MPC model contains 90 latent variables out of the 638 columns in the input vector of the model, but still performs as the DM model. Comparing this validation results with those obtained in the previous example in which there was no perturbation, Figures 10.15 and 10.16, one can see the predictor performs slightly worse for $\hat{y}_1$, whereas almost does not change for $\hat{y}_2$.

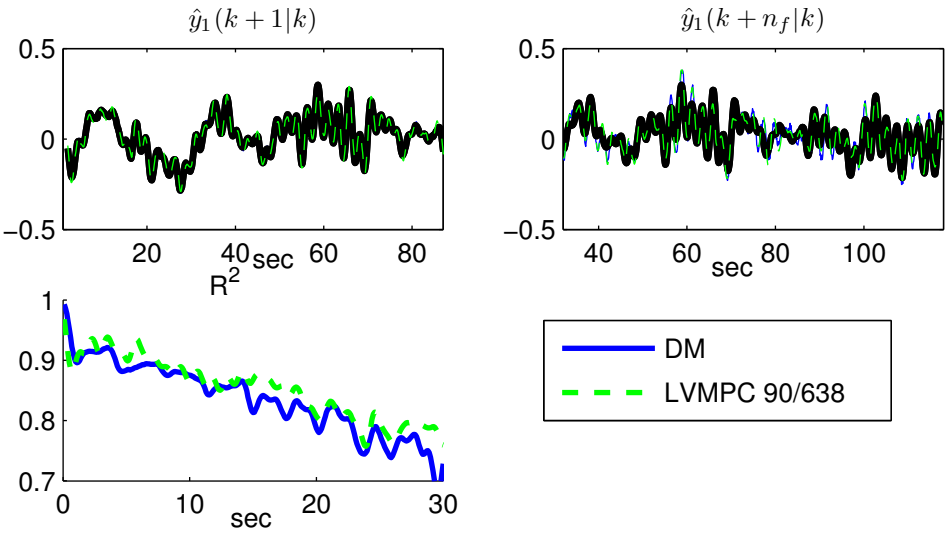


Figure 10.26: Validation results for y_1.

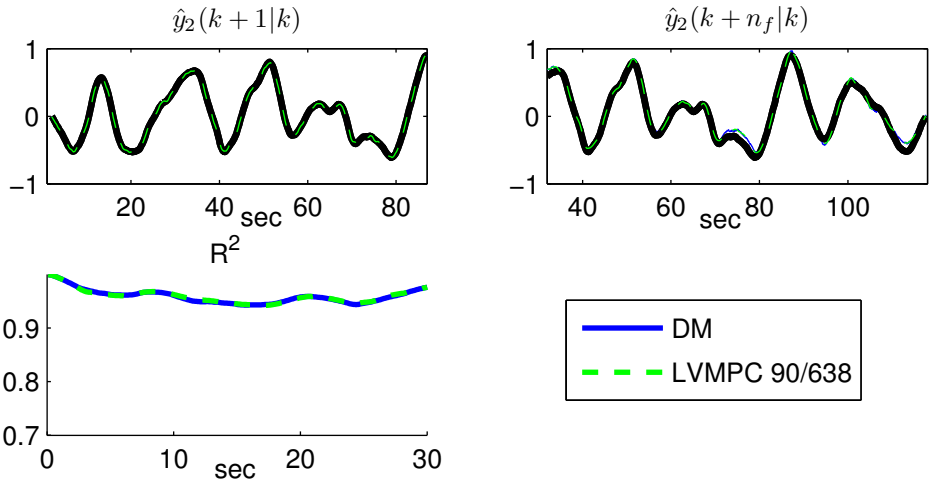


Figure 10.27: Validation results for y_2.

10.3.4. Control results

The DM and LV-MPC controllers are compared. Constraints on $\check{J}_t$ and $\check{J}_e$ are set to ensure model validity in LV-MPC. The closed-loop time response in the event of perturbation is plot in Figure 10.30. As shown in the figure, LV-MPC outperforms DM in the event of additive perturbation to the CVs.

The value of $\check{J}_t$ and $\check{J}_e$ obtained for the control experiment is plot in Figure 10.28, it can be seen from the figure that quadratic constraints hold since $\check{J}_t \leq 1$ and $\check{J}_e \leq 1$ for the whole experiment. Figure 10.29 plots the number of iterations needed to linearise quadratic constraints.

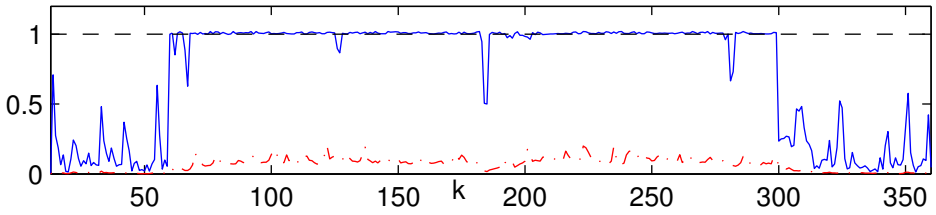


Figure 10.28: $\check{J}_t$, continuos blue line; $\check{J}_e$, dash-dotted red line.

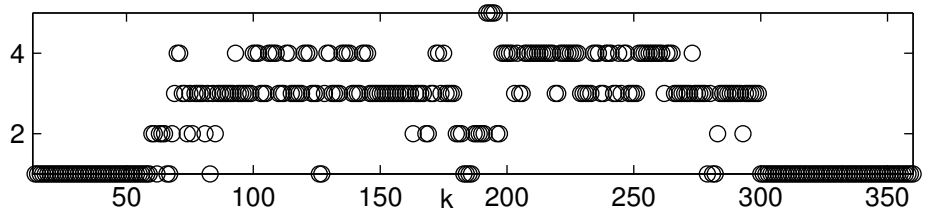


Figure 10.29: Number of iterations to linearise quadratic constraints.

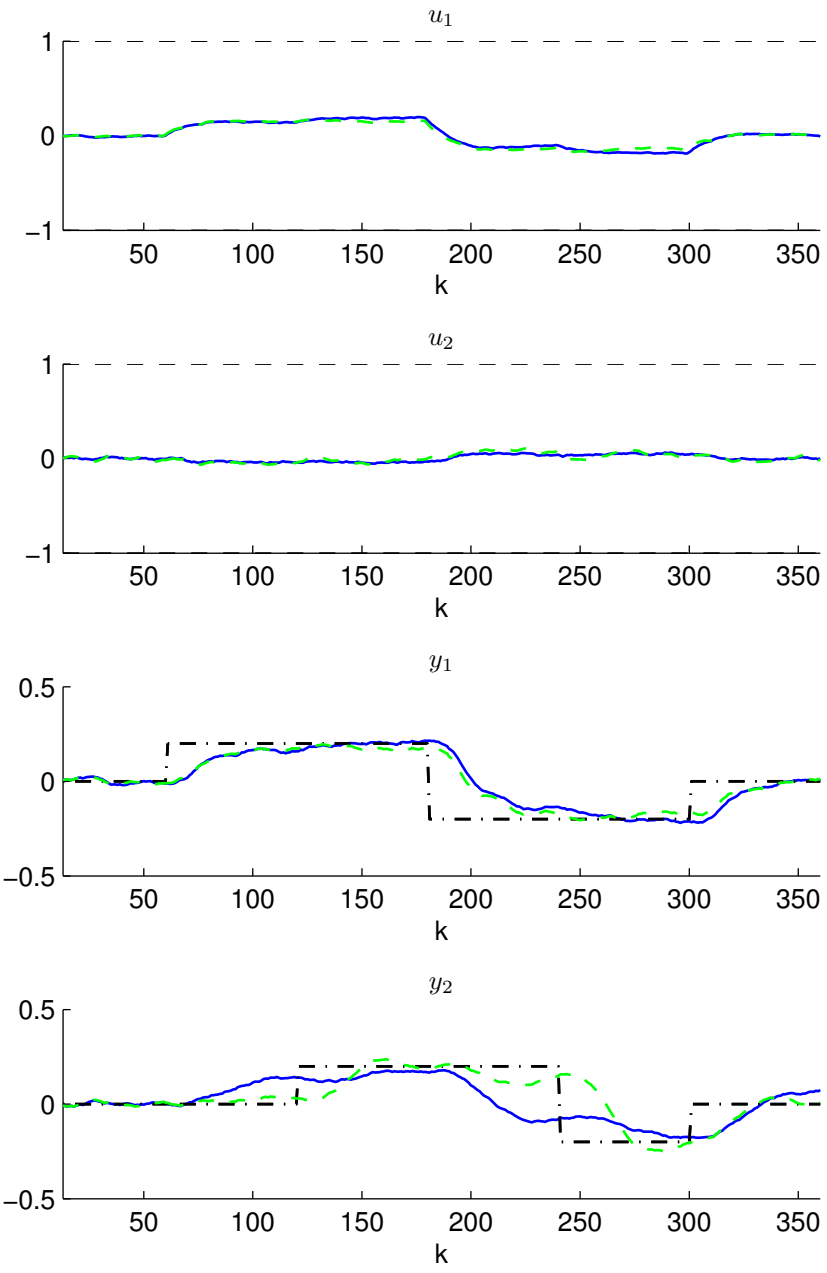


Figure 10.30: Closed-loop time response in the event of perturbation. DM, continuous blue line; LV-MPC, dashed green line.

10.4. Control of a Boiler

In this section the model of a boiler is controlled by means of

DM Traditional data-driven MPC approach with no validity indicators

LV-MPC Latent variable MPC with no validity indicators

LV-MPC-cons LV-MPC with constraints on $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ to ensure validity of predictions.

