



Master thesis

# Parametric model order reduction: a projection and interpolation approach

**Jules Matz**

Supervised by:

- MIPS Laboratory -

Benjamin Mourllion

Abderazik Birouche

- Politecnico di Torino -

Diego Regruto

September 2017

# Contents

<b>1</b>	<b>System Modelling</b>	<b>4</b>
1.1	State-space representation . . . . .	4
1.2	State transformation . . . . .	5
1.3	Stability . . . . .	5
1.3.1	Lyapunov based method . . . . .	5
1.4	Reachability . . . . .	6
1.5	Observability . . . . .	7
1.6	Balanced Realisation . . . . .	8
1.7	System norms . . . . .	9
1.8	LPV model . . . . .	9
1.9	Parametric LTI model . . . . .	10
1.9.1	LFT representation . . . . .	10
1.9.2	Discretization of the parameter space . . . . .	10
<b>2</b>	<b>LTI Model Order Reduction</b>	<b>12</b>
2.1	Projection framework . . . . .	12
2.1.1	Projection in mathematics . . . . .	12
2.1.2	Petrov-Galerkin projection . . . . .	16
2.2	Balanced truncation . . . . .	18
2.2.1	Concept . . . . .	18
2.2.2	Method . . . . .	19
2.2.3	Bound on the $H_\infty$ -norm of the error . . . . .	20
2.3	Conclusion . . . . .	20
<b>3</b>	<b>Parametric LTI Model Order Reduction</b>	<b>21</b>
3.1	Sampling of local models . . . . .	21
3.2	Reduction of the local models . . . . .	22
3.2.1	Reduction via balanced truncation . . . . .	23
3.2.2	Reduction via adapted balanced truncation . . . . .	23
3.3	Adaptation of the reduced local models . . . . .	25
3.3.1	Subspace $\mathcal{R}$ . . . . .	25

3.3.2	Generalized coordinates . . . . .	26
3.4	Interpolation . . . . .	28
3.4.1	Vector space where to interpolate . . . . .	28
3.4.2	Interpolation method . . . . .	29
<b>4</b>	<b>Derived pMOR Method and Results</b>	<b>31</b>
4.1	Derived method . . . . .	31
4.2	Results on academic examples . . . . .	34
	<b>Appendices</b>	<b>41</b>
<b>A</b>	<b>LFT representation</b>	<b>41</b>
<b>B</b>	<b>Other MOR techniques</b>	<b>43</b>
B.1	Modal truncation . . . . .	43
B.1.1	Concept . . . . .	43
B.1.2	Method . . . . .	43
B.1.3	Bound on $\ \Sigma - \Sigma_r\ _{\text{H}\infty}$ . . . . .	44
B.2	Balanced residualization . . . . .	44
<b>C</b>	<b>Proof for the mean subspace <math>\mathcal{R}</math></b>	<b>46</b>

# Introduction

Numerical simulations of dynamical systems are increasingly used in numerous applications. To enhance this process by reducing the simulation time and/or the needed processing power, model order reduction aims to approximate a complex model with a simpler one, easier (faster and/or requiring less computations) to simulate. Model order reduction for Linear Time-Invariant (LTI) systems has become a quite mature field, and now researchers are focusing on more complex models, such as nonlinear models, Time-Varying models or parameterized models. This work studies the so-called parametric Model Order Reduction (pMOR), where the reduction of models depends on some parameters (constant in time). This parameter dependency can modelize a parameter uncertainty, a different configuration of the system, etc.

This work is organized as follow: a first chapter reviews the system modelling and introduces usefull tools for model order reduction and parameterized models. The second chapter studies LTI model order reduction, particularly the balanced truncation, a special type of model order reduction based on projection. Finally the last chapter is devoted to parametric model order reduction and uses of the notions and techniques from the previous chapters.

# Chapter 1

## System Modelling

### Introduction

This first chapter introduces definitions and notions that are useful for the present work. These are general definitions and can be found in most books related to system analysis (e.g. [1], [2]). Notably the concepts of reachability and observability Gramians, and parametric LTI model are important here.

### 1.1 State-space representation

A LTI system  $\Sigma$  is defined by a set of differential equations. It can be described by a state-space representation  $\mathbb{S}$ :

$$\mathbb{S}_{\text{LTI}} : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) & \text{state equation} \\ y(t) = Cx(t) + Du(t) & \text{output equation} \end{cases} \quad (1.1)$$

$A \in \mathbb{R}^{n \times n}$ the state (or system) matrix		$u(t) \in \mathbb{R}^m$ the input vector
$B \in \mathbb{R}^{n \times m}$ the input matrix		$y(t) \in \mathbb{R}^p$ the output vector
$C \in \mathbb{R}^{p \times n}$ the output matrix		$x(t) \in \mathbb{R}^n$ the state vector
$D \in \mathbb{R}^{p \times m}$ the feedforward matrix		

The states  $\{x_i, i = 1, \dots, n\}$  are internal variables of the system, and  $n$  is the system dimension.

Equation (1.1) is also written  $\mathbb{S} = \left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$

**Solution to the state equation** a system with initial state  $x_0$  at a time  $t_0$  and an input  $u$  evolves accordingly to

$$\phi(u, x_0, t) = e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (1.2)$$

**Minimal realisation** a state-space representation is also called a *realisation* of the system  $\Sigma$ . It is said to be *minimal* if it has the smallest possible number of states (dimension  $n$ ).

## 1.2 State transformation

A LTI system has an infinity of state-space representations, in which the state vector is expressed in different bases. Applying a state transformation  $x = T\tilde{x}$  to (1.1) and pre-multiplying by  $T^{-1}$  results in a different (but equivalent) state-space representation of the system  $\Sigma$  in which the matrices  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$  and  $\tilde{D}$  are now expressed according to a new state basis.

$$\left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \xrightarrow{x=T\tilde{x}} \left[ \begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right] = \left[ \begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array} \right]$$

## 1.3 Stability

A system  $\Sigma$  is *stable* if for any initial condition  $x_0$ , its state vector  $x(t)$  is bounded when the system autonomously evolves. Moreover,  $\Sigma$  is *asymptotically stable* if for any initial condition  $x_0$ ,  $x(t)$  tends to zero when the system autonomously evolves. It can be seen from (1.2) that  $x(t)$  being bounded or going to zero depends on the system matrix  $A \in \mathbb{R}^{n \times n}$ .

The system  $\Sigma$  is

**stable**  $\iff \Re(\lambda_i(A)) \geq 0 \forall i$  and pure imaginary eigenvalues have multiplicity one.

**asymptotically stable**  $\iff \Re(\lambda_i(A)) < 0 \forall i$  ( $A$  is *Hurwitz*).

### 1.3.1 Lyapunov based method

The Lyapunov stability consists in showing the stability of a system by finding a positive function  $V(x(t)) > 0$  strictly decreasing in time when no input acts on the system. Constructing  $V$  as a quadratic function,  $V(x(t)) = x(t)^T P x(t)$ , with  $P$  a constant symmetric positive definite matrix, the derivative of  $V$  is

$$\begin{aligned} \dot{V}(x(t)) &= \dot{x}(t)^T P x(t) + x(t)^T P \dot{x}(t) \\ &= x(t)^T A^T P x(t) + x(t)^T P A x(t) \\ &= x(t)^T (A^T P + P A) x(t) \end{aligned}$$

hence for  $V$  to be a strictly decreasing function,  $P$  must verify

$$A^T P + P A \prec 0$$

that is, for any matrix  $Q \succeq 0$ ,  $P$  must satisfy the Lyapunov equation

$$A^T P + P A + Q = 0.$$

If such a matrix  $P$  exists, the system is asymptotically stable.

## 1.4 Reachability

Reachability characterizes which states the system can attain.

### Definition 1: Reachability [1]

A state  $x$  is *reachable* from the origin if there exists an input  $u(t)$ , of finite energy, and a finite time  $T < \infty$ , such that

$$x = \phi(u, 0, T)$$

The *reachable subspace*  $\mathbb{X}^{\text{reach}}$  is the set of all reachable states. The system is reachable if  $\mathbb{X}^{\text{reach}} = \mathbb{R}^n$ .

The *reachability matrix* is

$$\mathcal{R}(A, B) = [ B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B ]$$

### Definition 2: Reachability Gramian

The infinite reachability Gramian

$$W_r = \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau \succeq 0 \quad (1.3)$$

is solution to the reachability Lyapunov equation

$$A W_r + W_r A^T + B B^T = 0$$

The reachability Gramian  $W_r$  is related to the energy necessary to drive the system to a state. In fact  $x^T W_r^{-1} x$  is the minimum amount of energy required to steer the system from  $x_0 = 0$  at time  $t = 0$  to  $x$  at  $t = \infty$ . Hence  $W_r$  represents how much input energy is required to move a state.

**Theorem 1.** *The reachable subspace is given by*

$$\mathbb{X}^{\text{reach}} = \text{span } W_r = \text{span } \mathcal{R}(A, B) \quad , \forall t > 0$$

**Theorem 2.** *The following statements are equivalent:*

1. *The system is completely reachable.*
2. *The reachability Gramian is positive definite,  $W_r \succ 0 \forall t$ .*
3. *The reachability matrix is full rank,  $\text{rank } \mathcal{R}(A, B) = n$ .*

## 1.5 Observability

A state is *observable* if it can be recovered knowing the output  $y(t)$  and the input  $u(t)$  at a finite time  $T$ .

**Definition 3: Observability [1]**

The system is completely observable if  $\mathbb{X}^{\text{obs}} = \mathbb{R}^n$ .  
The *observability matrix* is

$$\mathcal{O}(C, A) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

**Definition 4: Observability Gramian**

The infinite observability Gramian

$$W_o = \int_0^\infty e^{A^T \tau} C^T C e^{A \tau} d\tau \succeq 0 \quad (1.4)$$

is solution to the observability Lyapunov equation

$$A^T W_o + W_o A + C^T C = 0$$

The observability Gramian  $W_o$  is related to the energy obtained at the output of the system. In fact  $x^T W_o x$  is the obtained energy on the output when the system autonomously evolves from the initial state  $x$  at  $t = 0$  to 0 at  $t = \infty$ . Hence  $W_o$  represents how much the move of a state affects the output energy.



**Theorem 3.** *The unobservable subspace is given by*

$$\mathbb{X}^{\text{unobs}} = \ker W_o = \ker \mathcal{O}(C, A) \quad , \forall t > 0$$

**Theorem 4.** *The following statements are equivalent:*

1. *The system is completely observable.*
2. *The observability Gramian is positive definite,  $W_o \succ 0 \forall t > 0$ .*
3. *The observability matrix is full rank,  $\text{rank } \mathcal{O}(C, A) = n$ .*

## 1.6 Balanced Realisation

### Expression of the Gramians after state transformation

The Gramians obtained after a state transformation  $T$  are

$$\begin{aligned}\hat{W}_r &= T^{-1}W_rT^{-T} \\ \hat{W}_o &= T^TW_oT\end{aligned}$$

$W_r^{-1}$  and  $W_o$  undergo a *congruent* transformation ( $\diamond \mapsto T^T \diamond T$ ).

