MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Recent Advances in Parametric Model Order Reduction (of Descriptor Systems)

Peter Benner   Lihong Feng   Yao Yue

EU-MORNET Workshop
MOR 4 MECHATRONICS 2018

Wilhemshaven, February 2, 2018

Outline

Outline

Introduction

Parametric Dynamical Systems

$$\Sigma(p): \begin{cases} E(p)\dot{x}(t;p) &= f(t, x(t;p), u(t), p), \quad x(t_0) = x_0, & \text{(a)} \\ y(t;p) &= g(t, x(t;p), u(t), p) & \text{(b)} \end{cases}$$

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n\times n}$),
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantity of interest) $y(t;p) \in \mathbb{R}^q$,
  (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Introduction

Parametric Dynamical Systems

$$\Sigma(p): \left\{ \begin{array}{rcll} E(p)\dot{x}(t;p) & = & f(t,x(t;p),u(t),p), \quad x(t_0)=x_0, & \text{(a)} \\ y(t;p) & = & g(t,x(t;p),u(t),p) & \text{(b)} \end{array} \right.$$

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n \times n}$),
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantity of interest) $y(t;p) \in \mathbb{R}^q$,
  (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

$E(p)$ singular $\Rightarrow$ (a) is system of differential-algebraic equations $\rightsquigarrow$ descriptor system,
otherwise $\Rightarrow$ (a) is system of ordinary differential equations.

Introduction

Parametric Dynamical Systems

$$\Sigma(p) : \begin{cases} E(p)\dot{x}(t;p) &= f(t, x(t;p), u(t), p), \quad x(t_0) = x_0, & \text{(a)} \\ y(t;p) &= g(t, x(t;p), u(t), p) & \text{(b)} \end{cases}$$

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n \times n}$),
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantity of interest) $y(t;p) \in \mathbb{R}^q$,
  (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Applications:

- Repeated simulation for varying material or geometry parameters, boundary conditions,
- control, optimization and design,
- of models, often generated by FE software (e.g., ANSYS, NASTRAN,...) or automatic tools (e.g., Modelica).

Parametric Dynamical Systems

$$\Sigma(p): \begin{cases} E(p)\dot{x}(t;p) = f(t, x(t;p), u(t), p), \quad x(t_0) = x_0, & \text{(a)} \\ y(t;p) = g(t, x(t;p), u(t), p) & \text{(b)} \end{cases}$$

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ $(E \in \mathbb{R}^{n\times n})$,
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantity of interest) $y(t;p) \in \mathbb{R}^q$, (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Underlying PDE and boundary conditions often not accessible!

Introduction

Parametric Dynamical Systems

$$\Sigma(p) : \begin{cases} E(p)\dot{x}(t;p) &= f(t, x(t;p), u(t), p), \quad x(t_0) = x_0, & \text{(a)} \\ y(t;p) &= g(t, x(t;p), u(t), p) & \text{(b)} \end{cases}$$

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n \times n}$),
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantity of interest) $y(t;p) \in \mathbb{R}^q$,
  (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Underlying PDE and boundary conditions often not accessible!

Parametric discretized model often not available,
but matrices for certain parameter values can be extracted
(or output data for given u and p can be generated!)

Linear, Time-Invariant (Parametric) Systems

$$\begin{array}{rcll} E(p)\dot{x}(t;p) & = & A(p)x(t;p) + B(p)u(t), & A(p),\ E(p) \in \mathbb{R}^{n\times n}, \\ y(t;p) & = & C(p)x(t;p), & B(p) \in \mathbb{R}^{n\times m},\ C(p) \in \mathbb{R}^{q\times n}. \end{array}$$

Introduction

Linear, Time-Invariant (Parametric) Systems

$$\begin{aligned} E(p)\dot{x}(t;p) &= A(p)x(t;p) + B(p)u(t), \quad & A(p),\ E(p) \in \mathbb{R}^{n\times n},\\ y(t;p) &= C(p)x(t;p), & B(p) \in \mathbb{R}^{n\times m},\ C(p) \in \mathbb{R}^{q\times n}. \end{aligned}$$

Laplace Transformation / Frequency Domain

Application of Laplace transformation

$$x(t;p) \mapsto x(s;p), \quad \dot{x}(t;p) \mapsto sx(s;p)$$

to linear system with $x(0;p) \equiv 0$:

$$sE(p)x(s;p) = A(p)x(s;p) + B(p)u(s), \quad y(s;p) = C(p)x(s;p),$$

yields I/O-relation in frequency domain:

$$y(s;p) = \Big(\underbrace{C(p)(sE(p) - A(p))^{-1}B(p)}_{=:G(s,p)} \Big) u(s).$$

$G(s,p)$ is the parameter-dependent transfer function of $\Sigma(p)$.