LV-MPC-cons-neg LV-MPC with constraints on $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$ to ensure validity of predictions.

In this example first a description of the process is provided; second control parameters are set; third the predictor is obtained from data; and finally two control scenarios are considered: normal operation, and large changes in set points and perturbance.

10.4.1. Process description

The process to control is the simulink model of a boiler. The model can be downloaded at http://www.dia.uned.es/~fmorilla/benchmarkPID2012/.

The variables in the process and their working points are:

- MVs:
  - u_1: Fuel flow; range [0 100]; operating point 35.21%
  - u_2: Water flow; range [0 100]; operating point 57.57%

- Perturbation:
  - m_1: load level; range [0 100]; operating point 46.36%
- CVs:
  - y_1: Steam pressure; range [0 100]; operating point 60%
  - y_2: Water level; range [0 100]; operating point 50%

10.4.2. Control parameters

The sampling time is defined from the step response of the process to a step of 10% in the inputs (Figures 10.31 and 10.32) and perturbance (Figure 10.33). From Figure 10.31, the steepest slope takes about 50 seconds, and T_s is set so that 10 samples are taken during the fastest slope of the process, then $T_s = 5\ sec.$

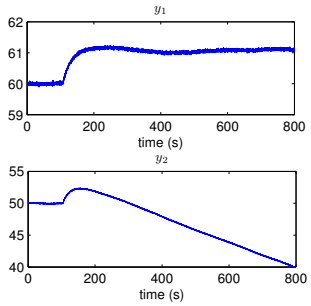


Figure 10.31: Step u_1

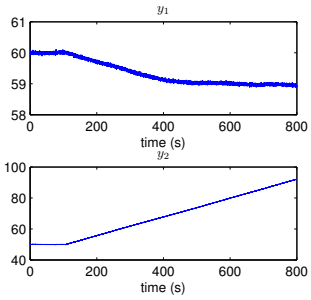


Figure 10.32: Step u_2

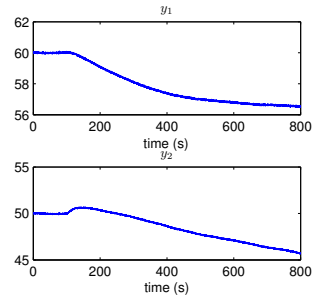


Figure 10.33: Step m_1

It is convenient in MPC to choose n_u and n_f such that $n_\mathrm{f} - n_\mathrm{u}$ is greater than or equal to the process settling time towards changes in the MVs. From Figure 10.32, settling time for output y_1 is about 300 seconds. y_2 has an integrator hence it does not settle, however it moves at a constant rate 100 seconds after the step. Hence $n_\mathrm{f} - n_\mathrm{u}$ is set to 300sec, and for $T_s = 5$sec

$$n_\mathrm{f} - n_\mathrm{u} = \frac{300}{5} = 60\ samples.$$

As a rule of thumb, n_{u} can be set as half the value set for $n_{\mathrm{f}} - n_{\mathrm{u}}$, which in this case would yield $n_{\mathrm{u}} = 30$. However, from Figure 10.33, the settling time of y_1 to a change in m_1 is about 600 seconds, then n_{f} should be at least 600 seconds:

$$n_{\mathrm{f}} = \frac{600}{5} = 120\, samples$$

then

$$n_{\mathrm{f}} - n_{\mathrm{u}} = 60 \;\Rightarrow n_{\mathrm{u}} = n_{\mathrm{f}} - 60 = 120 - 60 = 60\, samples.$$

Constraints are defined for the MVs and their rate:

- $0 \leq u_{\mathrm{i}} \leq 100, \quad \forall i \in [1, 2]$
- $|\Delta u_{\mathrm{i}}| \leq 1\%/sec, \quad \forall i \in [1, 2]$

Future references are assumed unknown,

$$\mathbf{r}_{\mathrm{k+i}} = \mathbf{r}_{\mathrm{k+1}}, \quad \forall i \in [1, n_{\mathrm{f}}]$$

The weight of the control moves is set so that a fast response is obtained

$$\lambda_u = 1$$

10.4.3. Identification

The identification and validation data sets in Figure 10.34 are obtained in closed-loop. The continuous blue plots represent the identification data set, and the discontinuous green plots represent the validation data set. To obtain the identification and validation data sets the process is controlled using two PID controllers. The set points of the CVs are moved around the working point; steps of 10% amplitude are added to m_1; and steps of 20% amplitude are added to the MVs.

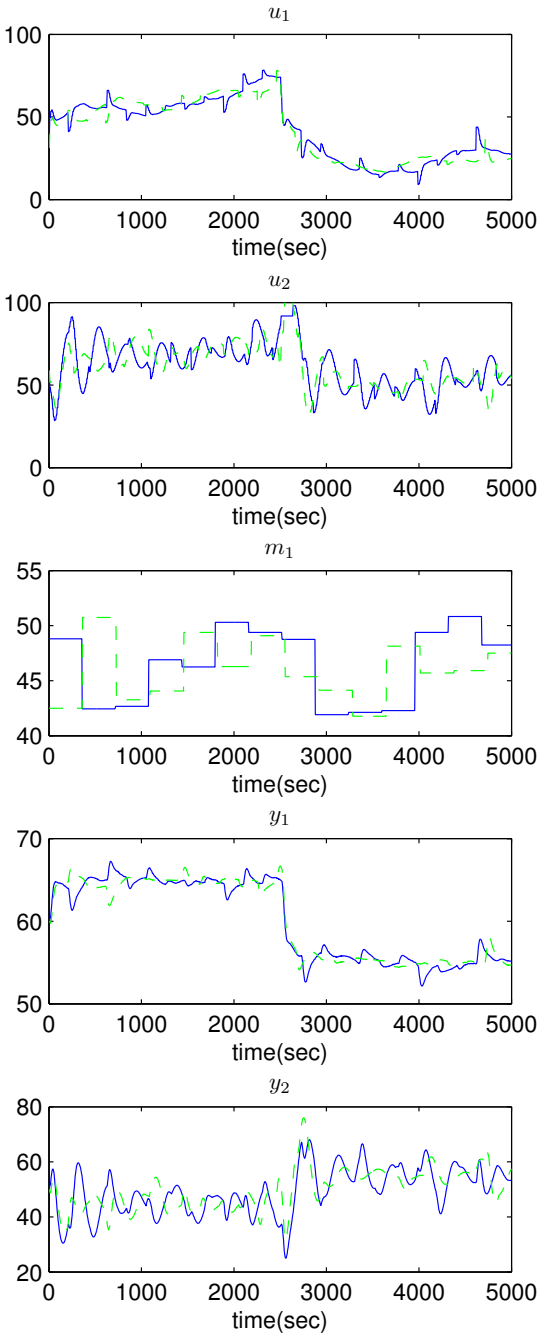
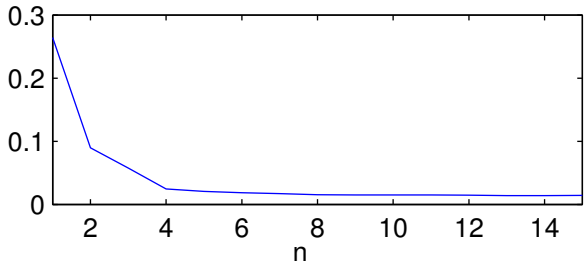


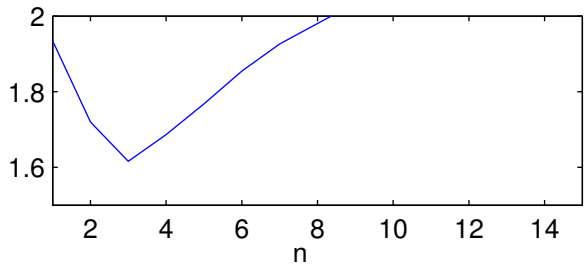
Figure 10.34: Identification and validation data sets

Prior to identification, the set point is removed form the MVs and the CVs. m_1 is not considered in the model. To decide the order of the model, n_a and n_b are swept and two indicators are evaluated for the validation data set. The two indicators are the sum of squared prediction errors one-step ahead $\bar{J}_{\text{OSAPI}}$ from Equation (5.13), and multi-step ahead $\bar{J}_{\text{LRPI}}$ from Equation (5.12), both of them normalized to the number of quadratic terms to sum.

Figure 10.35 plots the values of $\bar{J}_{\text{OSAPI}}$ and $\bar{J}_{\text{LRPI}}$ for different values of n, being $n_a = n$ and $n_b = n$. From Figure 10.35(a) $n \geq 4$, and from Figure 10.35(b) $2 \leq n \leq 4$, then a fourth order linear model is used to approximate the nonlinear process: $n_a = n_b = 4$.



(a) $\bar{J}_{\text{OSAPI}}(n)$



(b) $\bar{J}_{\text{LRPI}}(n)$

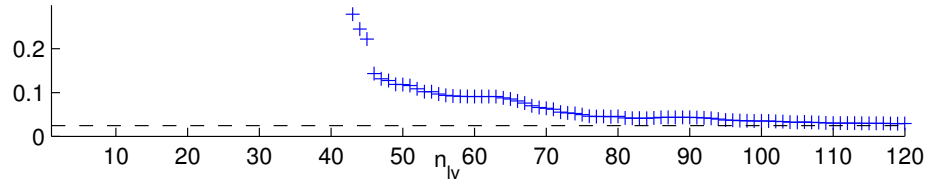
Figure 10.35: Identification Indicators versus n.