The product of the two Gramians is

$$\hat{W}_r\hat{W}_o = T^{-1}W_rW_oT \tag{1.5}$$

### Hankel singular values

The Hankel singular values describe how much energy each state conveys from the input to the output.

#### Definition 5

The Hankel singular values are defined as the square roots of the eigenvalues of the product  $W_rW_o$ .

$$\sigma_i = \sqrt{\lambda_i(W_rW_o)}$$

It follows from (1.5) that the Hankel singular values are invariant under state transformation. They are a characteristic of the system  $\Sigma$ , related to the energy transiting in the system.

## Balanced realisation

### Definition 6

To *balance*  $\mathbb{S}$  is to apply a state transformation  $T$  such that the obtained Gramians are equal and diagonal, with their diagonal elements (the Hankel singular values) in decreasing order.

$$\tilde{W}_r = \tilde{W}_o = \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$$

where the  $\sigma_i$  are the Hankel singular values of the system.

If  $\mathbb{S}$  is balanced, that means the energy transitting from the input to the output through the state  $x_1$  is greater than the energy transitting through  $x_2$ , which is greater than the one transitting through  $x_3$ , etc.

## 1.7 System norms

System norms characterize the amplification induced by the system (the ratio between the input  $u$  and output  $y$ ), measured with some signal norms. Particularly, the  $H_\infty$  norm of the system  $\Sigma$  with transfer function  $G(s)$  is

$$\|\Sigma\|_{H_\infty} = \sup_{\omega} \sigma_{\max}(G(j\omega))$$

and corresponds to the maximal amplification over all frequencies. Also the  $H_2$  norm is useful

$$\|\Sigma\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{+\infty} |G(j\omega)|^2 d\omega}$$

corresponds to the  $L_2$  norm of the impulse response. Also it is the RMS value of the system response to a white noise.

## 1.8 LPV model

A linear parameter-varying model has its matrices  $A, B, C$  and  $D$  depending on some time-varying parameters. These parameters are represented by a vector function of time  $\rho(t)$ . Hence a state-space representation of a LPV model

$$\mathbb{S}_{\text{LPV}} = \left[ \begin{array}{c|c} A(\rho(t)) & B(\rho(t)) \\ \hline C(\rho(t)) & D(\rho(t)) \end{array} \right]$$

with  $\rho(t)$  the vector of uncertain parameters, function of time.

## 1.9 Parametric LTI model

Parametric LTI models are a particular case of LPV models. A parametric LTI model has its matrices  $A, B, C$  and  $D$  depending on parameters that are constant in time. Its state-space representation has the form

$$\mathbb{S} = \left[ \begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array} \right]$$

with  $\rho$  the vector of uncertain parameters, having bounds  $\rho_{min} < \rho < \rho_{max}$ .

It can be used for instance to model a system in different operating conditions, to model parameter uncertainties, or to model a set of variants of a system (e.g. for geometric optimization).

parameter-dependant models can be described in several ways, for instance using the LFT representation or by discretization of the parameter space in a set of LTI models.

### 1.9.1 LFT representation

Linear Fractional Transformation (LFT) is a possible representation of a parametric model [3]. It is often used when describing a system with uncertainties. Appendix A describe this representation, although it is not used in the chapters to come.

### 1.9.2 Discretization of the parameter space

The parameter space is discretized into a set of local LTI models. Then the model for a particular parameter value is obtained by interpolation of the sampled local LTI models. A possibility is to sample only models for the minimal and maximal values of the parameters (polytopic representation). More models can be sampled for better accuracy, for instance distributed along a grid in the parameter space. There exists several techniques for choosing how many and where in the parameter space to sample the local models, it will be discussed in section 3.1.

#### Polytopic representation

The parameter vector  $\rho$  lies in a convex polytope in the parameter vector space  $\mathcal{P}$ . A system with  $p$  parameters has  $2^p$  vertices  $\omega_i$  corresponding to all possible combinations of minimum and maximum values for each parameter. The matrices  $A(\rho), B(\rho), C(\rho)$  and  $D(\rho)$  are expressed as linear combination

of the matrices at each vertex.

$$\left[ \begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array} \right] = \sum_{i=1}^{2^p} \alpha_i(\rho) \left[ \begin{array}{c|c} A(\omega_i) & B(\omega_i) \\ \hline C(\omega_i) & D(\omega_i) \end{array} \right]$$

where  $\sum_{i=1}^{2^p} \alpha_i(\rho) = 1$  ,  $\alpha_i(\rho) \geq 0$ , and  $\alpha_i(\omega_j) = \delta_{ij} \forall i, j \in [1; 2^p]$

$$\alpha_i(\rho) = \prod_{j=1}^p \frac{|\rho_j - \omega_i^c(j)|}{\bar{\rho}_j - \underline{\rho}_j}$$

$\omega_i^c$  denotes the complement of  $\omega_i$ :  $\omega_i^c(j) = \begin{cases} \bar{\rho}_j & \text{if } \omega_i(j) = \underline{\rho}_j \\ \underline{\rho}_j & \text{if } \omega_i(j) = \bar{\rho}_j \end{cases}$

## Grid representation

Similar to the polytopic representation, grid representation consists of sampling a set of models over the parameter space equally distant from one another to build a set of local LTI models. The model  $\Sigma(\rho)$  is then computed as an interpolation between the set of models  $\{\Sigma_i : i = 1, \dots, l\}$  obtained from the sampled parameters. A higher number of local models gives better accuracy at the cost of more memory storage. The choice of the number and location (over the parameter space) of the sampled parameter can be guided by different methods. It is possible (although it can be computationally expensive) to sample a very high number of parameters either equally distant from one another or randomly chosen. Another approach is to select the sampled parameters using a sensitivity analysis.

## Conclusion

A number of important definitions related to LTI models and to LPV/parametric models have been reviewed. These notions and tools are the base of the model order reduction approaches studied in this work.

# Chapter 2

## LTI Model Order Reduction

### Introduction

Model order reduction (MOR) aims at approximating a high order model with a simpler model. The complexity of a model is indicated by its number of states, hence the purpose of MOR is to approximate a model of  $n$  states with a reduced model of  $k \ll n$  states. In this work only projection-based MOR techniques (specifically balanced truncation) are considered, which can be seen as projections of the original state vector  $x \in \mathbb{R}^n$  onto a subspace  $\mathcal{V} \subset \mathbb{R}^n$ . Then the obtained reduced order model can be used for the intended simulations, having its state vector  $\hat{x}$  evolving in the subspace  $\mathcal{V}$  of dimension  $k \ll n$ .

The reduced order model can be simulated in an acceptable time, however reducing the state vector to a lower dimensional space, an error is introduced between the original model and the reduced one. Fortunately, for balanced truncation and some other techniques, there exists bounds for this error. Although only balanced truncation is useful for the present work, also the modal truncation and balanced residualization are presented in appendix B.

### 2.1 Projection framework

This first section deals with the concept of projection from a mathematical theory view, then projection as a tool for MOR is introduced.

#### 2.1.1 Projection in mathematics

**Definition 7: Projection**

A linear operator  $\Pi$  is a projection if  $\Pi^2 = \Pi$  (idempotence property).

**Definition 8: Adjoint operator**

Let  $E$  and  $F$  be two vector spaces having inner products  $\langle \cdot, \cdot \rangle_E$  and  $\langle \cdot, \cdot \rangle_F$ , the adjoint of a linear operator  $A : E \rightarrow F$  is defined as the operator  $A^* : F \rightarrow E$  such that

$$\forall e \in E, \forall f \in F, \quad \langle Ae, f \rangle_F = \langle e, A^*f \rangle_E$$

**Lemma 1. Expression of a projection**

Any projection  $\Pi$  from a vector space  $F$  onto a vector space  $E$  can be expressed using a linear operator  $A : E \rightarrow F$  and its adjoint  $A^* : F \rightarrow E$ . The projection is done orthogonally onto  $\text{span } A$ .

$$\Pi = A(A^*A)^{-1}A^*$$

where  $A : E \rightarrow F$  is a linear operator and  $A^* : F \rightarrow E$  is its adjoint.

*Proof.* Any vector in  $F$  can be decomposed as two vectors in  $\text{span } A$  and  $(\text{span } A)^\perp$ . In the following it is shown that by application of the projection, vectors in  $\text{span } A$  are left unchanged while vector in  $(\text{span } A)^\perp$  are set to zero.

- for any vector  $y \in \text{span } A$ ,

$$y \in \text{span } A \implies \exists x \in E, y = Ax$$

and the projection of  $y$  is

$$\begin{aligned} \Pi y &= A(A^*A)^{-1}A^*y \\ &= A(A^*A)^{-1}A^*Ax \\ &= Ax \\ &= y \end{aligned}$$

- for any vector  $y \in (\text{span } A)^\perp$ , for any nonzero vector  $x \in E$ ,

$$y \in (\text{span } A)^\perp \implies \langle y, Ax \rangle = 0$$

by definition of the adjoint, and since  $x \neq 0$ ,

$$\langle A^*y, x \rangle = 0 \implies y \in \ker A^*$$

and the projection of  $y$  is

$$\Pi y = A(A^*A)^{-1} \underbrace{A^*y}_0 = 0$$

□

The projection is characterized by the vector spaces  $E$  and  $F$  and the considered inner products  $\langle \cdot, \cdot \rangle_E$  and  $\langle \cdot, \cdot \rangle_F$ . The projection obtained using the standard inner product is an *orthogonal* projection, while the projection obtained using a weighted inner product is an *oblique* projection.

The orthogonal projection is a projection onto a subspace orthogonally to this same subspace, while the oblique projection is a projection onto a subspace orthogonally (in the sense of the standard inner product) to another subspace.

**Lemma 2.** *Orthogonal projection*

Defining  $\langle \cdot, \cdot \rangle_E$  and  $\langle \cdot, \cdot \rangle_F$  as the standard inner products,

$$\begin{aligned} \forall x, y \in E, \langle x, y \rangle_E &= x^T y \\ \forall x, y \in F, \langle x, y \rangle_F &= x^T y \end{aligned}$$

then the adjoint of a linear operator  $A : E \rightarrow F$  is defined by

$$\begin{aligned} \forall e \in E, f \in F, \langle Ae, f \rangle_F &= \langle e, A^*f \rangle_E \\ (Ae)^T f &= e^T A^*f \\ e^T A^T f &= e^T A^*f \end{aligned}$$

hence  $A^* = A^T$ , the orthogonal projection is described by

$$\Pi = A(A^T A)^{-1} A^T. \tag{2.1}$$

**Lemma 3.** *Oblique projection*

Defining  $\langle \cdot, \cdot \rangle_E$  and  $\langle \cdot, \cdot \rangle_F$  as a weighted inner product,

$$\begin{aligned} \forall x, y \in E, \langle x, y \rangle_E &= x^T Q_E y \\ \forall x, y \in F, \langle x, y \rangle_F &= x^T Q_F y \end{aligned}$$

where  $Q_E, Q_F$  are positive definite matrices.