Linear, Time-Invariant (Parametric) Systems

$$
\begin{aligned}
E(p)\dot{x}(t;p) &= A(p)x(t;p) + B(p)u(t), \quad & A(p),\ E(p) \in \mathbb{R}^{n\times n},\\
y(t;p) &= C(p)x(t;p), & B(p) \in \mathbb{R}^{n\times m},\ C(p) \in \mathbb{R}^{q\times n}.
\end{aligned}
$$

Laplace Transformation / Frequency Domain

Application of Laplace transformation

$$x(t;p) \mapsto x(s;p), \quad \dot{x}(t;p) \mapsto sx(s;p)$$

to linear system with $x(0;p) \equiv 0$:

$$sE(p)x(s;p) = A(p)x(s;p) + B(p)u(s), \quad y(s;p) = C(p)x(s;p),$$

yields I/O-relation in frequency domain:

$$y(s;p) = \Big(\underbrace{C(p)(sE(p) - A(p))^{-1}B(p)}_{=:G(s,p)}\Big)u(s).$$

$G(s,p)$ is the parameter-dependent transfer function of $\Sigma(p)$.

Goal: Fast evaluation of mapping $(u,p) \rightarrow y(s;p)$.

The PMOR Problem

The Parametric Model Order Reduction (PMOR) Problem

Approximate the dynamical system

$$\begin{array}{rcll} E(p)\dot{x} & = & A(p)x + B(p)u, & E(p),\ A(p) \in \mathbb{R}^{n\times n}, \\ y & = & C(p)x, & B(p) \in \mathbb{R}^{n\times m},\ C(p) \in \mathbb{R}^{q\times n}, \end{array}$$

by reduced-order system

$$\begin{array}{rcll} \hat{E}(p)\dot{\hat{x}} & = & \hat{A}(p)\hat{x} + \hat{B}(p)u, & \hat{E}(p),\ \hat{A}(p) \in \mathbb{R}^{r\times r}, \\ \hat{y} & = & \hat{C}(p)\hat{x}, & \hat{B}(p) \in \mathbb{R}^{r\times m},\ \hat{C}(p) \in \mathbb{R}^{q\times r}, \end{array}$$

of order $r \ll n$, such that w.r.t. some appropriate multivariate function space

$$\|y - \hat{y}\| = \left\| Gu - \hat{G}u \right\| \leq \left\| G - \hat{G} \right\| \cdot \|u\| < \text{tolerance} \cdot \|u\|$$

The PMOR Problem

The Parametric Model Order Reduction (PMOR) Problem

Approximate the dynamical system

$$
\begin{array}{rcll}
E(p)\dot{x} & = & A(p)x + B(p)u, & E(p),\ A(p) \in \mathbb{R}^{n\times n}, \\
y & = & C(p)x, & B(p) \in \mathbb{R}^{n\times m},\ C(p) \in \mathbb{R}^{q\times n},
\end{array}
$$

by reduced-order system

$$
\begin{array}{rcll}
\hat{E}(p)\dot{\hat{x}} & = & \hat{A}(p)\hat{x} + \hat{B}(p)u, & \hat{E}(p),\ \hat{A}(p) \in \mathbb{R}^{r\times r}, \\
\hat{y} & = & \hat{C}(p)\hat{x}, & \hat{B}(p) \in \mathbb{R}^{r\times m},\ \hat{C}(p) \in \mathbb{R}^{q\times r},
\end{array}
$$

of order $r \ll n$, such that w.r.t. some appropriate multivariate function space

$$\|y - \hat{y}\| = \left\|Gu - \hat{G}u\right\| \leq \left\|G - \hat{G}\right\| \cdot \|u\| < \text{tolerance} \cdot \|u\|$$