The number of latent variables to include in the predictor is a decision to be made in the identification stage. n_{lv} can take any value in-between 1 and the

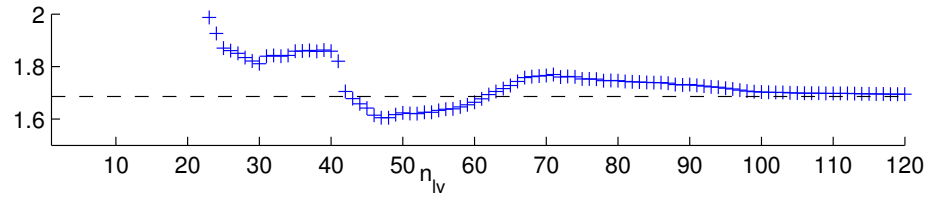
number of columns in $\mathbf{x}(k)$, which in this example yields $(n_\mathrm{b}-1)n_\mathrm{i}+n_\mathrm{a}n_\mathrm{o}+n_\mathrm{f}n_\mathrm{i}=254$ columns in $\mathbf{x}(k)$. Consequently n_{lv} can take any value in-between 1 and 254, however, the controller has $n_\mathrm{u}n_\mathrm{i}=120$ d.o.f., then it makes sense to sweep n_{lv} in-between 1 and 120.

Figure 10.36 plots in continuous blue line $\bar{J}_{\mathrm{OSAPI}}(n)$ and $\bar{J}_{\mathrm{LRPI}}(n)$ for the LV-MPC model, and in discontinuous black line $\bar{J}_{\mathrm{OSAPI}}(n)$ and $\bar{J}_{\mathrm{LRPI}}(n)$ for the DM model. From Figure 10.36(a), $n_\mathrm{lv}\geq 80$. From Figure 10.36(b), $40\leq n_\mathrm{lv}\leq 60$ or $n_\mathrm{lv}\geq 100$. Then

$$n_\mathrm{lv}=100.$$



(a) $\bar{J}_{\mathrm{OSAPI}}(n_{lv})$



(b) $\bar{J}_{\mathrm{LRPI}}(n_{lv})$

Figure 10.36: Identification Indicators versus n_{lv}.

Predictive performance of the two models obtained is tested performing predictions for the validation data set. Figures 10.37 and 10.38 contain one-step ahead predictions, predictions in the far prediction horizon, and the index R^2 evaluated for predictions from $k+1$ up to $k+n_\mathrm{f}$. Note the LV-MPC model contains 100 latent variables out of the 254 columns in the input vector of

the model, but still performs as the DM model. One-step ahead predictions are almost exact for both outputs, then R^2 in the near horizon reaches 1. Predictions at $k + n_{\mathrm{f}}$ slightly differ from the real output, but R^2 is always above 0.8, then the predictor is considered to successfully approximate the process in the prediction window.

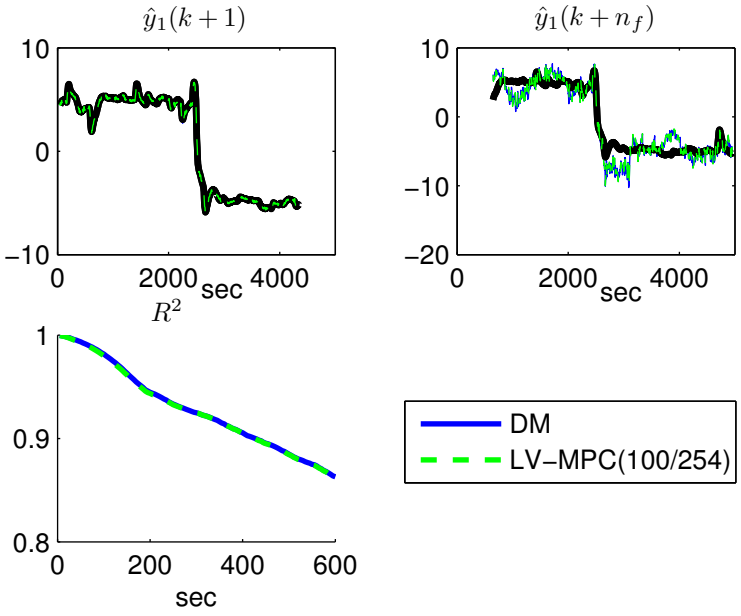


Figure 10.37: Validation results for y_1 removing the working point.

10.4.4. Control results: normal operation

The different control strategies are tested in a situation similar to the identification experiment. Steps are applied to the set points of the CVs. From Figure 10.39

- DM: Presents a little bit more interaction than LV-MPC techniques, but the response is almost equivalent.

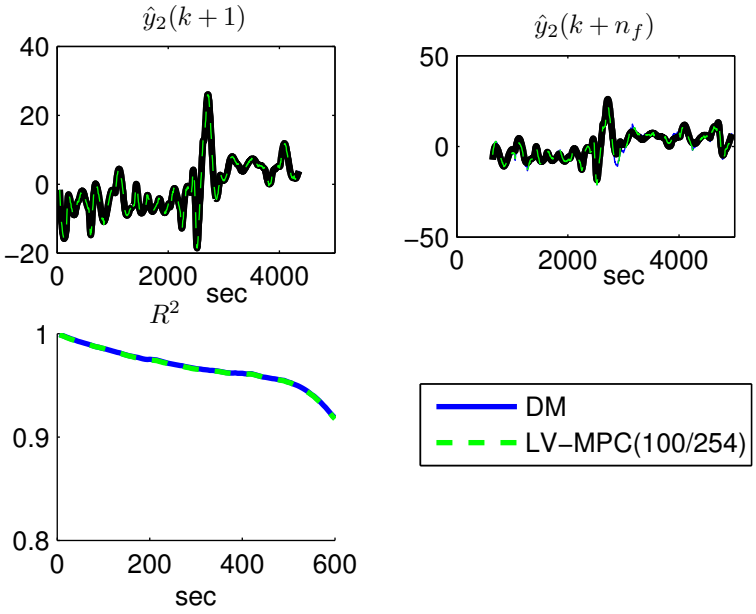


Figure 10.38: Validation results for y_2 removing the working point.

- LV-MPC: Slightly outperforms DM and has less d.o.f.. Note in LV-MPC the controller has 100 d.o.f., whilst in DM there are $n_{\mathrm{u}} n_{\mathrm{i}} = 120$ d.o.f..
- LV-MPC-cons: quadratic constraints in Figure 10.40 are active in some intervals, which provides a different control result to that obtained in LV-MPC. However, since the use of constraints is not too important in this scenario, the difference is negligible.
- LV-MPC-cons-neg: Quadratic constraints neglecting past data are not active for large periods of time, Figure 10.41, then the same result as in LV-MPC is obtained

Summing up, in normal operation all the control strategies evaluated perform similarly.

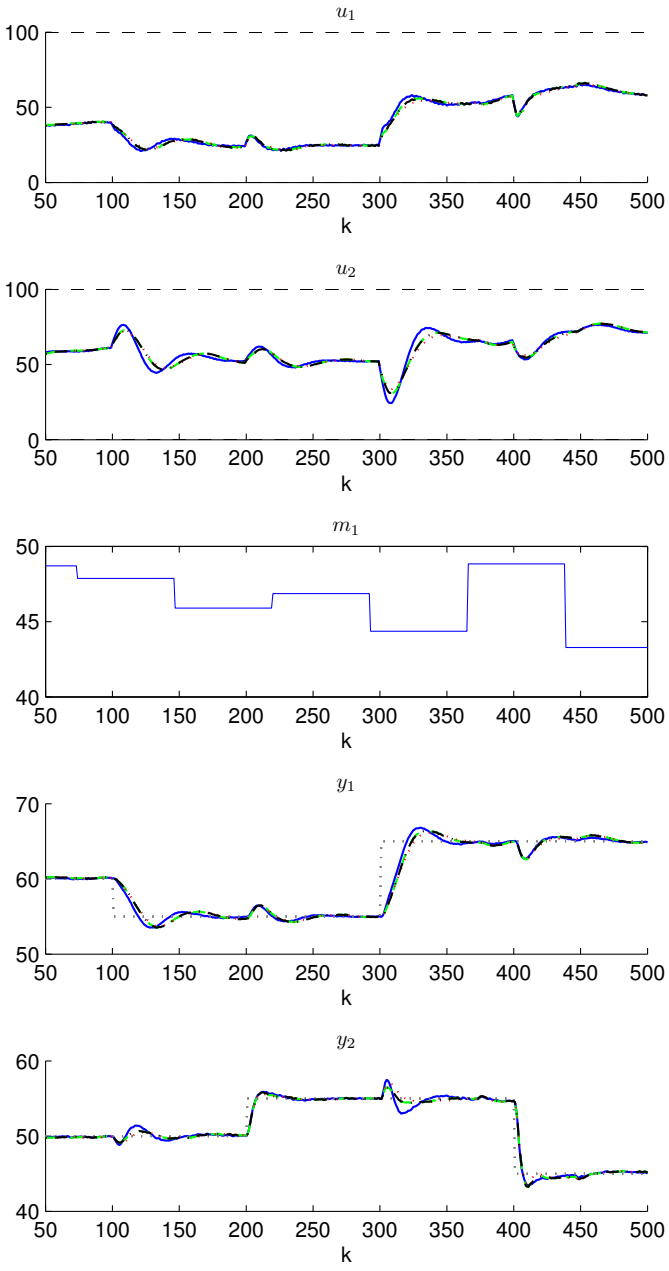


Figure 10.39: Closed-loop time response in normal operation. DM, continuous blue line; LV-MPC, dashed green line; LV-MPC-cons, dotted red line; LV-MPC-cons-neg, dash-dotted black line.