Then the adjoint of a linear operator  $A : E \rightarrow F$  is defined by

$$\begin{aligned} \forall e \in E, f \in F, \langle Ae, f \rangle_F &= \langle e, A^*f \rangle_E \\ (Ae)^T Q_F f &= e^T Q_E A^*f \\ e^T A^T Q_F f &= e^T Q_E A^*f \end{aligned}$$

hence  $A^* = Q_E^{-1}A^TQ_F$  the orthogonal projection is described by

$$\begin{aligned}\Pi &= A(A^*A)^{-1}A^* \\ &= A(Q_E^{-1}A^TQ_FA)^{-1}Q_E^{-1}A^TQ_F \\ &= A(A^TQ_FA)^{-1}A^TQ_F\end{aligned}$$

**MOR projection framework** The purpose of projection-based MOR techniques is to project the original state vector  $x \in \mathbb{R}^n$  onto a subspace  $\mathcal{V} \subset \mathbb{R}^n$ ,  $\dim \mathcal{V} = k \ll n$  orthogonally to a subspace  $\mathcal{W}$ ,  $\dim \mathcal{W} = k$ . Thus the previously considered inner product spaces  $E$  and  $F$  are replaced by  $\mathcal{V}$  and  $\mathbb{R}^n$ , respectively.

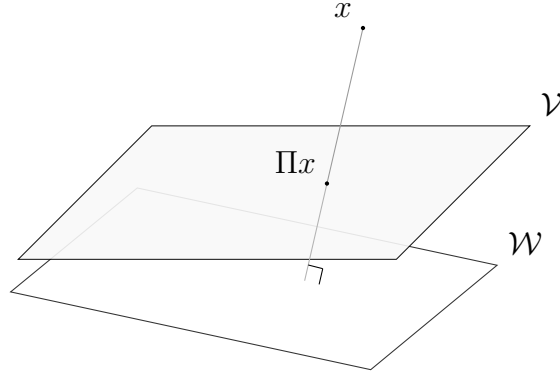


Figure 2.1: Oblique projection of a 3d vector  $x$  onto a subspace  $\mathcal{V}$

**Matrix expression of an orthogonal projection** For the computation, the projection has to be expressed as matrix operations on  $\mathbb{S}$ . Representing  $A : E \rightarrow F$  by a matrix  $V \in \mathbb{R}^{n \times k}$  spanning the subspace  $\mathcal{V}$ , the orthogonal projection onto  $\mathcal{V}$  is expressed as

$$\Pi = V(V^TV)^{-1}V^T \quad (2.2)$$

**Matrix expression of an oblique projection** Representing  $A : E \rightarrow F$  by  $V \in \mathbb{R}^{n \times k}$  and  $A^TQ_F : F \rightarrow E$  by  $W^T \in \mathbb{R}^{k \times n}$ , with  $W \in \mathbb{R}^{n \times k}$  spanning the subspace  $\mathcal{W}$ , the oblique projection onto  $\mathcal{V}$  orthogonally to  $\mathcal{W}$  (or along  $\mathcal{W}^\perp$ ) is expressed as

$$\Pi = V(W^TV)^{-1}W^T. \quad (2.3)$$



By not directly introducing  $W$  and denoting  $Q = Q_F$  another (similar) expression is obtained

$$\Pi = V(V^T Q V)^{-1} V^T Q. \quad (2.4)$$

### 2.1.2 Petrov-Galerkin projection

Model order reduction can be achieved by projecting the state vector  $x$  onto a subspace  $\mathcal{V}$  orthogonally to another subspace  $\mathcal{W}$ , as illustrated in figure 2.1.

The subspaces  $\mathcal{V}$  and  $\mathcal{W}$  are spanned by the matrices  $V$  and  $W$ . The objective of any MOR technique using projections is to choose appropriate matrices  $V, W \in \mathbb{R}^{n \times k}$  such that the reduced order model approximates well the original model

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

The state vector is described by  $x \in \mathbb{R}^n$ .

The reduced state vector is expressed in  $\mathbb{R}^n$  by  $\tilde{x}$  and in  $\mathbb{R}^k$  by  $\hat{x}$ .

The relation between  $\tilde{x}$  and  $\hat{x}$  is  $\tilde{x} = V\hat{x}$ .

From Figure 2.1, it seems that  $\tilde{x} = \Pi x$ . This is the case for a simple vector but when projecting a model, there is also an error in the subspace  $\mathcal{V}$ , because not only the state  $x$  is restrained to  $\mathcal{V}$  but also its dynamics  $\dot{x}$ . While Figure 2.1 illustrates the oblique projection of a vector, Figure 2.2 gives a more accurate insight into the oblique projection of a dynamical system. The state vector is decomposed as follow

$$x = \tilde{x} + \epsilon \quad (2.5)$$

$$\epsilon = \epsilon_{\parallel} + \epsilon_{\perp} \quad (2.6)$$

$\epsilon$  is the error made approximating  $x$  by  $\tilde{x}$

$\epsilon_{\perp}$  is the neglected dynamics (component of  $\epsilon$  orthogonal to  $\mathcal{W}$ )

$\epsilon_{\parallel}$  is error induced by the neglected dynamics (component of  $\epsilon$  on  $\mathcal{V}$ ).

As a consequence, these two equations with the projection of  $x$

$$x = \Pi x + \epsilon_{\perp} \quad (2.7)$$

$$\Pi x = \tilde{x} + \epsilon_{\parallel} \quad (2.8)$$

Given an original state equation

$$\dot{x} = Ax + Bu \quad (2.9)$$

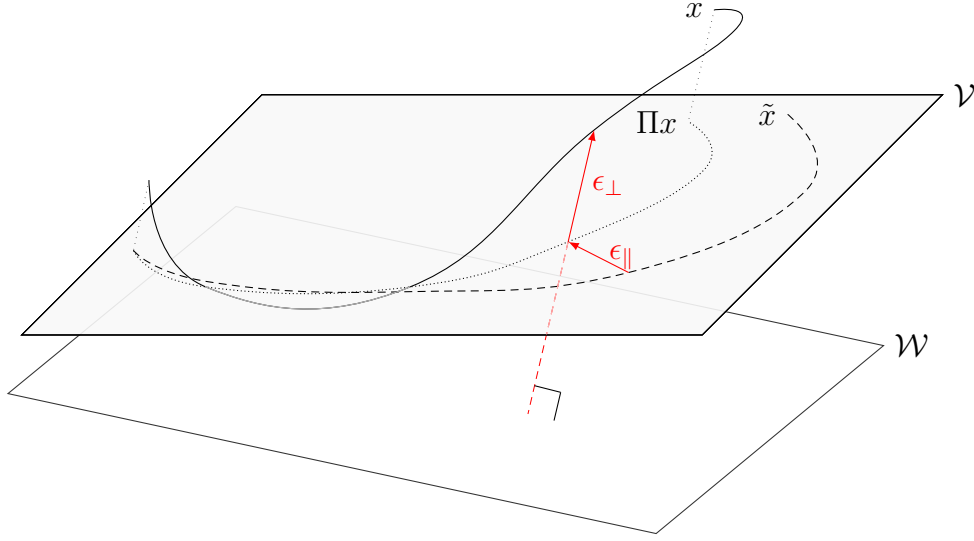


Figure 2.2: Petrov-Galerkin projection

Projecting it onto  $\mathcal{V}$  orthogonally to  $\mathcal{W}$ , then using (2.5) and (2.8)

$$\begin{aligned}\Pi\dot{x} &= \Pi Ax + \Pi Bu \\ \dot{\tilde{x}} + \dot{\epsilon}_{\parallel} &= \Pi A\tilde{x} + \Pi Bu + \Pi A\epsilon\end{aligned}$$

Therefore the reduced order model expressed in  $\mathbb{R}^n$  is

$$\begin{aligned}\dot{\tilde{x}}(t) &= \Pi A\tilde{x}(t) + \Pi Bu(t) \\ y(t) &= C\tilde{x}(t) + Du(t)\end{aligned}\tag{2.10}$$

The error dynamics is expressed as:

$$\dot{\epsilon}_{\parallel} = \Pi A\epsilon_{\parallel} + \Pi A\epsilon_{\perp}$$

hence the error  $\epsilon_{\parallel}$  behaves as a dynamical LTI system having for input the neglected dynamics  $\epsilon_{\perp}$ .

By introducing (2.9) into the derivative of (2.7) the differential equation related to the error  $\epsilon_{\perp}$  is obtained

$$\dot{\epsilon}_{\perp} = (I_n - \Pi)Ax + (I_n - \Pi)Bu\tag{2.11}$$

where  $(I_n - \Pi)$  is the projection onto the neglected subspace  $\mathcal{W}^{\perp}$ .

Finally, replacing  $\tilde{x} = V\hat{x}$  and  $\Pi = V(W^T V)^{-1}W^T$  into (2.10),

$$\begin{aligned}V\dot{\hat{x}}(t) &= V(W^T V)^{-1}W^T AV\hat{x}(t) + V(W^T V)^{-1}W^T Bu(t) \\ y(t) &= CV\hat{x}(t)\end{aligned}$$

and premultiplying by  $W^T$  the expression of the reduced order model in  $\mathbb{R}^k$  is obtained

$$\begin{aligned}\hat{\dot{x}}(t) &= (W^T V)^{-1} W^T A V \hat{x}(t) + (W^T V)^{-1} W^T B u(t) \\ y(t) &= C V \hat{x}(t)\end{aligned}$$

yielding the matrices of the reduced order model

$$\begin{cases} \hat{A} = (W^T V)^{-1} W^T A V \\ \hat{B} = (W^T V)^{-1} W^T B \\ \hat{C} = C V \end{cases}$$

If  $V$  and  $W$  are bi-orthogonal (with  $W^T V = I_k$ ), then the matrices simplify to

$$\begin{cases} \hat{A} = W^T A V \\ \hat{B} = W^T B \\ \hat{C} = C V \end{cases}$$

## 2.2 Balanced truncation

Also known as Lyapunov balanced truncation. This method was introduced in system theory by Moore in 1981 [4].

### 2.2.1 Concept

The idea behind balanced truncation is to remove the states which have small influence on the energy transiting from the input  $u$  to the output  $y$ .

To study the relative importance of each state, we use the Gramians. The reachability Gramian  $W_r$  tells us how sensitive is each state to the input  $u$  (in term of energy). Similarly, the observability Gramian  $W_o$  tells us how sensitive is the output  $y$  to each state.

To sort the states, we have to examine separately how reachable and how observable is each state, so we need the Gramians to be diagonal. Furthermore to decide which states are the more important, we need them to be as observable as they are reachable, so we have to equalize the two Gramians. (Since we cannot decide which one of two states, one easily reachable and hardly observable, one hardly reachable and easily observable, is the most important).