$\Longrightarrow$ Approximation problem: $\displaystyle \min_{\text{order}\,(\hat{G})\leq r} \left\|G - \hat{G}\right\|.$

Generation of Reduced-Order Model

Parametric System

$$\Sigma(p) : \begin{cases} E(p)\dot{x}(t;p) & = & A(p)x(t;p) + B(p)u(t), \\ y(t;p) & = & C(p)x(t;p). \end{cases}$$

Generation of Reduced-Order Model

Parametric System

$$\Sigma(p): \begin{cases} E(p)\dot{x}(t;p) &= A(p)x(t;p) + B(p)u(t), \\ y(t;p) &= C(p)x(t;p). \end{cases}$$

Parametric model reduction goal:

preserve parameters as *symbolic quantities* in reduced-order model:

$$\widehat{\Sigma}(p): \begin{cases} \hat{E}(p)\dot{\hat{x}}(t;p) &= \hat{A}(p)\hat{x}(t;p) + \hat{B}(p)u(t), \\ \hat{y}(t;p) &= \hat{C}(p)\hat{x}(t;p) \end{cases}$$

with states $\hat{x}(t;p) \in \mathbb{R}^r$ and $r \ll n$.

Generation of Reduced-Order Model

Parametric System

$$\Sigma(p) : \left\{ \begin{array}{rcl} E(p)\dot{x}(t;p) & = & A(p)x(t;p) + B(p)u(t), \\ y(t;p) & = & C(p)x(t;p). \end{array} \right.$$

Assuming parameter-affine representation:

$$\begin{array}{rcl} E(p) & = & E_0 + e_1(p)E_1 + \ldots + e_{q_E}(p)E_{q_E}, \\ A(p) & = & A_0 + a_1(p)A_1 + \ldots + a_{q_A}(p)A_{q_A}, \\ B(p) & = & B_0 + b_1(p)B_1 + \ldots + b_{q_B}(p)B_{q_B}, \\ C(p) & = & C_0 + c_1(p)C_1 + \ldots + c_{q_C}(p)C_{q_C}, \end{array}$$

allows easy parameter preservation for projection based model reduction.

PMOR Techniques

Petrov-Galerkin-type projection

For given projection matrices $V, W \in \mathbb{R}^{n\times r}$ with $W^T V = I_r$
($\rightsquigarrow (VW^T)^2 = VW^T$ is projector), compute

$$
\begin{aligned}
\hat{E}(p) &= W^T E_0 V + e_1(p) W^T E_1 V + \ldots + e_{q_E}(p) W^T E_{q_E} V \\
\hat{A}(p) &= W^T A_0 V + a_1(p) W^T A_1 V + \ldots + a_{q_A}(p) W^T A_{q_A} V \\
\hat{B}(p) &= W^T B_0 + b_1(p) W^T B_1 + \ldots + b_{q_B}(p) W^T B_{q_B} \\
\hat{C}(p) &= C_0 V + c_1(p) C_1 V + \ldots + c_{q_C}(p) C_{q_C} V
\end{aligned}
$$

Petrov-Galerkin-type projection

For given projection matrices $V, W \in \mathbb{R}^{n\times r}$ with $W^T V = I_r$
($\rightsquigarrow (VW^T)^2 = VW^T$ is projector), compute

$$
\begin{aligned}
\hat{E}(p) &= W^T E_0 V + e_1(p) W^T E_1 V + \ldots + e_{q_E}(p) W^T E_{q_E} V \\
&= \hat{E}_0 + e_1(p)\hat{E}_1 + \ldots + e_{q_E}(p)\hat{E}_{q_E} \\
\hat{A}(p) &= W^T A_0 V + a_1(p) W^T A_1 V + \ldots + a_{q_A}(p) W^T A_{q_A} V \\
&= \hat{A}_0 + a_1(p)\hat{A}_1 + \ldots + a_{q_A}(p)\hat{A}_{q_A} \\
\hat{B}(p) &= W^T B_0 + b_1(p) W^T B_1 + \ldots + b_{q_B}(p) W^T B_{q_B} \\
&= \hat{B}_0 + b_1(p)\hat{B}_1 + \ldots + b_{q_B}(p)\hat{B}_{q_B} \\
\hat{C}(p) &= C_0 V + c_1(p) C_1 V + \ldots + c_{q_C}(p) C_{q_C} V \\
&= \hat{C}_0 + c_1(p)\hat{C}_1 + \ldots + c_{q_C}(p)\hat{C}_{q_C}
\end{aligned}
$$

But: affine parametrization not always given, in particular in model-free, data-driven context!