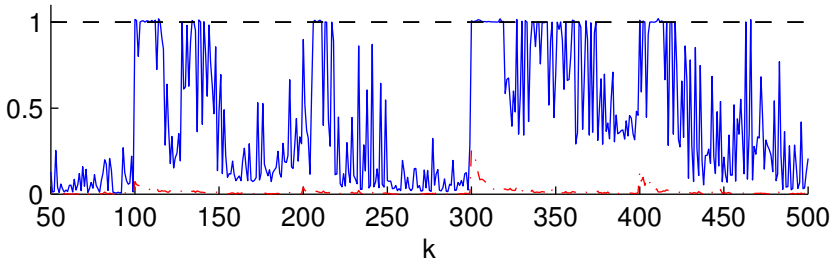


Figure 10.40: LV-MPC-cons: $\bar{J}_{\mathrm{t}}$, continuous blue line; $\bar{J}_{\mathrm{e}}$ dash-dotted red line.

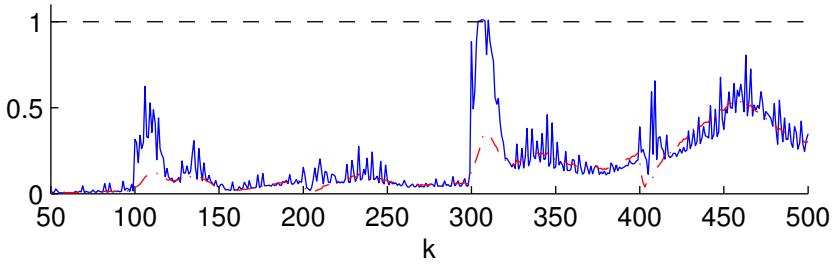


Figure 10.41: LV-MPC-cons-neg: $\check{J}_{\mathrm{t}}$, continuous blue line; $\check{J}_{\mathrm{e}}$ dash-dotted red line.

10.4.5. Control results: large changes in set points and perturbance

The control strategies are evaluated in a situation different to that in the identification data set. Three events happen during the experiment: first a large change in the set point for y_1, second a large change in m_1, and third a large change in the set point for y_2. From Figure 10.42:

- DM: Presents the strongest interaction, but can reach any point as model validity is not ascertained.

- LV-MPC: Equivalent to DM, but has less d.o.f..
- LV-MPC-cons: Interaction to a change in y_1 is considerably reduced as the model is used in the range in which it is valid. From Figure 10.43, $\bar{J}_{\mathrm{t}}$ is at the boundaries of constraints continuously, and $\bar{J}_{\mathrm{e}}$ increases when m_1 changes. Constraints on $\bar{J}_{\mathrm{t}}$ and $\bar{J}_{\mathrm{e}}$ reduce interaction and provide better control if the process is in the area in which it has been identified. However, the resulting control is biased when there is a change in y_2 because of the constraints to ensure validity of predictions.
- LV-MPC-cons-neg: Interaction to a change in y_1 is considerably smaller to that obtained in DM and LV-MPC, but slightly above that obtained if past is not neglected. From Figure 10.44, $\check{J}_{\mathrm{t}}$ and $\check{J}_{\mathrm{e}}$ go to saturation only when changes happen in the experiment. Neglecting past data relaxes constraints on validity and prevents the controller to be biased.

Summing up, constraints on validity of the model neglecting past data provide better results in the event of situations not included in the identification experiment.

Finally, the difference between linearising quadratic constraints at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ and at $\Delta\mathbf{t}_{\mathrm{d_i}}$ is compared in Figure 10.45. The mean value of the constraint for the different instants of the control experiment versus the number of iteration of the sequential QP is represented. The continuous blue plot represents the vale of $\bar{J}_{\mathrm{t}}$ linearising at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$, and the dashed red plot represents the vale of $\bar{J}_{\mathrm{t}}$ linearising at $\Delta\mathbf{t}_{\mathrm{d_i}}$. In the first iteration the QP runs with no linearised constraint, thus both approaches present the same vale of $\bar{J}_{\mathrm{t}}$. For the second iteration, the approach linearising at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ provides a mean value of $\bar{J}_{\mathrm{t}}$ closer to 1. In both approaches $\bar{J}_{\mathrm{t}}$ converges to 1, but if the linearisation is performed at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$, the algorithm converges at a faster rate. In real-time applications, computing time bounds the maximum number of iterations of the QP to implement, hence a fast convergence can be of importance, then linearising at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$ is better than linearising at linearising at $\Delta\mathbf{t}_{\mathrm{d_i}}$.

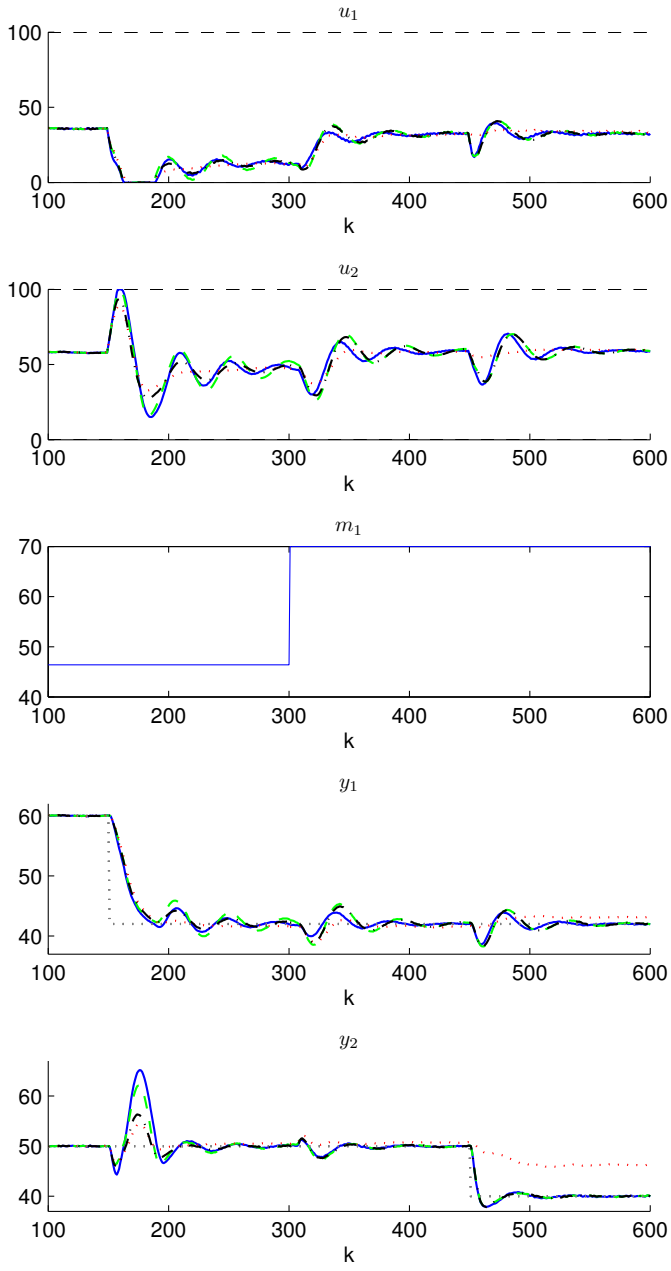


Figure 10.42: Closed-loop time response in the event of a large change in the set point for y_1. DM, continuous blue line; LV-MPC, dashed green line; LV-MPC-cons, dotted red line; LV-MPC-cons-neg, dash-dotted black line.

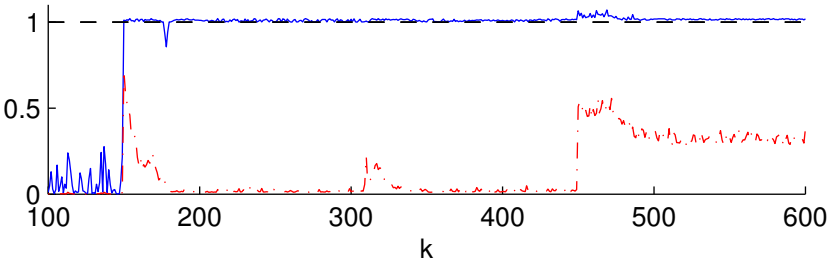


Figure 10.43: LV-MPC-cons: $\bar{J}_{\mathrm{t}}$, continuous blue line; $\bar{J}_{\mathrm{e}}$ dash-dotted red line.

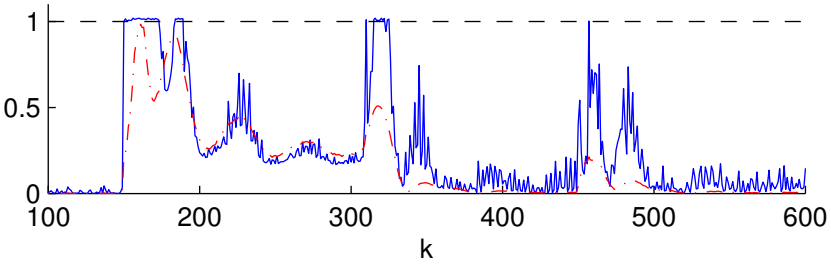


Figure 10.44: LV-MPC-cons-neg: $\check{J}_{\mathrm{t}}$, continuous blue line; $\check{J}_{\mathrm{e}}$ dash-dotted red line.