### 2.2.2 Method

We balance the system  $\Sigma$ , that means we find a minimal realisation of  $\Sigma$  where its reachability and observability Gramians are equal and diagonal:

$$\left[ \begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array} \right] = \left[ \begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right], \text{ such that } \begin{cases} \tilde{W}_r = T^{-1}W_rT^{-T} = \Sigma \\ \tilde{W}_o = TW_oT^T = \Sigma \end{cases}$$

with  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ , where  $\sigma_i \geq \dots \geq \sigma_k > \sigma_{k+1} \geq \dots \geq \sigma_n > 0$  where the  $\sigma_i$ , the Hankel singular values of the system, are sorted.

Finally, we neglect the least important states  $x_{k+1}, \dots, x_n$ .

**Algorithm for computing the transformation  $T$ :** this efficient algorithm was introduced by Laub in 1987 [5]. It efficiently computes the balancing state-space transformation from the Cholesky factors of the Gramians (The actual Gramians never need to be computed). The Cholesky factors are computed from the two Lyapunov equations, by means of algorithms (see [6]).

1. Compute the Cholesky decomposition of the Gramians

$$W_r = L_r L_r^T, \quad W_o = L_o L_o^T$$

2. Compute the singular value decomposition:

$$L_o^T L_r = U \Lambda N^T$$

3. We then have

$$T = L_r N \Lambda^{-1/2} \quad \text{and} \quad T^{-1} = \Lambda^{-1/2} U^T L_o^T$$

Replacing  $T$  in the expressions  $T^{-1}W_rT^{-T}$  and  $T^TW_oT$  we can easily verify  $\tilde{W}_r = \tilde{W}_o = \Sigma$ .

Note that at step 2, it is possible to compute the SVD of  $L_o^T L_r$  instead of  $L_r^T L_o$ , it only interchanges  $U$  and  $V$  in the ensuing computations.

#### Direct method

It is possible to directly compute the matrices  $V$  and  $W^T$  for the projection, without first balancing the state-space representation and then truncating it.

To do so using the algorithm of Laub [5] it is needed the computation of the singular value decomposition

$$L_o^T L_r = [ U_1 \quad U_2 ] \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} [ N_1 \quad N_2 ]^T$$

( $N$  is used instead of  $V$  to avoid confusion with the projection matrix). The projection matrices are then

$$V = L_r N_1 \Lambda_1^{-1/2}$$

$$W^T = \Lambda_1^{-1/2} U_1^T L_o^T$$

and the projection achieving balanced truncation is

$$\Pi = \underbrace{L_r N_1 \Lambda_1^{-1/2}}_V \underbrace{\Lambda_1^{-1/2} U_1^T L_o^T}_{W^T} \quad (2.12)$$

(the projection is given by  $\Pi = VW^T$  instead of  $\Pi = V(W^T V)W^T$  since  $V$  and  $W$  are biorthogonal,  $W^T V = I_k$ ).

### 2.2.3 Bound on the $H_\infty$ -norm of the error

An interesting feature of the balanced truncation is that the  $H_\infty$ -norm of the difference between the original system  $\Sigma$  and the reduced one  $\Sigma_r$  is upper bounded by twice the sum of the neglected Hankel singular values.

$$\|\Sigma - \Sigma_r\|_{H_\infty} \leq 2(\sigma_{k+1} + \dots + \sigma_n)$$

## 2.3 Conclusion

MOR via projections was introduced, and the balanced truncation technique was studied in this chapter. In the next chapter these techniques are used again and extended to the reduction of parametric models.

# Chapter 3

## Parametric LTI Model Order Reduction

### Introduction

This chapter is at the heart of this work, it addresses the problem of parametric model order reduction (pMOR). That is the approximation of a parametric model of order  $n$  by a model of order  $k \ll n$  with the parameter dependency of the original model (the parameters are constant and their range is known).

The approach considered here is to first sample a set of LTI models over the parameter space, then to reduce and adapt them (all this being *offline computations*) for finally being able to compute via an interpolation, possibly in real-time, a reduced order model (ROM) at any desired parameter value (this last part being the *online computations*).

Each of these steps can be tackled in many ways, what [7] pointed out as the *degrees of freedom*. The choices to be made depends on the considered system, the memory storage and processing power available, as well as specifications like real-time capability, maximum error allowed, etc.

In this work, a few techniques are investigated and compared, mostly techniques concerning the reduction and adaptation of the local models and their interpolation.

### 3.1 Sampling of local models

The parametric model  $\Sigma(\rho)$  is discretized over the parameter space into a set of LTI models  $\{\Sigma_i : i = 1, \dots, l\}$ . As detailed in [8], a few different methods can be used to choose at which parameter values to sample. The most intu-

itive is to construct a grid (or a sparse grid) over the parameter space and sample a model at all grid points, another possibility is to randomly sample many models over the parameter space. Although these two methods can be effective for models with few parameters and limited parameter range, they are unsuitable for models with many parameters ( $> 10$ ), so other techniques are used, which better represent the parameter variations while sampling fewer models.

An iterative method that can be used is parameter sampling via greedy search, it is an algorithm that searches for the point on the parameter space where the error between the original model and the reduced one is maximal. Then it samples a model at this point, uses it to recompute the reduced model, and finds the new point where the error is now maximal, samples it, etc. until the error is acceptable.

Another interesting method is to use local sensitivity analysis to determine where to sample, consequently more samples are placed where the system behavior depends greatly on the parameter variations and fewer where the parameter variations have less impact on the system.

Although parameter sampling via local sensitivity analysis seems a very interesting technique, only grid-based sampling is used in this work, for reasons of simplicity.

## 3.2 Reduction of the local models

All local models  $\{\Sigma_i : i = 1, \dots, l\}$  are reduced to the same order  $k \ll n$  by means of a projection-based model order reduction technique. In this work, balanced truncation or a closely related method are considered. Recalling what has been detailed in chapter 2, model order reduction by projection of a model  $\Sigma$  restricts the original state  $x \in \mathbb{R}^n$  to a lower dimensional subspace  $\mathcal{V}$  of dimension  $k$ . Now since the local models  $\Sigma_i$  are reduced separately, their reduced states  $\hat{x}_i$  do not necessarily lie in a same subspace, but in different subspaces  $\mathcal{V}_i$ .

The purpose of pMOR is to interpolate the state-space representations  $\{\hat{\mathbb{S}}_i : i = 1, \dots, l\}$  of the reduced local models, for this, the  $\hat{\mathbb{S}}_i$  need to be compatible in some sense, that is, the states  $\hat{x}_i$  have to be expressed in the same basis. One approach to this problem is to separately reduce the local models by balanced truncation, then to adapt them with a common basis. Another approach is to reduce the local models by projection onto a single subspace, so the adaptation process is no more needed.

### 3.2.1 Reduction via balanced truncation

Each local model  $\Sigma_i$  is separately reduced via balanced truncation (see section 2.2). This projection is expressed in accordance with the algorithm of Laub [5] as

$$\Pi_i = \underbrace{L_r N_1 \Lambda_1^{-1/2}}_{V_i} \underbrace{\Lambda_1^{-1/2} U_1^T L_o^T}_{W_i^T} \quad (3.1)$$

where  $L_r$  and  $L_o$  are respectively the Cholesky factors of the reachability and observability Gramians of the model  $\Sigma_i$ .

$N_1, U_1 \in \mathbb{R}^{n \times k}$  and  $\Lambda_1 \in \mathbb{R}^{k \times k}$  are obtained from the SVD

$$L_o^T L_r = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \begin{bmatrix} N_1 & N_2 \end{bmatrix}^T.$$

The reduced models are obtained as

$$\hat{\Sigma}_i = \begin{cases} \dot{\hat{x}}_i(t) &= W_i^T A V_i \hat{x}_i(t) + W_i^T B u(t) \\ y(t) &= C V_i \hat{x}_i(t) + D u(t) \end{cases}$$

The obtained set of reduced local models  $\{\hat{\Sigma}_i : i = 1, \dots, l\}$  with state-space representations  $\{\hat{S}_i : i = 1, \dots, l\}$  have their states lying in different subspaces  $\{\mathcal{V}_i : i = 1, \dots, l\}$ .

### 3.2.2 Reduction via adapted balanced truncation

In [9], the authors propose a modified balanced truncation to tackle the problem of LPV model order reduction. For LPV models, the Petrov-Galerkin projection gives a reduced model of the form

$$\begin{cases} \dot{\hat{x}}(t) &= W_\rho^T \left( A V_\rho - \sum_{j=1}^d \frac{\partial V_\rho}{\partial \rho_j} \dot{\rho}_j \right) \hat{x}(t) + W_\rho^T B u(t) \\ y(t) &= C V_\rho \hat{x}(t) + D u(t) \end{cases}$$

The term  $\sum_{j=1}^d \frac{\partial V_\rho}{\partial \rho_j} \dot{\rho}_j$  can be avoided by projecting on a constant subspace  $\mathcal{V}$  spanned by a matrix  $V$ .

The idea is to project all the local models  $\Sigma_i$  onto a single subspace  $\mathcal{V}$  and orthogonally to different subspaces  $\mathcal{W}_i$ . Since the reduced states all lie in the same subspace and are expressed in the same basis, the interpolation can then be immediately applied.

For pMOR this method permits to skip the adaptation step, at the cost of a larger error in the Petrov-Galerkin approximation. Indeed, each projection is



done along the orthogonal complement of  $\mathcal{W}_i$ , same as balanced truncation. However, all projections are done onto a unique subspace  $\mathcal{V}$ , which induces a larger errors  $\epsilon_i$  than if it were onto the subspaces  $\mathcal{V}_i$  as in balanced projection. First, the projection achieving balanced truncation has to be rewritten, then the new formulation is used to project the local models onto a “mean” subspace  $\mathcal{V}$  orthogonally to different subspaces  $\mathcal{W}_i$ .

### Rewriting the projection of balanced truncation

Since the projection achieving balanced truncation is an oblique projection, it can be rewritten in the form of (2.4)

$$\Pi = V(V^T Q V)^{-1} V^T Q$$

It can be seen that choosing  $V$  as an orthonormal basis for  $\text{span}(L_r N_1)$  (using for example the QR decomposition of  $L_r N_1$ ) and  $Q = W_o$ , the projection

$$\Pi = V(V^T W_o V)^{-1} V^T W_o$$

achieves balanced truncation.