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n\times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation (= interpolate $y(s,.)$ in frequency domain) [B./Baur 2008/09]

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation ($=$ interpolate $y(s, .)$ in frequency domain) [B./Baur 2008/09]
3. matrix interpolation [Panzer/Mohring/Eid/Lohmann 2010, Amsallam/Farhat 2011]

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation ($=$ interpolate $y(s, .)$ in frequency domain) [B./Baur 2008/09]
3. matrix interpolation [Panzer/Mohring/Eid/Lohmann 2010, Amsallam/Farhat 2011]

Advantage:

no need for affine parametrization, requires only system matrices $A(p^{(k)}), B(p^{(k)}), \ldots$.

Disadvantages:

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation ($=$ interpolate $y(s, .)$ in frequency domain) [B./Baur 2008/09]
3. matrix interpolation [Panzer/Mohring/Eid/Lohmann 2010, Amsallam/Farhat 2011]

Advantage:

no need for affine parametrization, requires only system matrices $A(p^{(k)}), B(p^{(k)}), \ldots$.

Disadvantages:

1. manifold interpolation: originally, requires $\mathcal{O}(nr)$ operations in "online" phase. [Brunsch 2017] overcomes this problem, but only for negative definite matrix pencils $(A(p), E(p))$.

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n \times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation ($=$ interpolate $y(s, .)$ in frequency domain) [B./Baur 2008/09]
3. matrix interpolation [Panzer/Mohring/Eid/Lohmann 2010, Amsallam/Farhat 2011]

Advantage:

no need for affine parametrization, requires only system matrices $A(p^{(k)}), B(p^{(k)}), \ldots$.

Disadvantages:

1. manifold interpolation: originally, requires $\mathcal{O}(nr)$ operations in "online" phase. [Brunsch 2017] overcomes this problem, but only for negative definite matrix pencils $(A(p), E(p))$.
2. transfer function interpolation: spurious poles of the parametric transfer function.

PMOR Techniques

Basis Generation — Global vs. Local

Local Bases

Obtain $V_k, W_k \in \mathbb{R}^{n\times r_k}$ using any non-parametric linear MOR method for a number of full-order models $\Sigma(p^{(k)})$, $k = 1, \ldots, \ell$. Then compute reduced-order model by

1. manifold interpolation [Amsallam/Farhat 2008, Brunsch 2017]
2. transfer function interpolation (= interpolate $y(s, .)$ in frequency domain) [B./Baur 2008/09]
3. matrix interpolation [Panzer/Mohring/Eid/Lohmann 2010, Amsallam/Farhat 2011]

Advantage:

no need for affine parametrization, requires only system matrices $A(p^{(k)}), B(p^{(k)}), \ldots$.

Disadvantages:

1. manifold interpolation: originally, requires $\mathcal{O}(nr)$ operations in "online" phase. [Brunsch 2017] overcomes this problem, but only for negative definite matrix pencils $(A(p), E(p))$.
2. transfer function interpolation: spurious poles of the parametric transfer function.
3. matrix interpolation: different models obtained in different coordinate systems $\rightsquigarrow$ Procrustes problem $\rightsquigarrow$ potential loss of accuracy; efficiency in "online" phase suffers from evaluating the interpolation operator.

Basis Generation — Global vs. Local

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-) Galerkin projection, exploiting affine parametrization of the linear parametric system.

Basis Generation — Global vs. Local

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-) Galerkin projection, exploiting affine parametrization of the linear parametric system.

Obtain global basis from

1. concatenation of local basis matrices:

$$V := [\, V_1, \ldots, V_\ell \,], \qquad W := [\, W_1, \ldots, W_\ell \,]$$

   and orthogonalization (truncation), using, e.g., SVD;

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-) Galerkin projection, exploiting affine parametrization of the linear parametric system.