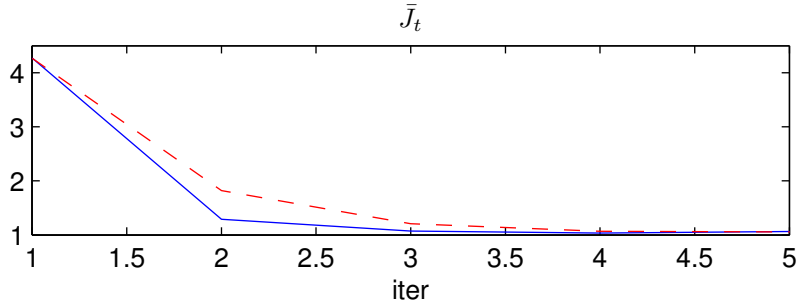


Figure 10.45: Decrease rate of $\bar{J}_{\mathrm{t}}$. Linearising at $\Delta\mathbf{t}_{\mathrm{d_{ti}}}$, continuous blue line; linearising at $\Delta\mathbf{t}_{\mathrm{d_i}}$, dashed red line.

CHAPTER 11

CONCLUSIONS

This **Part III** of the thesis proposes LV-MPC for continuous processes. LV-MPC is a model predictive control strategy for continuous processes implemented in the space of the latent variables. Part of the LV-MPC methodology has been published in [Laurí 10b].

The major advantages of LV-MPC are:

- The identification is MPC relevant (MRI) provided the multi-step ahead predictor is directly identified from the identification data set.
- Identification is performed by means of PLS
  - Accounts for correlation in the identification data set.
  - Acts as a prefilter reducing the effect of noisy data.
- Control is implemented in the reduced space of the latent variables which

  - Can reduce computational complexity by reducing the decision space.
  - Improves robustness for tightly tuned controllers.
  - Makes it possible to constrain the decision space of the controller to ensure model validity.

- Systematic tuning strategy for n_{u}

The major limitations of LV-MPC are:

- The structure of the model is linear and Industrial processes are non-linear. However, some methodologies can be combined with LV-MPC to cope with non-linearities: add constraints to MVs, keep the process around the operating point, compensate static non-linearities, or combine linear local models.
- LV-MPC, As presented in this thesis, is a dual mode strategy with no terminal control. Thus, LV-MPC stability can only be achieved for open-loop stable processes with $n_{\mathrm{f}} - n_{\mathrm{u}}$ greater than the process settling time.

Part IV

Concluding remarks

CHAPTER 12

CONCLUSIONS AND FUTURE WORK

12.1. Conclusions

This thesis deals with identification for MPC and implementation of MPC in the space of the latent variables.

As for identification, the different approaches to MRI are commented and it is proven that unlike in one-step ahead identification, the MIMO approach is preferable to the multiple MISO approach in MRI for a sufficiently large prediction horizon. Additionally, a PLS line search numerical optimization approach to deal with parametric MRI is proposed. PLS-PH is simpler than traditional numerical optimization methods provided derivatives of the non-linear cost function are not required. The PLS-PH approach outperforms LS in an MPC framework and also outperforms traditional numerical optimization when the data is ill-conditioned, being equivalent to it otherwise.

As for control, a model predictive control strategy for continuous processes implemented in the space of the latent variables is provided. The major advantages of LV-MPC are:

- The identification is MRI provided the multi-step ahead predictor is directly identified from the identification data set.
- Identification is performed by means of PLS
  - Accounts for correlation in the identification data set.
  - Acts as a prefilter reducing the effect of noisy data.
- Control is implemented in the reduced space of the latent variables which
  - Can reduce computational complexity by reducing the decision space.
  - Improves robustness for tightly tuned controllers.
  - Makes it possible to constrain the decision space of the controller to ensure model validity.
- Systematic tuning strategy.

12.2. Contributions

This thesis yields the following contributions:

- [Under review]: LV-MPC: Ensure Validity of Predictions. Journal of Process Control.
- [Laurí 10b]: Driven Latent-Variable Model-Based Predictive Control for Continuous Processes. Journal of Process Control, vol. 20, pages 1207–1219, 2010. **(TOP25, October to December 2010, Chemical Engineering)**

- [Laurí 10a]: PLS-based model predictive control relevant identification: PLS-PH algorithm. Chemometrics and Intelligent Laboratory Systems, vol. 100, pages 118–126, 2010. **(TOP25, January to March 2010, Mathematics)**
- [Laurí 10c]: Model predictive control relevant identification: multiple input multiple output against multiple input single output. IET Control Theory and Applications, vol. 9, pages 1756–1766, 2010.
- [Laurí 09]: A PLS Approach to Identifying Predictive ARX Models, 3rd IEEE Multi-conference on Systems and Control (MSC 2009)

12.3. Future work

Some ideas for future work:

- Program the algorithm in C so that it can be implemented in real-time applications for fast processes.
- Consider measurable perturbations in the model for better predictions.
- Use PCR instead of PLS to perform the projection onto the latent variable space and compare both approaches.
- The dynamic matrix approach used in this thesis to account for MRI comes from SMI [Huang 08]. Causality of the model can be enforced in SMI [Qin 06]. In the identification approach in this paper however, causality of the model is not enforced; thus, it can be considered future work to improve quality of predictions.
- Combine LV-MPC with two-stage subspace identification (SSID) proposed in [Kano 09]. In SSID highly accurate soft-sensors that take into account the influence of unmeasured disturbances on estimated key variables are implemented.

BIBLIOGRAPHY

[Albertos 04] P. Albertos & A. Sala. Multivariable control systems. Springer, 2004.

[AlGhazzawi 09] A. AlGhazzawi & B. Lennox. *Model predictive control monitoring using multivariate statistics.* Journal of Process Control, vol. 19, pages 314–327, 2009.

[Bars 06] R. Bars, P. Colaneri, C. E. de Souza, L. Dugard, F. Allgöwer, A. Kleimenov & C. Scherer. *Theory, algorithms and technology in the design of control system.* Annual Reviews in Control, vol. 30, pages 19–30, 2006.

[Bemporad 99] A. Bemporad & M. Morari. *Robust Model Predictive Control: A Survey.* Lecture notes in control and information sciences, pages 207–226, 1999.

[Blet 02] N. Blet, D. Megías, J. Serrano & C. de Prada. *Non linear MPC versus MPC using on-line linearisation - a comparative study.* In IFAC, 2002.

[Boyd 04] S. Boyd & L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[Cagienard 07] R. Cagienard, P. Grieder, EC Kerrigan & M. Morari. *Move blocking strategies in receding horizon control.* Journal of Process Control, vol. 17, pages 563–570, 2007.

[Camacho 04] E. F. Camacho & C. Bordons. Model Predictive Control. Springer, 2004.

[Chen 98] G. Chen, T.J. McAvoy & M.J. Piovoso. *A multivariatle statistical controller for on-line quality improvement.* Journal of Process Control, vol. 8, pages 139–150, 1998.

[Fletcher 87] R. Fletcher. Practical methods of optimization. Wiley, 1987.

[Flores-Cerrillo 04] J. Flores-Cerrillo & J.F. MacGregor. *Control of batch product quality by trajectory manipulation using latent variable models.* Journal of Process Control, vol. 14, pages 539–553, 2004.

[Flores-Cerrillo 05] J. Flores-Cerrillo & J.F. MacGregor. *Latent variable MPC for trajectory tracking in batch processes.* Journal of Process Control, vol. 15, pages 651–663, 2005.

[Geladi 86] P. Geladi & B. R. Kowalski. *Partial least-squares regressions: a tutorial.* Analytica Chimica Acta, vol. 185, pages 1–17, 1986.

[Gevers 02] M. Gevers. *A decade of progress in iterative process control design: from theory to practice.* Journal of Process Control, vol. 12, pages 519–531, 2002.

[Gopaluni 02a] R. B. Gopaluni, R. S. Patwardhan & S. L. Shah. *Bias distribution in MPC relevant identification.* In Proceedings of IFAC world congress, Barcelona, 2002.

[Gopaluni 02b] R. B. Gopaluni, R. S. Patwardhan & S. L. Shah. *Experiment design for MPC relevant identification.* In Proceedings of the American Control Conference, 2002.

[Gopaluni 04] R. B. Gopaluni, R. S. Patwardhan & S. L. Shah. *MPC relevant identification-tuning the noise model.* Journal of Process Control, vol. 14, pages 699–714, 2004.

[Hjalmarsson 06] H. Hjalmarsson & B. Ninness. *Least-squares estimation of a class of frequency functions: A finite sample variance expression.* Automatica, vol. 42, pages 589–600, 2006.

[Höskuldsson 88] A. Höskuldsson. *PLS regression methods.* Journal of Chemometrics, vol. 2, no. 3, pages 211–228, 1988.

[Huang 08] B. Huang & R. Kadali. Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-driven Subspace Approach. Springer, 2008.

[Jämsä-Jounela 07] S.-L. Jämsä-Jounela. *Future trends in process automation.* Annual Reviews in Control, vol. 31, pages 211–220, 2007.

[Kadali 03] R. Kadali, B. Huang & A. Rossiter. *A data driven subspace approach to predictive controller design.* Control Engineering Practice, vol. 11, pages 261–278, 2003.

[Kano 08] M. Kano & Y. Nakagawa. *Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry.* Computers and Chemical Engineering, vol. 32, pages 12–24, 2008.

[Kano 09] M. Kano, S. Lee & S. Hasebe. *Two-stage subspace identification for softsensor design and disturbance estimation.* Journal of Process Control, vol. 19, pages 179–186, 2009.

[Kiers 07] H. A. L. Kiers & A. K. Smilde. *A comparison of various methods for multivariate regression with highly collinear variables.* Statistical Methods and Applications, vol. 16, pages 193–228, 2007.