*Proof.* Using the QR decomposition of  $L_r N_1$

$$V R_V = L_r N_1$$

and seeing the projection as

$$\Pi = V ((W_o V)^T V)^{-1} (W_o V)^T = V (W^T V) W^T$$

$W = W_o V$  is developed

$$\begin{aligned} W_o V &= L_o L_o^T L_r N_1 R_V^{-1} \\ &= L_o U_1 \Lambda_1 N_1^T N_1 R_V^{-1} \\ &= L_o U_1 \Lambda_1 R^{-1}. \end{aligned}$$

Since  $\Lambda_1$  is only a scaling and multiplying  $L_o U_1 \Lambda_1$  by  $R^{-1}$  ( $R$  is an upper diagonal matrix) results in a linear combination of  $L_o U_1 \Lambda_1$ , then

$$\text{span}(W_o V) = \text{span}(L_o U_1 \Lambda_1 R^{-1}) = \text{span}(L_o U_1) = \text{span}(W_{bal})$$

and

$$\text{span}(V) = \text{span}(L_r N_1) = \text{span}(V_{bal})$$

so this projection achieves balanced truncation ( $V_{bal}$  and  $W_{bal}$  are the matrices  $V_i$  and  $W_i$  obtained in (3.1) by Laub’s algorithm for balanced truncation).  $\square$

### Projection onto a same subspace along different directions

To be close to balanced truncation, the (constant) subspace  $\mathcal{V}$  is chosen to best approximates the set of subspaces  $V_i$  that would have been obtained with the balanced truncation for the set of sampled models  $\{\Sigma_i : i = 1, \dots, l\}$ . This is done through a principal component analysis, that is, by means of a SVD of the collection of the matrices  $V_i$ , then  $V$  is chosen as the first  $k$  columns of  $U$  (these columns span the “most representative” subspace) representing the most important directions of the matrices  $V_i$  (details in section 3.3.1).

$$[V_1 \ V_2 \ \dots \ V_l] = U\Sigma N^T$$

The subspaces  $\mathcal{W}_i$  orthogonally to which the projection is done are exactly the subspaces obtained by balanced truncation for each  $\Sigma_i$ .

Then the state-space representations  $\{\hat{\Sigma}_i : i = 1, \dots, l\}$  of the obtained reduced models can be directly interpolated since their states lie in the same subspace  $\mathcal{V}$ .

## 3.3 Adaptation of the reduced local models

Each reduced local model  $\hat{\Sigma}_i$  has its state  $\hat{x}_i$  lying in a subspace  $\mathcal{V}_i$  (spanned by matrices  $V_i$ ) and expressed in the basis built by the columns of the matrix  $V_i$ . To interpolate the state-space representations  $\hat{\Sigma}_i$ , the states have to be expressed in a *generalized coordinate system* [7]. This is done by applying a state transformation to each  $\hat{\Sigma}_i$  that mirrors a projection onto a mean subspace  $\mathcal{R}$ .

### 3.3.1 Subspace $\mathcal{R}$

A “mean” subspace  $\mathcal{R}$  of dimension  $k$  is needed to approximate a set of subspaces  $\mathcal{V}_i$  of same dimension.

It is obtained by concatenating the matrices  $V_i$  spanning the subspaces  $\mathcal{V}_i$  into a matrix  $V_{all} = [V_1 \ V_2 \ \dots \ V_l]$ , then computing its SVD

$$V_{all} = U\Sigma N^T$$

The “mean” subspace is spanned by the first  $k$  columns of  $U$  (these columns span the “most representative” subspace), that represent the most important directions of the matrices  $V_i$ . This is a principal component analysis, the proof is derived in appendix C.

### 3.3.2 Generalized coordinates

The concept of generalized coordinates allows us to interpolate state-space representations having their states lying in different subspaces by giving them a common meaning. It has been introduced in [10], and further used in [7], [11]. Although in these articles the concept of generalized coordinates has been described by means of matrix transformations, here another slightly different approach is adopted.

To give the states a common meaning while they lie in different subspaces, we express them in new bases which are *compatible w.r.t.* the basis of another subspace  $\mathcal{R}$  of same dimension  $k$  (the “mean” subspace from section 3.3.1). The idea is to mirror the projection of  $\hat{x}_i$  onto  $\mathcal{R}$  (which would yield a vector  $\Pi_{\mathcal{R}}\hat{x}_i \in \mathbb{R}^k$ ) via a simple change of basis  $\hat{x}_i = T\hat{x}_i^* \in \mathbb{R}^k$  in  $\mathcal{V}_i$ .

It is important to distinguish a geometric vector (preexisting to any basis) from its algebraic expression in a basis. Similarly to LTI models that can be represented by an infinity of state-space representations via different bases, a geometric vector have an infinity of different algebraic expression.

The considered geometric vectors are the state-vectors of the reduced local models, noted  $\hat{x}_i$ . Three different algebraic expressions of these vectors represented in different bases are used:

- $[\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} \in \mathbb{R}^k$ ,  $\hat{x}_i$  expressed in  $\mathcal{V}_i$  in the basis span by the columns of  $V_i$ .
- $V_i[\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} \in \mathbb{R}^n$ ,  $\hat{x}_i$  expressed in the original state-space.
- $[\hat{x}_i]_{\mathcal{B}^*(\mathcal{V}_i)} \in \mathbb{R}^k$ ,  $\hat{x}_i$  expressed in  $\mathcal{V}_i$  in the new basis  $\mathcal{B}^*(\mathcal{V}_i)$ .

#### Definition 9: Generalized coordinate system

Given a set of subspaces  $\{\mathcal{V}_i : i = 1, \dots, l\}$  of dimension  $k$  with bases  $\mathcal{B}(\mathcal{V}_i) = [v_{i_1} \ v_{i_2} \ \dots \ v_{i_k}]$  and a subspace  $\mathcal{R}$  with basis  $\mathcal{B}(\mathcal{R}) = [r_1 \ r_2 \ \dots \ r_k]$  also of dimension  $k$ ,

The bases  $\mathcal{B}(\mathcal{V}_i)$  are compatible w.r.t. the basis  $\mathcal{B}(\mathcal{R})$ , that is, they describe a generalized coordinate system w.r.t. the basis  $\mathcal{B}(\mathcal{R})$ , if

$$\forall i, j, \quad \Pi_{\mathcal{R}}v_{i_j} = r_j$$

In other words, the bases  $\mathcal{B}(\mathcal{V}_i)$  define a set of generalized coordinates w.r.t. the basis  $\mathcal{B}(\mathcal{R})$  if the projection of the basis vectors  $v_{i_j}$  onto  $\mathcal{R}$  is equal to the basis vectors  $r_j$ .

Equivalently, considering any vector instead of the basis vector,

$$\forall x_i \in \mathcal{V}_i, \quad [x_i]_{\mathcal{B}(\mathcal{V}_i)} = [\Pi_{\mathcal{R}}x_i]_{\mathcal{B}(\mathcal{R})}$$

$[x_i]_{\mathcal{B}(\mathcal{V}_i)}$  is the expression of the geometric vector  $x_i$  in the basis  $\mathcal{B}(\mathcal{V}_i)$ ,  
 $[\Pi_{\mathcal{R}}x_i]_{\mathcal{B}(\mathcal{R})}$  is the expression of the projection of  $x_i$  onto  $\mathcal{R}$  in the basis  $\mathcal{B}(\mathcal{R})$ .

### Generalized coordinate system by orthogonal projection

As it is used in [10], new bases  $\mathcal{B}_{(\mathcal{V})}^*$  are constructed to obtain a generalized coordinate system defined via an orthogonal projection  $\Pi_{\mathcal{R}}$ . Since the subspaces  $\mathcal{V}_i$  are spanned by matrices  $V_i$  and the subspace  $\mathcal{R}$  is spanned by  $R$ , the vector  $\hat{x}_i$  projected onto  $\mathcal{R}$  and expressed in  $\mathcal{B}(\mathcal{R})$ , the basis described by the columns of  $R$ , is

$$[\Pi_{\mathcal{R}}\hat{x}_i]_{\mathcal{B}(\mathcal{R})} = R^T V_i [\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} \quad (3.2)$$

Note that  $V_i[x_i]_{\mathcal{B}(\mathcal{V}_i)}$  is the vector  $x_i$  expressed in the original state-space  $\mathbb{R}^n$  and then by multiplying by  $R^T$  we get its projection onto  $\mathcal{R}$  expressed in  $\mathcal{B}(\mathcal{R})$ .

The expression of the  $\hat{x}_i$  in the new bases  $\mathcal{B}_{(\mathcal{V}_i)}^*$  are obtained by suitable change of bases

$$[\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} = T_i [\hat{x}_i]_{\mathcal{B}_{(\mathcal{V}_i)}^*} \iff [\hat{x}_i]_{\mathcal{B}_{(\mathcal{V}_i)}^*} = T_i^{-1} [\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} \quad (3.3)$$

The new bases must define a set of generalized coordinates, hence must satisfy

$$[\hat{x}_i]_{\mathcal{B}_{(\mathcal{V}_i)}^*} = [\Pi_{\mathcal{R}}\hat{x}_i]_{\mathcal{B}(\mathcal{R})}$$

introducing equations (3.2) and (3.3) into this last equation, the appropriate change of bases  $T_i$  are obtained

$$T_i = (R^T V_i)^{-1}$$

After the state-space transformations  $T_i$ , the state-space representations of the reduced local models

$$\left\{ \hat{\mathcal{S}}_i^* = \left[ \begin{array}{c|c} \hat{A}_i^* & \hat{B}_i^* \\ \hat{C}_i^* & \hat{D}_i^* \end{array} \right] = \left[ \begin{array}{c|c} T_i^{-1} \hat{A}_i T_i & T_i^{-1} \hat{B}_i \\ \hat{C}_i T_i & \hat{D}_i \end{array} \right] : i = 1, \dots, l \right\}$$

are now expressed in generalized coordinates w.r.t  $R$ , hence the interpolation between these new state-space representation of the reduced local models is possible.

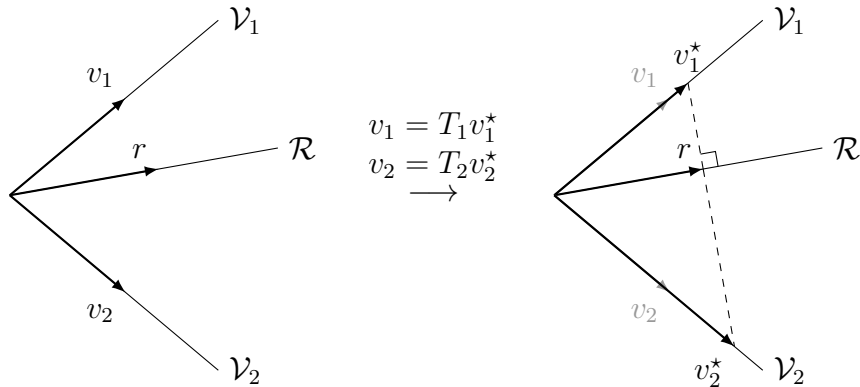


Figure 3.1:  $v_1^*$  and  $v_2^*$  generalized coordinates w.r.t.  $r$

## 3.4 Interpolation

To interpolate the adapted reduced local state-space representations  $\hat{S}_i^*$  choices have to be made on the manifold in which to interpolate and the method for computing the interpolation coefficients.

### 3.4.1 Vector space where to interpolate

The interpolation of the matrices of  $\hat{S}_i^*$  can be done in different vector spaces, in the vector space of matrices (of which they are an element) or in the tangent space to an element of the matrix manifold. Note that the set of Hurwitz matrices is not a convex set, hence interpolating matrices corresponding to stable systems does not necessarily results in a stable interpolated model.

#### Matrix space

The matrices  $\hat{A}_i^*$ ,  $\hat{B}_i^*$ ,  $\hat{C}_i^*$  can be interpolated in their respective vector spaces, specifically the spaces  $\mathbb{R}^{k \times k}$  for the matrices  $\hat{A}_i^*$ ,  $\mathbb{R}^{k \times m}$  for matrices  $\hat{B}_i^*$  and  $\mathbb{R}^{p \times k}$  for matrices  $\hat{C}_i^*$ .