Obtain global basis from

1. concatenation of local basis matrices:

$$V := [\, V_1, \ldots, V_\ell \,], \qquad W := [\, W_1, \ldots, W_\ell \,]$$

   and orthogonalization (truncation), using, e.g., SVD;
2. bilinearization and using bilinear MOR techniques;

Basis Generation — Global vs. Local

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-) Galerkin projection, exploiting affine parametrization of the linear parametric system.

Obtain global basis from

1. concatenation of local basis matrices:

$$V := [\, V_1, \ldots, V_\ell \,], \qquad W := [\, W_1, \ldots, W_\ell \,]$$

   and orthogonalization (truncation), using, e.g., SVD;
2. bilinearization and using bilinear MOR techniques;
3. parametric balanced truncation [Son/Stykel 2017].

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-)Galerkin projection, exploiting affine parametrization of the linear parametric system.

Obtain global basis from

1. concatenation of local basis matrices:

$$V := [\, V_1, \ldots, V_\ell \,], \qquad W := [\, W_1, \ldots, W_\ell \,]$$

   and orthogonalization (truncation), using, e.g., SVD;
2. bilinearization and using bilinear MOR techniques;
3. parametric balanced truncation [Son/Stykel 2017].

Avoids most of the problems encountered with local bases, but requires parameter-affine representation of system.

Basis Generation — Global vs. Local

Global Basis

Obtain $V, W \in \mathbb{R}^{n \times r_k}$ such that $V^T W = I_r$ and perform structure preserving (Petrov-) Galerkin projection, exploiting affine parametrization of the linear parametric system.

Obtain global basis from

1. concatenation of local basis matrices:

$$V := [\, V_1, \ldots, V_\ell \,], \qquad W := [\, W_1, \ldots, W_\ell \,]$$

   and orthogonalization (truncation), using, e.g., SVD;
2. bilinearization and using bilinear MOR techniques;
3. parametric balanced truncation [Son/Stykel 2017].

Avoids most of the problems encountered with local bases, but requires parameter-affine representation of system.

Here: no basis available, but only reduced-order models for different p-values!

Outline

Interpolating Reduced Models obtained from Data

The Non-Parametric Loewner Framework in a Nutshell

State-space system (unknown)

$$\Sigma(s) \equiv \begin{cases} (sE - A)x(s) &= Bu(s), \\ y(s) &= Cx(s). \end{cases}$$

transfer function (TF) (only known as operator)

$$G(s) = C(sE - A)^{-1}B.$$

Algorithm: identify reduced-order model (ROM) from data.

Step 1: Collect data: (V, W are composed of tangential directions!)

- "Right Data": (λ_i, r_i, w_i) satisfying $G(\lambda_i)r_i = w_i$;
- "Left Data": (μ_j, ℓ_j, v_j) satisfying $\ell_j G(\mu_j) = v_j$.

Step 2: Compute the *Loewner matrix* $\mathbb{L}$ and the *shifted Loewner matrix* $\mathbb{L}_\sigma$.

$$(\mathbb{L})_{ij} = \frac{v_i r_j - \ell_i w_j}{\mu_i - \lambda_j}, \qquad (\mathbb{L}_\sigma)_{ij} = \frac{\mu_i v_i r_j - \ell_i w_j \lambda_j}{\mu_i - \lambda_j}.$$

Step 3: Compute the reduced model:

- If the matrix pencil $(\mathbb{L}_\sigma, \mathbb{L})$ is regular, the reduced model is: $\widehat{E} = -\mathbb{L}$, $\widehat{A} = -\mathbb{L}_\sigma$, $\widehat{B} = V$, $\widehat{C} = W$.
- If the matrix pencil $(\mathbb{L}_\sigma, \mathbb{L})$ is (numerically) singular:
  1. Compute rank-revealing SVD: $s\mathbb{L} - \mathbb{L}_\sigma = Y\Sigma X^* \approx Y_k \Sigma_k X_k^*$; $(s \in \{\lambda_i\} \cup \{\mu_j\})$
  2. Compute $\widehat{E} = -Y_k^* \mathbb{L} X_k$, $\widehat{A} = -Y_k^* \mathbb{L}_\sigma X_k$, $\widehat{B} = Y_k^* V$, $\widehat{C} = WX_k$.