[Kourti 05] T. Kourti. *Application of latent variable methods to process control and multivariate statistical process control in industry*. International Journal of Adaptive Control and Signal Processing, vol. 19, pages 213–246, 2005.

[Kvasnica 04] M. Kvasnica, P. Grieder, M. Baotic & M. Morari. *Multi-parametric toolbox (MPT)*. Hybrid Systems: Computation and Control, pages 121–124, 2004.

[Laurí 09] D. Laurí, J. Salcedo, S. García-Nieto & M. Martínez. *A PLS Approach to Identifying Predictive ARX Models*. In 3rd IEEE Multi-conference on Systems and Control (MSC 2009), 2009.

[Laurí 10a] D. Laurí, M. Martínez, J.V. Salcedo & J. Sanchis. *PLS-based model predictive control relevant identification: PLS-PH algorithm*. Chemometrics and Intelligent Laboratory Systems, vol. 100, pages 118–126, 2010.

[Laurí 10b] D. Laurí, J.A. Rossiter, J. Sanchis & M. Martínez. *Data-Driven Latent-Variable Model-Based Predictive Control for Continuous Processes*. Journal of Process Control, vol. 20, pages 1207–1219, 2010.

[Laurí 10c] D. Laurí, J.V. Salcedo, S. García-Nieto & M. Martínez. *Model predictive control relevant identification: multiple input multiple output against multiple input single output*. IET Control Theory and Applications, vol. 9, pages 1756–1766, 2010.

[Ljung 99] L. Ljung. System identification. Theory for the user. Prentice Hall, 1999.

[MacGregor 09] J. MacGregor, M.J. Bruwer & M. Goldshan. *System and method for the model predictive control of batch processes using latent variable dynamic models*, 2009.

[Martens 01] H. Martens. *Reliable and relevant modelling of real world data: a personal account of the development of PLS Regression.* Chemometrics and Intelligent Laboratory Systems, vol. 58, pages 85–95, 2001.

[Mårtensson 09] J. Mårtensson & H. Hjalmarsson. *Variance-error quantification for identified poles and zeros.* Automatica, vol. 45, pages 2512–2525, 2009.

[McAvoy 02] T. McAvoy. *Model Predictive Statistical Process Control of Chemical Plants.* Industrial and Engineering Chemistry Research, vol. 41, pages 6337–6344, 2002.

[Morari 99] M. Morari & J. H. Lee. *Model predictive control: past, present and future.* Computers and Chemical Engineering, vol. 23, pages 667–682, 1999.

[Nelson 96] P.R.C. Nelson, P.A. Taylor & J.F. MacGregor. *Missing data methods in PCA and PLS: Score calculations with incomplete observations.* Chemometrics and intelligent laboratory systems, vol. 35, pages 45–65, 1996.

[Ninness 04] B. Ninness & H. Hjalmarsson. *Variance error quantifications that are exact for finite-model order.* IEEE Transactions on Automatic Control, vol. 49, pages 1275–1291, 2004.

[Nocedal 99] J. Nocedal & S. J. Wright. Numerical Optimization. Springer, 1999.

[Pukrushpan 04] J. Pukrushpan, H. Peng & A. Stefanopoulou. *Control-oriented modeling and analysis for automotive fuel cell systems.* Journal of Dynamic Systems, Measurement and Control, vol. 126, pages 14–25, 2004.

[Qin 03] S. J. Qin & T. A. Badgwell. *A survey of industrial model predictive control technology.* Control Engineering Practice, vol. 11, pages 733–764, 2003.

[Qin 06] S. J. Qin. *An overview of subspace identification.* Computers and Chemical Engineering, vol. 30, pages 1502–1513, 2006.

[Rivera 92] D. E. Rivera, I. F. Pollard & C. E. Garcia. *Control-relevant prefiltering: a systematic design approach and case study.* Automatic Control, IEEE Transactions on, vol. 37, pages 964–974, 1992.

[Rossiter 01] J. A. Rossiter & B. Kouvaritakis. *Modelling and implicit modelling for predictive control.* International Journal of Control, vol. 74, pages 1085–1095, 2001.

[Rossiter 03] J. A. Rossiter. Model-Based Predictive Control: a Practical Approach. CRC press, 2003.

[Sanchis 02] J. Sanchis. *GPC mediante descomposición en valores singulares (SVD).* PhD thesis, Universidad Politécnica de Valencia, 2002.

[Shook 91] D. S. Shook, C. Mohtadi & S. L. Shah. *Identification for long-range predictive control.* In IEE proceeedings, 1991.

[Shook 92] D. S. Shook, C. Mohtadi & S. L. Shah. *A Control-Relevant Identification Strategy for GPC.* IEEE Transactions on Automatic Control, vol. 37, pages 975–980, 1992.

[Shridhar 98] R. Shridhar & D.J. Cooper. *A tuning strategy for unconstrained multivariable model predictive control.* Ind. Eng. Chem. Res, vol. 37, pages 4003–4016, 1998.

[Skogestad 97] S. Skogestad. *Dynamics and control of distillation columns: a tutorial introduction.* Chemical Engineering Research and Design, vol. 75, pages 539–562, 1997.

[Song 02] K. Song, P-Y. Jang, H. Cho & C. H. Jun. *Partial least square-based model predictive control for large-scale manufacturing processes.* IIE Transactions, vol. 34, pages 881–890, 2002.

[Sotomayor 09] O. A. Z. Sotomayor, D. Odloak & L. F. L. Moro. *Closed-loop model re-identification of processes under MPC with zone control*. Control Engineering Practice, vol. 17, pages 551–563, 2009.

[Thwaites 07] P. Thwaites. *Process control in metallurgical plants-from an Xstrata perspective*. Annual Reviews in Control, vol. 31, pages 211–239, 2007.

[Tropsha 03] A. Tropsha, P. Gramatica & V. K. Gombar. *The importance of being earnest: Validation is the absolute essential for successful application and interpretation of QSPR models*. QSAR and Combinatorial Science, vol. 22, no. 1, pages 69–77, 2003.

[Veselý 10] V. Veselý, D. Rosinová & M. Foltin. *Robust model predictive control design with input constraints*. ISA Transactions, vol. 49, pages 114–120, 2010.

[Wang 05] C. Wang, M. H. Nehrir & S. R. Shaw. *Dynamic Models and Model Validation for PEM Fuel Cells Using Electrical Circuits*. IEEE Transactions on Energy Conversion, vol. 20, pages 442–451, 2005.

[Wang 09] L. Wang. Model Predictive Control System Design and Implementation Using Matlab. Springer, 2009.

[Wood 73] R. K. Wood & M. W. Berry. *Terminal composition control of a binary distillation column*. Chemical Engineering Science, vol. 28, pages 1707–1717, 1973.

[Zhu 02] Y. Zhu & F. Butoyi. *Case studies on closed-loop identification for MPC*. Control Engineering Practice, vol. 10, pages 403–417, 2002.

APPENDIX A

MIMO ARX MODEL

Let us take the MIMO ARX model:[1]

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_{\mathrm{n_o}}(k) \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{B}_{1,1}}{\mathbf{A}_1} & \cdots & \frac{\mathbf{B}_{\mathrm{n_i},1}}{\mathbf{A}_1} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{B}_{1,\mathrm{n_o}}}{\mathbf{A}_{\mathrm{n_o}}} & \cdots & \frac{\mathbf{B}_{\mathrm{n_i},\mathrm{n_o}}}{\mathbf{A}_{\mathrm{n_o}}} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_{\mathrm{n_i}}(k) \end{bmatrix} + \begin{bmatrix} \frac{\xi_1(k)}{\mathbf{A}_1} \\ \vdots \\ \frac{\xi_{\mathrm{n_o}}(k)}{\mathbf{A}_{\mathrm{n_o}}} \end{bmatrix} \tag{A.1}$$

Where:

$$\mathbf{B}_{\mathrm{e,s}} = b_{1_{\mathrm{e,s}}} z^{-1} + \ldots + b_{\mathrm{n_{b_{e,s}}}} z^{-n_\mathrm{b}}$$

$$\mathbf{A}_\mathrm{s} = 1 + a_{1_\mathrm{s}} z^{-1} + \ldots + a_{\mathrm{n_{a_s}}} z^{-n_\mathrm{a}}$$

$$\mathrm{e} \in [1, 2, \ldots, n_\mathrm{i}] \quad ; \quad \mathrm{s} \in [1, 2, \ldots, n_\mathrm{o}].$$

For simplicity, all the numerators and denominators are assumed to be of the same order. Otherwise, polynomial operations may be applied, and some

[1] For ease of notation, the polynomials in z^{-1} are represented only by their name, $\mathbf{A}_{\mathrm{n_o}}(z^{-1}) \equiv \mathbf{A}_{\mathrm{n_o}}$.

values may be fixed to zero to transform the original structure into the given general expression for a MIMO ARX model.