#### Tangent space

Introduced in [12],[13], [14] and also [15], This technique uses the tangent space to the matrix manifold at a particular point. For interpolating the matrices  $\hat{A}_i^*$ , one of the matrices to be interpolated is selected as a reference in the matrix manifold  $\mathbb{R}^{k \times k}$ . Its tangent space is constructed and the other matrices to be interpolated are mapped into this tangent space via the logarithmic map. The matrices are then interpolated in the tangent space and

the obtained interpolated matrix is mapped back to the matrix manifold via the exponential map.

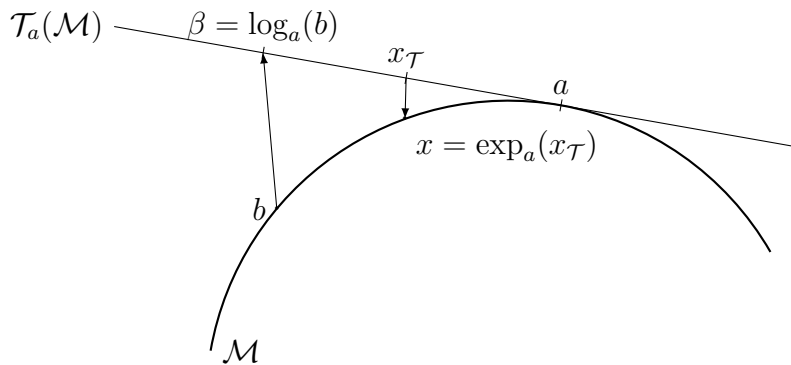


Figure 3.2: interpolation in the tangent space of  $\mathcal{M}$  at  $a$

As shown in the illustration of Figure 3.2, a manifold  $\mathcal{M}$  is considered. To interpolate two of its elements  $a$  and  $b$  in the tangent space of  $a$ ,  $b$  is mapped to this tangent space  $\mathcal{T}_a(\mathcal{M})$ , obtaining  $\beta = \log_a(b)$ . Then  $a$  and  $\beta$  are interpolated using any interpolation method (e.g. linear, splines) and the result  $x_{\mathcal{T}}$  is mapped back to  $\mathcal{M}$ .

For details on the interpretation of such an interpolation, see [16]. For matrices  $A$  and  $B$  in the manifold of nonsingular matrices,  $B$  is mapped on the tangent space to the manifold at  $A$  by

$$\beta = \log(BA^{-1})$$

and the exponential map from the tangent space to the manifold is

$$B = \exp(\beta)A$$

using the matrix exponential and logarithmic exponential.

For other manifolds (e.g. positive-semidefinite matrices), the exponential and logarithmic mapping are different.

### 3.4.2 Interpolation method

There exists many interpolation methods such as piecewise constant, linear, polynomial, spline, etc. It is not clear whether a method is better than another one and as [7] pointed out, it may depend on the considered model. In this work, two methods are applied, namely linear interpolation and cubic spline interpolation. The interpolation occurs element by element between the matrices.

## Conclusion

Different methods were considered in this chapter. For the reduction-adaptation process, two methods were studied: the reduction by oblique projection onto a mean subspace, and the projection achieving balanced truncation followed by an adaption step to ensure the generalized coordinates property w.r.t. a mean subspace. This was the main concern, but also a quick overview on how to achieve each step was given. In fact there are many possibilities on what technique to use for the sampling and also for the interpolation parts.

# Chapter 4

## Derived pMOR Method and Results

### Introduction

This last chapter introduces a method, combination of the oblique projection [9] and the generalized coordinate system [10],[7]. Then the different methods are compared by applying them on an academic model.

### 4.1 Derived method

The proposed technique combines the ideas of the projection orthogonally to a varying subspace from [9] and of the generalized coordinates [10]. The idea is to reduce the local models by balanced truncation, then to construct a set of generalized coordinates w.r.t. a “mean” subspace  $\mathcal{R}$ , except it is not an orthogonal projection onto  $\mathcal{R}$  that is used to construct the new bases but a projection onto  $\mathcal{R}$  along the same direction as the balanced truncation.

Figures 4.1, 4.2 and 4.3 illustrates the differences between all the considered methods. The illustrations depicts a reduction from  $n = 2$  to  $k = 1$  of the  $i$ -th local model having state  $x_i$ . The subspaces  $\mathcal{V}_i$  and  $\mathcal{W}_i$  are the one obtained by balanced truncation, and the subspace  $\mathcal{R}$  is computed via an SVD to approximate at best the subspaces  $\mathcal{V}_i$ .

**Oblique projection** In this method, the states are projected onto the subspace  $\mathcal{R}$  orthogonally to the subspaces  $\mathcal{W}_i$ , hence the reduction of the local models is not exactly a balanced truncation (due to the projection being onto  $\mathcal{R}$  and not onto  $\mathcal{V}_i$ ). The advantage is that the interpolation can be directly applied and this method is also valid for LPV models.



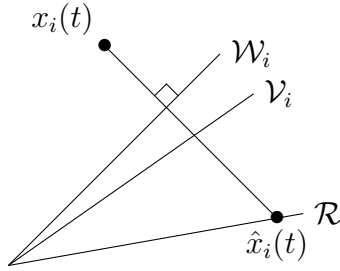


Figure 4.1: Oblique projection method

**Generalized coordinates via orthogonal projection** This method first achieves an exact balanced truncation of the reduced models and then builds a set of generalized coordinates w.r.t.  $\mathcal{R}$  so the interpolation is meaningful. Here the set of generalized coordinates is defined via an orthogonal balanced projection.

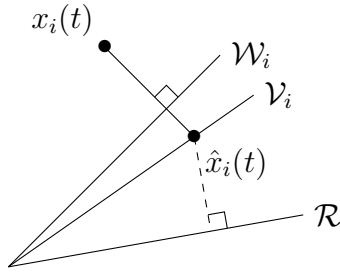


Figure 4.2: Generalized coordinates via orthogonal projection

**Generalized coordinates via oblique projection** Here, an exact balanced truncation of the local models is achieved, then a set of generalized coordinates w.r.t.  $\mathcal{R}$  is constructed by means of an oblique projection. The projection is done orthogonally to the subspaces  $\mathcal{W}_i$  so the obtained models are more in accordance to the balanced truncation.

The subspaces  $\mathcal{R}$ ,  $\mathcal{V}_i$ ,  $\mathcal{W}_i$  are spanned respectively by the matrices  $R$ ,  $V_i$ ,  $W_i$ . The expression of the projection is

$$[\Pi_{\mathcal{R}\mathcal{W}}\hat{x}_i]_{\mathcal{B}(\mathcal{R})} = (W^T R)^{-1} W^T V_i [\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)}$$

The change of bases is

$$[\hat{x}_i]_{\mathcal{B}(\mathcal{V}_i)} = T_i [\hat{x}_i]_{\mathcal{B}^*(\mathcal{V}_i)}$$

So the generalized coordinates definition

$$[\hat{x}_i]_{\mathcal{B}^*(\mathcal{V}_i)} = [\Pi_{\mathcal{R}}\hat{x}_i]_{\mathcal{B}(\mathcal{R})}$$

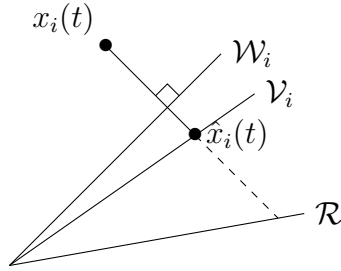


Figure 4.3: Generalized coordinates via oblique projection

gives the state transformations

$$T_i = (W_i^T V_i)^{-1} W_i^T R$$

After the state-space transformations  $T_i$ , the state-space representations of the reduced local models

$$\left\{ \hat{S}_i^* = \left[ \begin{array}{c|c} \hat{A}_i^* & \hat{B}_i^* \\ \hline \hat{C}_i^* & \hat{D}_i^* \end{array} \right] = \left[ \begin{array}{c|c} T_i^{-1} \hat{A}_i T_i & T_i^{-1} \hat{B}_i \\ \hline \hat{C}_i T_i & \hat{D}_i \end{array} \right] : i = 1, \dots, l \right\}$$

are now expressed in generalized coordinates w.r.t  $R$ , hence the interpolation between these new state-space representations of the reduced local models is possible.

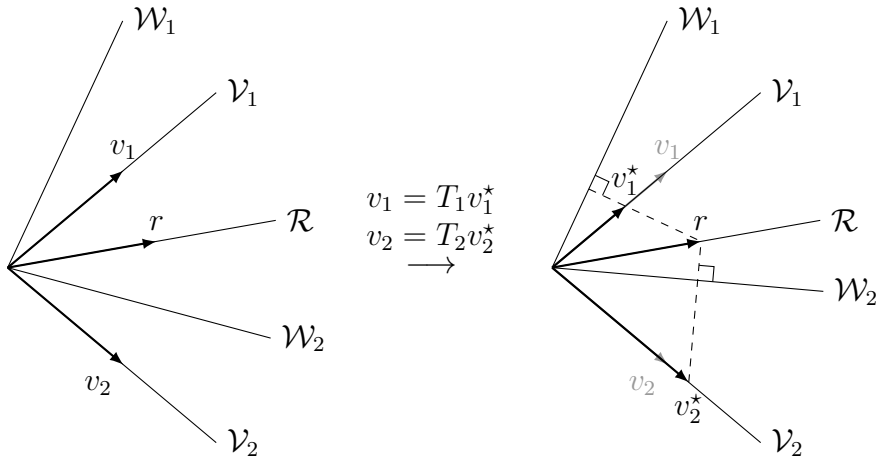


Figure 4.4:  $v_1^*$  and  $v_2^*$  generalized coordinates w.r.t.  $r$

## 4.2 Results on academic examples

The investigated methods were coded in Matlab and applied to two academic mass-spring-damper systems. Each of them emphasizes different aspects of the methods.

### 1D example

This first example has been used to assess pMOR methods in [11] and then in [14]. The model has one uncertain parameter  $\mu \in [0, 1]$  which impacts multiple elements of the system described in figure 4.5.

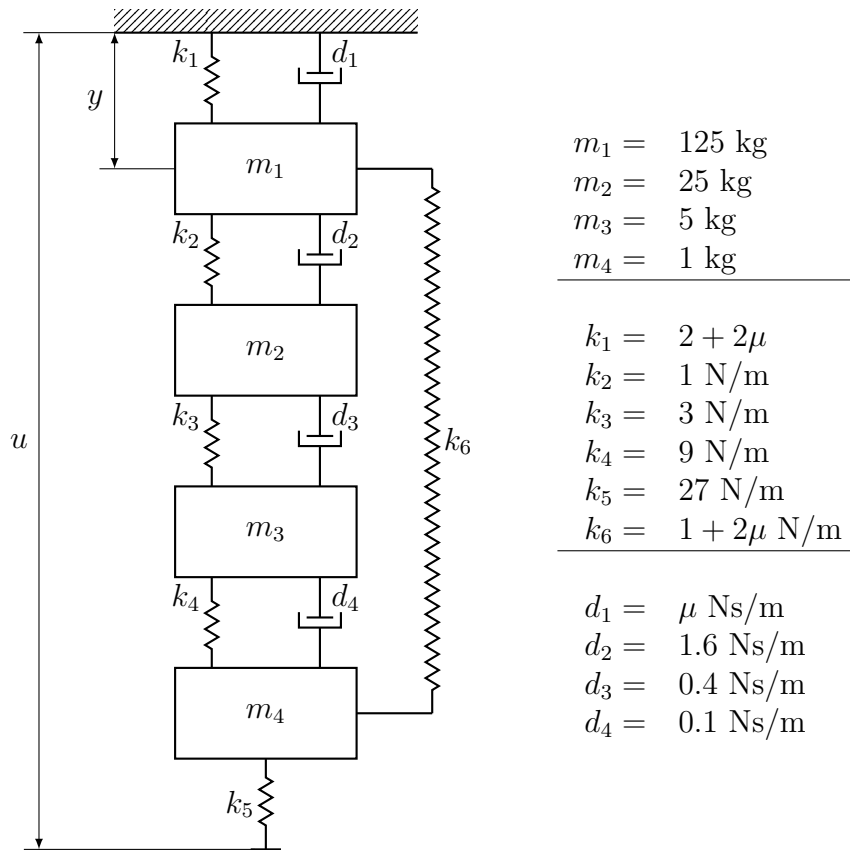


Figure 4.5: Mass-spring-damper system (example 1)

In the parameter space  $[0, 1]$ , three models are sampled at  $\mu = \{0, 0.5, 1\}$ . Three different methods are used to reduce and adapt the models, namely the oblique projection (method 1, [9]), and the balanced truncation in conjunction with the generalized coordinates via orthogonal projection (method 2, [10]) and via oblique projection (mixed method). The interpolation is done

directly amongst the matrices coefficients via a cubic spline interpolation using Matlab.

Figure 4.6 describes the reduced models at parameter value  $\mu = 0.25$ , and also shows the three local models from which the interpolation is done. Since the reduced models obtained by interpolation are very close to the reference (the model reduced by balanced truncation at parameter value  $\mu = 0.25$ ), Figure 4.7 shows the error between the considered models and the original one for a better comparison.

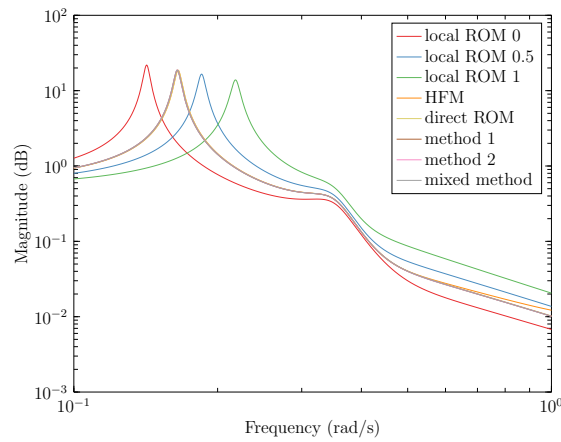


Figure 4.6: Bode plot (example 1)

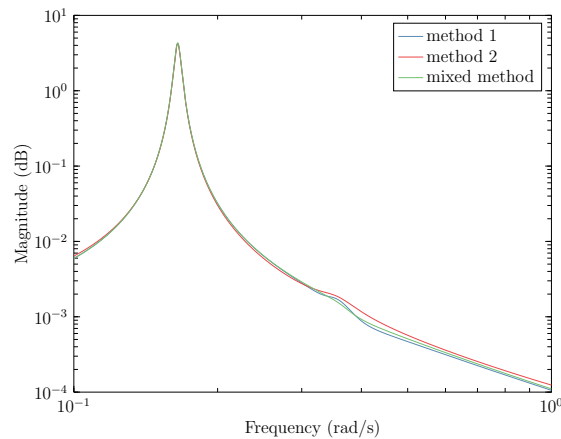


Figure 4.7: Bode plot of the error between the pMOR methods and the direct ROM (example 1)

To assess the validity of the pMOR methods over the parameter space, it is possible to compute the  $H_\infty$  (Figure 4.8) or the  $H_2$  norm of the error between

the reduced order models and the original one. In this case the objective is to be as close as possible to the error obtained by direct balanced truncation at each parameter value.

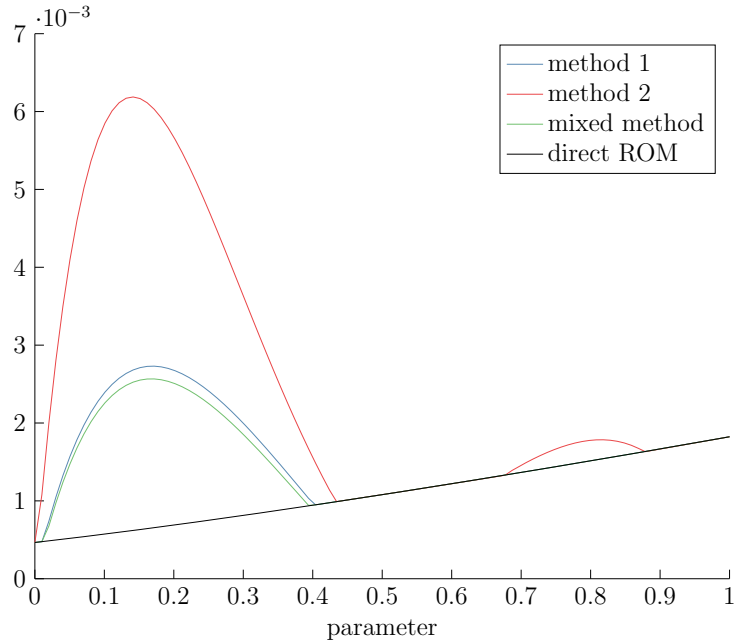


Figure 4.8: Relative  $H_\infty$  error between the reduced models and the original one (example 1)

## 2D example

Another example is needed to understand how valuable the generalized coordinates property is. This example is made of a number of masses, linked one to another by springs and dampers, and attached at both extremities by springs and dampers (figure 4.9). The states are the position and velocity of each mass. The input is the force exerted on the first mass, and the output is the position of the first mass. The system has 40 masses so the model has 80 states. The masses, springs and dampers have all the same values  $m = 10$  kg,  $d = 3$  Ns/m,  $k = 5$  N/m. The first spring and damper have uncertain value, ranging in their nominal value  $\pm 20\%$ .

An interesting advantage of methods 2 and the mixed method is that the obtained models at the sampling points are exactly the one obtained by balanced truncation. In this example the local parameters are sampled as the 4 edges of the parameter space describe by the variation of the first spring and first damper. Due to the generalized coordinate system, in figure

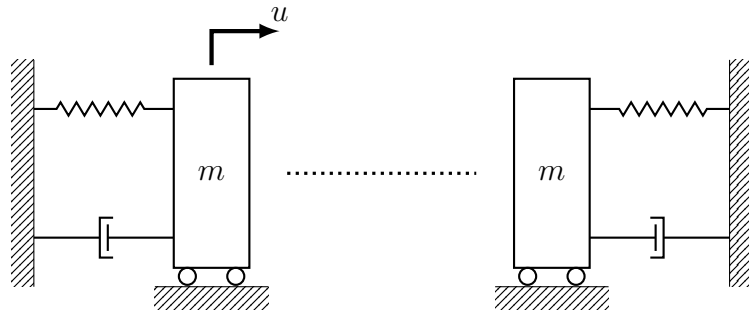


Figure 4.9: Mass-spring-damper system (example 2)

4.10 the reduced order models obtained by method 2 and the mixed method match exactly the objective function (obtained by balanced truncation at each parameter point) at the sampling points (the 4 edges).

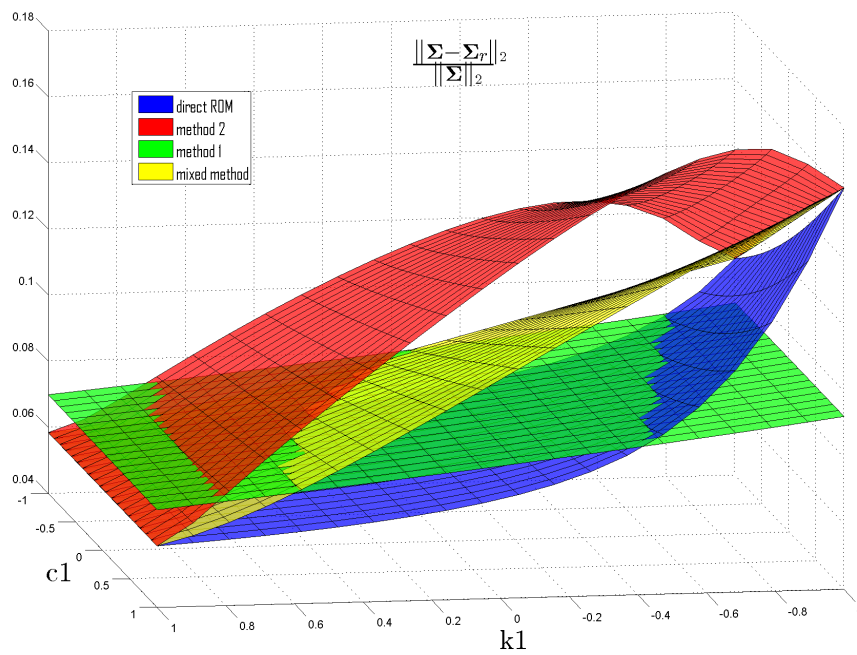


Figure 4.10: Relative error of the  $H_2$  norm (example 2)

## Conclusion

This chapter introduced a method, combination of the two method studied in the last chapter. this method has the advantage of achieving exact balanced truncation at the sampled parameter values (due to the generalized coordinates, as in [10]) and is closer to balanced truncation in the sense that the projection by which is define the generalized coordinates is the oblique projection onto a mean subspace (as in [9]).

# Conclusion

Model order reduction is an active field of research, it has emerged to simplify the numerical models used to simulate dynamical systems. The presented work consisted in studying a few methods in the particular context of parametric model order reduction using the well-known balanced truncation. The methods can be subject to many variations since each of the steps sampling - local reduction - adaptation - interpolation can be achieved in different ways. The sampling part is key feature for pMOR but was left aside in this study (a simple grid was used here), similarly the choice for the local reduction was to use balanced truncation. The focus was on the adaptation part, with a new method proposed, mix between two existing methods. In this context, an effort has been made to differentiate the different geometric objects involved and their algebraic expression. This was important since the generalized coordinate system principle is to equalize the algebraic expression of two different geometric vectors. Also the interpolation part is a key issue, and stability problems related to the interpolation have not been adressed here.