Output s can be expressed:

$$\mathbf{A}_{\mathrm{s}} y_{\mathrm{s}}(k) = [\mathbf{B}_{1,\mathrm{s}} \dots \mathbf{B}_{\mathrm{n_i},\mathrm{s}}] \begin{bmatrix} u_1(k) \\ \vdots \\ u_{\mathrm{n_i}}(k) \end{bmatrix} + \xi_{\mathrm{s}}(k)$$

$$\begin{aligned} y_{\mathrm{s}}(k) =& [\mathbf{B}_{1,\mathrm{s}} \dots \mathbf{B}_{\mathrm{n_i},\mathrm{s}}] \begin{bmatrix} u_1(k) \\ \vdots \\ u_{\mathrm{n_i}}(k) \end{bmatrix} + (1 - \mathbf{A}_{\mathrm{s}}) y_{\mathrm{s}}(k) + \xi_{\mathrm{s}}(k) \\ =& [\bar{\mathbf{b}}_{1_{\mathrm{s}}} \dots \bar{\mathbf{b}}_{\mathrm{n_{b_s}}}] \begin{bmatrix} \mathbf{u}_{\mathrm{k}-1}^T \\ \vdots \\ \mathbf{u}_{\mathrm{k}-\mathrm{n_b}}^T \end{bmatrix} + [-a_{1_{\mathrm{s}}} \dots - a_{\mathrm{n_{a_s}}}] \begin{bmatrix} y_{\mathrm{s}}(k-1) \\ \vdots \\ y_{\mathrm{s}}(k-n_{\mathrm{a}}) \end{bmatrix} + \xi_{\mathrm{s}}(k) \\ =& [\mathbf{u}_{\mathrm{k}-1} \dots \mathbf{u}_{\mathrm{k}-\mathrm{n_b}}, y_{\mathrm{s}}(k-1) \dots y_{\mathrm{s}}(k-n_{\mathrm{a}})] \begin{bmatrix} \bar{\mathbf{b}}_{1_{\mathrm{s}}}^T \\ \vdots \\ \bar{\mathbf{b}}_{\mathrm{n_{b_s}}}^T \\ -a_{1_{\mathrm{s}}} \\ \vdots \\ -a_{\mathrm{n_{a_s}}} \end{bmatrix} + \xi_{\mathrm{s}}(k) \end{aligned}$$

Where:

$$\bar{\mathbf{b}}_{\alpha_{\mathrm{s}}} = [b_{\alpha_{1,\mathrm{s}}} \dots b_{\alpha_{\mathrm{n_i},\mathrm{s}}}], \quad \forall \alpha \in [1, 2, \dots, n_{\mathrm{b}}]$$

$$\mathbf{u}_{\mathrm{k}} = [u_1(k) \dots u_{\mathrm{n_i}}(k)].$$

The general expression for linear models used in this thesis can be derived from the previous equation as

$$\mathbf{y}_{\mathrm{k}} = \mathbf{x}_{\mathrm{k}-1} \boldsymbol{\theta} + \boldsymbol{\xi}_k \tag{A.2}$$

Where:

$$\mathbf{y}_{\mathrm{k}} = [y_1(k) \dots y_{\mathrm{n_o}}(k)]$$

$$\mathbf{x}_{\mathrm{k}-1} = [\mathbf{u}_{\mathrm{k}-1} \dots \mathbf{u}_{\mathrm{k}-\mathrm{n_b}},\ \mathbf{y}_{\mathrm{k}-1} \dots \mathbf{y}_{\mathrm{k}-\mathrm{n_a}}]$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \mathbb{B}^T \\ \mathbb{A}_1 \\ \vdots \\ \mathbb{A}_{n_\mathrm{a}} \end{bmatrix}$$

$$\mathbb{B} = \begin{bmatrix} \bar{\mathbf{b}}_{1_1} & \dots & \bar{\mathbf{b}}_{\mathrm{n_{b1}}} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{b}}_{1_{\mathrm{n_o}}} & \dots & \bar{\mathbf{b}}_{\mathrm{n_{b n_o}}} \end{bmatrix}$$

$$\mathbb{A}_\beta = \begin{bmatrix} -a_{\beta_1} & 0 & \dots & \dots & 0 \\ 0 & & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & \dots & \dots & 0 & -a_{\beta_{n_\mathrm{o}}} \end{bmatrix}, \forall \beta \in [1, 2, \dots, n_\mathrm{a}] \tag{A.3}$$

$$\boldsymbol{\xi}(k) = [\xi_1(k) \dots \xi_{n_\mathrm{o}}(k)].$$

APPENDIX B

PLS-PH ALGORITHM

The pseudo-code for the algorithm is described using some Matlab® notation[1]:

1. Select n_{lv} using $\mathbf{X}$ and $\mathbf{Y}$ in PLS and crossed validation techniques[2]
2. $\boldsymbol{\theta}_{\mathrm{k}}$ is obtained by applying PLS to $\mathbf{X}$ and $\mathbf{Y}$ using n_{lv}
3. Form $\mathbf{Y}_{\mathrm{a}}$ and obtain $\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}$ using $\boldsymbol{\theta}_{\mathrm{k}}$
4. Obtain $\mathbf{W}$, $\mathbf{P}$, B and $\mathbf{Q}$ applying PLS to $\mathbf{X}_{\mathrm{a}}|_{\mathrm{k}}$ and $\mathbf{Y}_{\mathrm{a}}$
5. Obtain $\mathbf{p}_{\mathrm{k}}$ using eq. (7.15)

[1] Limiting the maximum number of iterations in each of the two loops of the algorithm is advisable.

[2] The LS-PH algorithm is a particular case of PLS-PH with $n_{\mathrm{lv}} = n_{\mathrm{x}}$. Such value of n_{lv} is obtained when there is no collinearity in the data set; thus, no reduction in the number of variables in the regression matrix is needed.

Search for α_{k}

6. $\overrightarrow{\alpha} = [0 \quad 0.5 \quad 1]$
7. Estimate a, b and c in eq. (7.12) using the three values in $\overrightarrow{\alpha}$.
8. Find the point in which the first derivative of eq. (7.12) vanishes:

$$\alpha' = \frac{-b}{2c}$$

9. Decide the new value of α_{k}:

   If $c > 0$ (the parabola moves up)

   If $0 < \alpha' < 1$

   $\alpha_{\mathrm{k}} = \alpha'$

   else

   If $\alpha' \leqq 0$

   α_{k} equals the middle point between $\min_{\alpha}(\overrightarrow{\alpha}), s.t.\ \alpha > 0$, and 0

   else

   α_{k} equals the middle point between $\max_{\alpha}(\overrightarrow{\alpha}), s.t.\ \alpha < 1$, and 1

   else

   If $\alpha' > \overrightarrow{\alpha}(1,1)$

   α_{k} equals the middle point between $\min_{\alpha}(\overrightarrow{\alpha}), s.t.\ \alpha > 0$, and 0

   else

   α_{k} equals the middle point between $\max_{\alpha}(\overrightarrow{\alpha}), s.t.\ \alpha < 1$, and 1

10. The value of $\overrightarrow{\alpha}$ with the largest cost is removed and α_{k} is added to $\overrightarrow{\alpha}$.
11. Go to step 12 if eq. (7.13) holds; otherwise, go to step 7
12. α_{k} equals the value in $\overrightarrow{\alpha}$ with the lowest J_{LRPI}
13. $\boldsymbol{\theta}_{\mathrm{k}+1} = \boldsymbol{\theta}_{\mathrm{k}} + \alpha_{\mathrm{k}}\, \mathbf{p}_{\mathrm{k}}$
14. Finish if eq. (7.14) holds; otherwise, $k = k + 1$ and go to step 3

APPENDIX C

OBTAINING $\mathbf{X}_{\mathrm{DOF}}$

Obtain $\mathbf{x}_{\mathrm{dof}}$ given $\mathbf{t}_{\mathrm{d}}$ and $\mathbf{x}_{\mathrm{p}}$. One may think $\mathbf{x}_{\mathrm{dof}}$ can be obtained by projecting $\mathbf{t}$ back to the original space by means of the corresponding submatrix in $\mathbf{P}$, from Equation (4.2):

$$\mathbf{x} = \mathbf{t}\mathbf{P}^T \quad \Rightarrow \quad \underbrace{[\mathbf{x}_{\mathrm{p}}\, \mathbf{x}_{\mathrm{f}}\, \mathbf{x}_{\mathrm{dof}}]}_{\mathbf{x}} = \mathbf{t}\underbrace{[\mathbf{P}_{\mathrm{p}}^T\, \mathbf{P}_{\mathrm{f}}^T\, \mathbf{P}_{\mathrm{dof}}^T]}_{\mathbf{P}^T} \quad \Rightarrow \quad \mathbf{x}_{\mathrm{dof}} = \mathbf{t}\mathbf{P}_{\mathrm{dof}}^T \tag{C.1}$$

Let us denote this, option A.

An alternative approach is presented in proposition 9.2.1, let us denote this approach option B.

$$\mathbf{x}_{\mathrm{dof}} = \mathbf{t}_{\mathrm{d}}\underbrace{(\mathbf{Z}_{\mathrm{dof}}^T\mathbf{Z}_{\mathrm{dof}})^{-1}\mathbf{Z}_{\mathrm{dof}}^T}_{\mathbf{M}_{\mathrm{dof}}}.$$

In this appendix options A and B are compared. Note $\mathbf{x}$ is used to perform predictions in the PLS model, and $\mathbf{x} = [\mathbf{x}_\mathrm{p} \quad \mathbf{x}_\mathrm{dof}]$.[1] When performing predictions, $\mathbf{x}$ is projected to obtain $\mathbf{t}$ using Equation (9.4). For the methodology to be consistent, if $\mathbf{t}$ is used to obtain $\mathbf{x}_\mathrm{dof}$, the projection of $\mathbf{x} = [\mathbf{x}_\mathrm{p} \quad \mathbf{x}_\mathrm{dof}]$ by means of $\mathbf{Z}$ should provide the same value of $\mathbf{t}$ if $\mathbf{t}$ is used to perform predictions. Note that from Equation (9.4), $\mathbf{t}$ can be expressed:

$$\mathbf{t} = \mathbf{t}_\mathrm{p} + \mathbf{t}_\mathrm{d} = \mathbf{x}_\mathrm{p}\mathbf{Z}_\mathrm{p} + \mathbf{t}_\mathrm{d}.$$

To ease understanding of the comparison, one figure for each option is shown. The examples in the figures assume $\mathbf{x}_\mathrm{p}$, $\mathbf{x}_\mathrm{dof}$, and $\mathbf{t}$ have one dimension. $\mathbf{t}_\mathrm{p}$ is obtained as the projection of $\mathbf{x}_\mathrm{p}$ by means of $\mathbf{Z}_\mathrm{p}$, Equation (9.4). $\mathbf{t}_\mathrm{dof}$[2] is the decision variable and is given an arbitrary value for the plots.