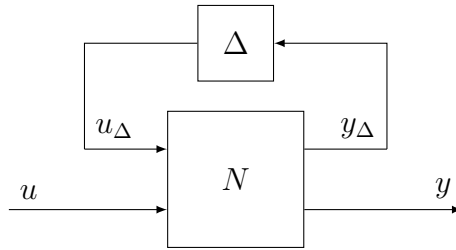
# Appendices

# Appendix A

## LFT representation

Linear Fractional Transformation (LFT) is a possible representation of a parametric model. It is often used when describing a system with uncertainties.

The general framework of the LFT representation is as follow:



$$N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$$

$$F(N, \Delta) = N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12}$$

is the linear fractional transformation of  $N$  and  $\Delta$ .

For a description of a model with parameter uncertainties,  $\Delta$  is a diagonal matrix describing the uncertainties and  $N$  is an augmented state-space representation: it consists of the state-space representation of the nominal model and a mapping of the uncertainties.

The uncertainties are described by

$$u_{\Delta} = \Delta y_{\Delta}$$

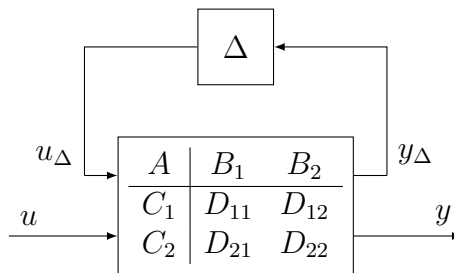
$$\begin{bmatrix} u_{\delta_1} \\ \vdots \\ u_{\delta_d} \end{bmatrix} = \begin{bmatrix} \delta_1 & & \\ & \ddots & \\ & & \delta_d \end{bmatrix} \begin{bmatrix} y_{\delta_1} \\ \vdots \\ y_{\delta_d} \end{bmatrix}$$

with  $d$  the number of uncertain parameters.

$$\begin{cases} \dot{x}(t) &= Ax(t) + B_1 u_\Delta(t) + B_2 u(t) \\ y_\Delta(t) &= C_1 x(t) + D_{11} u_\Delta(t) + D_{12} u(t) \\ y(t) &= C_2 x(t) + D_{21} u_\Delta(t) + D_{22} u(t) \end{cases}$$

$A, B_2, C_2, D_{22}$  are the system matrices corresponding to the nominal model.  $B_1, C_1, D_{11}, D_{12}, D_{21}$  are matrices representing how the uncertainties enter the model.

Hence the LFT representation of a LTI system with parametric uncertainties:



# Appendix B

## Other MOR techniques

### B.1 Modal truncation

#### B.1.1 Concept

The modal truncation method consists in retaining only the most dominant eigenvalues of  $A$ , i.e. the eigenvalues with the largest real part.

This method preserves the physical significance of the retained states, for instance reducing a finite element model of a mechanical structure, the retained state represents the dominant modes of the structure.

#### B.1.2 Method

Applying a state transformation to diagonalize  $A$  and ordering its eigenvalues, we obtain:

$$\tilde{A} = T^{-1}AT = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}, \text{Re}(\lambda_i) \geq \text{Re}(\lambda_{i+1}) \text{ and } \text{Re}(\lambda_n) > 0$$
$$\tilde{B} = \begin{bmatrix} b_1^T \\ \vdots \\ b_n^T \end{bmatrix} \quad \tilde{C} = [ c_1 \quad \dots \quad c_n ]$$

Then the system is truncated at the desired order.

Applying a state-space transformation and then truncating can be seen as a projection with matrices  $V$  and  $W$ , where  $V \in \mathbb{R}^{n \times k}$  is the first  $k$  columns of  $T$  and  $W^T \in \mathbb{R}^{k \times n}$  is the first  $k$  rows of  $T^{-1}$ .

### B.1.3 Bound on $\|\Sigma - \Sigma_r\|_{H_\infty}$

It is always interesting to have bounds on the error made by the approximation, to know how good the approximation is. For modal truncation, there exists a bound on the  $H_\infty$  norm of the difference between the original model and the reduced one,

$$\|\Sigma - \Sigma_r\|_{H_\infty} \leq \sum_{i=k+1}^n \frac{\|c_i b_i^T\|_2}{|\Re(\lambda_i)|}.$$

## B.2 Balanced residualization

In balanced residualization, instead of discarding the last  $n - k$  states, their derivatives are set to zero. Considering the following model, with  $x_1 \in \mathbb{R}^k$  and  $x_2 \in \mathbb{R}^{n-k}$  the vector whose derivative is set to zero.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du \end{aligned}$$

Setting  $\dot{x}_2 = 0$ , we obtain

$$\begin{aligned} 0 &= A_{21}x_1 + A_{22}x_2 + B_2u \\ \iff x_2 &= -A_{22}^{-1}A_{21}x_1 - A_{22}^{-1}B_2u \end{aligned}$$

then replacing  $x_2$  in the equation

$$\begin{aligned} \dot{x}_1 &= A_{11}x_1 + A_{12}x_2 + B_1u \\ y &= C_1x_1 + C_2x_2 + Du \end{aligned}$$

we obtain

$$\begin{aligned} \dot{x}_1 &= A_{11}x_1 - A_{12}A_{22}^{-1}A_{21}x_1 - A_{12}A_{22}^{-1}B_2u + B_1u \\ y &= C_1x_1 - C_2A_{22}^{-1}A_{21}x_1 - C_2A_{22}^{-1}B_2u + Du \end{aligned}$$

Hence the reduced matrices are:

$$\begin{aligned} \hat{A} &= A_{11} - A_{12}A_{22}^{-1}A_{21} \\ \hat{B} &= B_1 - A_{12}A_{22}^{-1}B_2 \\ \hat{C} &= C_1 - C_2A_{22}^{-1}A_{21} \\ \hat{D} &= D - C_2A_{22}^{-1}B_2 \end{aligned}$$

This method approximates very well the original model at low frequencies, but it requires more computation since the part of the model that is neglected in balanced truncation is here reintroduced in the equations, but the matrices  $A_{12}, A_{21}, A_{22}, B_2, C_2$  can be very large, so the method can be computationally expensive.

# Appendix C

## Proof for the mean subspace $\mathcal{R}$

The goal is to compute an orthonormal basis  $R = [r_1 \ r_2 \ \dots \ r_k]$  for the subspace  $\mathcal{R}$  which should approximate at best the set of subspaces  $\{\mathcal{V}_i, i = 1, \dots, l\}$  spanned by the matrices  $V_i = [v_{i_1} \ v_{i_2} \ \dots \ v_{i_k}]$ . This is done by a principal component analysis (PCA),  $R$  is computed as the first  $k$  principal axes of  $V_{all}$ .

The subspace  $\mathcal{R}$  best approximates the set  $\{\mathcal{V}_i, i = 1, \dots, l\}$  in the sense that it maximizes the variance of the 2-norm of the projections onto  $\mathcal{R}$  of the basis vectors  $v_{i_j}$  of the subspaces  $\mathcal{V}_i$ .

The proof is derived for the first basis vector  $r_1$ , then it is extended to  $r_2, r_3, \dots, r_k$  by iteration.

The maximization of the variance of the 2-norm of the projection onto  $r_1$  of the basis vectors  $v_{i_j}$  is formulated as

$$\begin{aligned} \max_{r_1 \in \mathbb{R}^n} \quad & \sum_{i=1}^l \sum_{j=1}^k \|v_{i_j}^T r_1\|_2^2 \\ \text{s.t.} \quad & \|r_1\|_2 = 1 \end{aligned} \tag{C.1}$$

The objective function can be rewritten as

$$\sum_{i=1}^l \sum_{j=1}^k \|v_{i_j}^T r_1\|_2^2 = \sum_{i=1}^l \sum_{j=1}^k r_1^T v_{i_j} v_{i_j}^T r_1 = \sum_{i=1}^l r_1^T V_i V_i^T r_1 = r_1^T V_{all} V_{all}^T r_1$$

The largest eigenvalue  $\lambda_{max}$  of a matrix  $A$  is solution of the optimization problem

$$\begin{aligned} \lambda_{max} = \max_{x \in \mathbb{R}^n} \quad & x^T A x \\ \text{s.t.} \quad & \|x\|_2 = 1 \end{aligned}$$

Hence

$$\begin{aligned} \max_{r_1 \in \mathbb{R}^n} \quad & r_1^T V_{all} V_{all}^T r_1 \\ \text{s.t.} \quad & \|r_1\|_2 = 1 \end{aligned}$$

is solved by the largest eigenvalue of  $V_{all}V_{all}^T$ .

Expressing  $V_{all}$  as its SVD

$$V_{all} = U\Sigma N^T$$

Then the eigendecomposition of the product  $V_{all}V_{all}^T$  is

$$V_{all}V_{all}^T = U\Sigma^2U^T$$

The solution of the optimization problem is  $r_1 = u_1$ , the first column of  $U$ , which is the eigenvector corresponding to the largest eigenvalue of  $V_{all}V_{all}^T$ . The other basis vectors  $r_2, r_3, \dots, r_k$  are the columns  $u_2, u_3, \dots, u_k$  of the matrix  $U$ .

The proof can be derived for every  $r_j$ ,  $j = 1, \dots, k$  by removing the first  $r_{j-1}$  basis vectors and projecting onto the subspace spanned by the basis  $(r_j, \dots, r_k)$ .



# Bibliography

- [1] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. 2005.
- [2] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. 1996.
- [3] W. Gu, Hr. Petkov, and M. Konstantinov. *Robust Control Design with MATLAB*. 2005.
- [4] Bruce C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26:17–32, 1981.
- [5] Alan J. Laub, Michael T. Heath, Chris C. Paige, and Robert C. Ward. Computation of system balancing transformations - and other applications of simultaneous diagonalization algorithms. *IEEE Transactions on Automatic Control*, AC-32:115–122, 1987.
- [6] S.J. Hammarling. Numerical solution of the stable, non-negative definite lyapunov equation. *IMA Journal of Numerical Analysis*, 2:303–323, 1985.
- [7] Matthias Geuss, Heiko Panzer, and Boris Lohmann. On parametric model order reduction by matrix interpolation. In *European Control Conference*, 2013.
- [8] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM*, 57:483–531, 2015.
- [9] Julian Theis, Peter Seiler, and Herbert Werner. Lpv model order reduction by parameter-varying oblique projection. *IEEE Transactions on Control Systems Technology*, 2017.

- [10] Heiko Panzer, Jan Mohring, Rudy Eid, and Boris Lohmann. Parametric model order reduction by matrix interpolation. *Automatisierungstechnik*, 2010.
- [11] Boris Lohmann and Rudy Eid. Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models. *Methoden und Anwendungen der Regelungstechnik*, 2009.
- [12] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46, 2008.
- [13] David Amsallem, Juien Cortial, Kevin Carlberg, and Charbel Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Meth. Engng*, 2009.
- [14] David Amsallem and Charbel Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM J. SCI. COMPUT.*, 33, 2011.
- [15] Joris Degroote, Jan Vierendeels, and Karen Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int. J. Numer. Meth. Fluids*, 2010.
- [16] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 2007.