Option A

$\mathbf{x}_\mathrm{dof}$ in Equation (C.1) is obtained as the projection of $\mathbf{t}$ onto the outer space, see Figure C.1. Using the known $\mathbf{x}_\mathrm{p}$ and the obtained $\mathbf{x}_\mathrm{dof}$, $\mathbf{x}$ yields

$$\mathbf{x} = [\mathbf{x}_\mathrm{p} \quad \mathbf{t}\mathbf{P}_\mathrm{dof}^T].$$

The projection of $\mathbf{x}$ onto the latent variable space, $\mathbf{t}^*$, should be equal to $\mathbf{t}$. The projection is performed by means of Equation (9.4).

$$\begin{aligned} \mathbf{t}^* &= \mathbf{x}\mathbf{Z} \\ &= \mathbf{x}_\mathrm{p}\mathbf{Z}_\mathrm{p} + \mathbf{t}\mathbf{P}_\mathrm{dof}^T\mathbf{Z}_\mathrm{dof} \\ &= \mathbf{t}_\mathrm{p} + \mathbf{t}\mathbf{P}_\mathrm{dof}^T\mathbf{Z}_\mathrm{dof} \end{aligned}$$

Figure C.1 shows how $\mathbf{t}^*$ is obtained by projecting $\mathbf{x}$ onto the latent variable space.

[1] In the sake of simplicity, n_u in this section is assumed to be equal to n_f, thus there is no $\mathbf{x}_\mathrm{f}$.

[2] Note $\mathbf{t}_\mathrm{d} = \mathbf{t}_\mathrm{dof}$.

Provided $\mathbf{t} = \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{dof}}$, from the previous equation $\mathbf{t}^* \neq \mathbf{t}$. This can also be appreciated in Figure C.1. Hence, if $\mathbf{x}_{\text{dof}}$ is obtained using Equation (C.1), $\mathbf{t}$ cannot be used to perform predictions, thus this methodology is inconsistent.

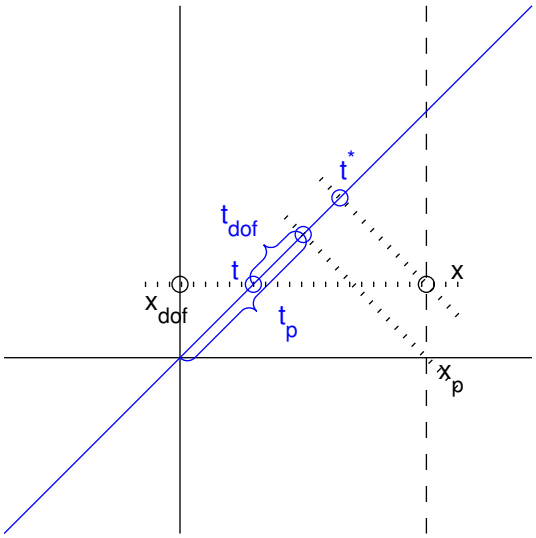


Figure C.1: $\mathbf{x}_{\text{dof}}$ option A

Option B

Figure C.2 shows how $\mathbf{x}_{\text{dof}}$ is obtained from $\mathbf{t}_{\text{dof}}$, proposition 9.2.1. Using the known $\mathbf{x}_{\text{p}}$ and the obtained $\mathbf{x}_{\text{dof}}$, $\mathbf{x}$ yields

$$\mathbf{x} = [\mathbf{x}_{\text{p}} \quad \mathbf{t}_{\text{dof}}(\mathbf{Z}_{\text{dof}}^T\mathbf{Z}_{\text{dof}})^{-1}\mathbf{Z}_{\text{dof}}^T].$$

The projection of $\mathbf{x}$ onto the latent variable space, $\mathbf{t}^*$, should be equal to $\mathbf{t}$:

$$\begin{aligned} \mathbf{t}^* &= \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{dof}}(\mathbf{Z}_{\text{dof}}^T\mathbf{Z}_{\text{dof}})^{-1}\mathbf{Z}_{\text{dof}}^T\mathbf{Z}_{\text{dof}} \\ &= \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{dof}} \end{aligned}$$

Figure C.2 shows how $\mathbf{t}^*$ is obtained by projecting $\mathbf{x}$ onto the latent variable space.

Provided $\mathbf{t} = \mathbf{t}_\mathrm{p} + \mathbf{t}_\mathrm{dof}$, from the previous Equation $\mathbf{t}^* = \mathbf{t}$, which can be appreciated in Figure C.2. Hence, the use of proposition 9.2.1 to obtain $\mathbf{x}_\mathrm{dof}$ is consistent with using $\mathbf{t}$ to perform predictions.

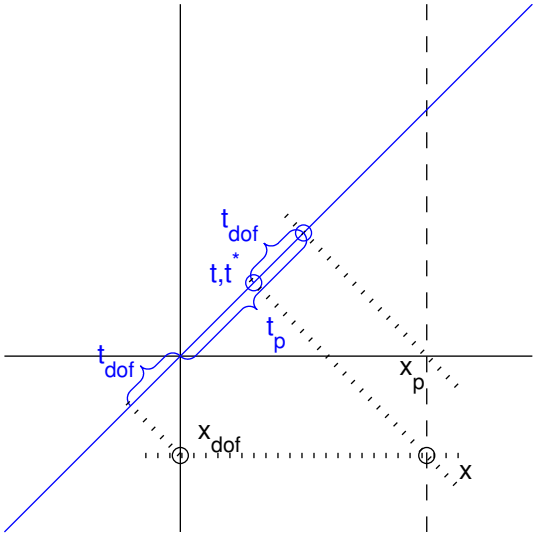


Figure C.2: $\mathbf{x}_\mathrm{dof}$ option B

For the methodology in option A to be consistent, $\mathbf{t}^*$ instead of $\mathbf{t}$ would need be used to perform predictions in Equation (9.9). In this paper option B has been used, however, one could alternatively use option A noting predictions need be performed using $\mathbf{t}^*$. Both methodologies have been implemented and equivalent results have been obtained in terms of closed-loop performance and Hessian conditioning, thus, any of them may be used. The reason to use option B is the formulation allows us to improve Hessian conditioning in Section 9.3.

APPENDIX D

CONSISTENCY

The basic control methodology described in Section 9.2 is proven to be consistent in appendix C. In Section 9.3 matrices $\bar{\mathbf{M}}_{\text{dof}}$ and $\bar{\mathbf{M}}_{\text{t}}$ are obtained to improve Hessian conditioning. This Appendix proves the methodology with the new matrices $\bar{\mathbf{M}}_{\text{dof}}$ and $\bar{\mathbf{M}}_{\text{t}}$ is still consistent.

For the methodology to be consistent, if $\mathbf{t}$ is used to obtain $\mathbf{x}_{\text{fdof}}$, the projection of $\mathbf{x} = [\mathbf{x}_{\text{p}} \quad \mathbf{x}_{\text{fdof}}]$ by means of $\mathbf{Z}$ should provide the same value of $\mathbf{t}$ if $\mathbf{t}$ is used to perform predictions.

Using the known $\mathbf{x}_{\text{p}}$ and the $\mathbf{x}_{\text{fdof}}$ obtained in Equation (9.19), $\mathbf{x}$ yields

$$\mathbf{x} = [\mathbf{x}_{\text{p}} \quad \mathbf{x}_{\text{fdof}}] = [\mathbf{x}_{\text{p}} \quad \mathbf{t}_{\text{d}} \underbrace{(\mathbf{M}_{\text{QP}} + \mathbf{Z}_{\text{fdof}}^{-1*})}_{\bar{\mathbf{M}}_{\text{fdof}}}].$$

Note $\bar{\mathbf{M}}_{\text{dof}}$ is a submatrix of $\bar{\mathbf{M}}_{\text{fdof}}$. The projection of $\mathbf{x}$ onto the latent variable space, $\mathbf{t}^*$, should be equal to $\mathbf{t}$:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{x}_{\text{p}}\mathbf{Z}_{\text{p}} + \mathbf{x}_{\text{fdof}}\mathbf{Z}_{\text{fdof}} \\ &= \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{d}}(\mathbf{M}_{\text{QP}} + \mathbf{Z}_{\text{fdof}}^{-1*})\mathbf{Z}_{\text{fdof}} \\ &= \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{d}}(\mathbf{M}_{\text{QP}}\mathbf{Z}_{\text{fdof}} + \mathbf{I})\end{aligned}$$

From Equation (9.18), $\mathbf{t}$ yields

$$\mathbf{t} = \mathbf{t}_{\text{p}} + \mathbf{t}_{\text{d}}\underbrace{(\mathbf{M}_{\text{QP}}\mathbf{Z}_{\text{fdof}} + \mathbf{I})}_{\bar{\mathbf{M}}_{\text{t}}}$$

From the previous equations $\mathbf{t}^* = \mathbf{t}$, then the methodology is consistent with using $\mathbf{t}$ to perform predictions.