Generating a Parametric Model in the Loewner Framework

In summary, we have two potential representations:

- The “original” representation:
  - $\widehat{E} = -\mathbb{L}$, $\widehat{A} = -\mathbb{L}_\sigma$, $\widehat{B} = V$, $\widehat{C} = W$.
  - Very likely to yield numerically singular matrix pencil!
- The “compressed” representation:
  - $\widehat{E} = -Y_k^* \mathbb{L} X_k$, $\widehat{A} = -Y_k^* \mathbb{L}_\sigma X_k$, $\widehat{B} = Y_k^* V$, $\widehat{C} = W X_k$.
  - Yields regular matrix pencil.

Generating a Parametric Model in the Loewner Framework

In summary, we have two potential representations:

- The "original" representation:
  - $\widehat{E} = -\mathbb{L}$, $\widehat{A} = -\mathbb{L}_\sigma$, $\widehat{B} = V$, $\widehat{C} = W$.
  - Very likely to yield numerically singular matrix pencil!
- The "compressed" representation:
  - $\widehat{E} = -Y_k^* \mathbb{L} X_k$, $\widehat{A} = -Y_k^* \mathbb{L}_\sigma X_k$, $\widehat{B} = Y_k^* V$, $\widehat{C} = W X_k$.
  - Yields regular matrix pencil.

Question: How to interpolate the ROMs built by the Loewner Framework

Generating a Parametric Model in the Loewner Framework

In summary, we have two potential representations:

- The "original" representation:
  - $\widehat{E} = -\mathbb{L}$, $\widehat{A} = -\mathbb{L}_\sigma$, $\widehat{B} = V$, $\widehat{C} = W$.
  - Very likely to yield numerically singular matrix pencil!
- The "compressed" representation:
  - $\widehat{E} = -Y_k^* \mathbb{L} X_k$, $\widehat{A} = -Y_k^* \mathbb{L}_\sigma X_k$, $\widehat{B} = Y_k^* V$, $\widehat{C} = W X_k$.
  - Yields regular matrix pencil.

Question: How to interpolate the ROMs built by the Loewner Framework

- There is no "FOM" in the Loewner Framework, and no local or global bases V and W like in projection-based methods.
- The Loewner Framework has been extended to yield parametric models for one design parameter.
- Here: explore another possibility—interpolating nonparametric ROMs built under the Loewner Framework.

Interpolating the "Original" Representation

- Assume that the system is parameterized with p.
- Assume that we use the same frequencies and the same left/right input vectors $\forall\ p$.
- In $G_p(\lambda_i)r_i = w_i(p)$ and $\ell_j G_p(\mu_j) = v_j(p)$, λ_j, μ_j and r_j, ℓ_j are independent of p, $1 \le i \le n_p$.
- Interpolating $\mathbb{L}(p)$ and $\mathbb{L}_\sigma(p)$ using interpolation operator $\tilde{g}(p) = \sum_{q=1}^{n_p} \phi_q(p) g(p_q)$

  is equivalent to

  interpolating $V(p)$ and $W(p)$ and using the Loewner framework:

$$\left(\widetilde{\mathbb{L}}(p)\right)_{ij} = \sum_{q=1}^{n_p} \left(\frac{v_i(p_q) r_j - \ell_i w_j(p_q)}{\mu_i - \lambda_j} \right) \phi_q(p) = \frac{\widetilde{v}_i(p) r_j - \ell_i \widetilde{w}_j(p)}{\mu_i - \lambda_j}$$

$$\left(\widetilde{\mathbb{L}_\sigma}(p)\right)_{ij} = \sum_{q=1}^{n_p} \left(\frac{\mu_i v_i(p_q) r_j - \ell_i w_j(p_q) \lambda_j}{\mu_i - \lambda_j} \right) \phi_q(p) = \frac{\mu_i \widetilde{v}_i(p) r_j - \ell_i \widetilde{w}_j(p) \lambda_j}{\mu_i - \lambda_j}$$

- In the "original representation":
  $\widehat{E}(p) = -\widetilde{\mathbb{L}}(p), \quad \widehat{A}(p) = -\widetilde{\mathbb{L}_\sigma}(p), \quad \widehat{B}(p) = \widetilde{V}(p), \quad \widehat{C}(p) = \widetilde{W}(p).$
  - They are all linear functions of $V(p)$ and $W(p)$.
  - Equivalent to interpolating the transfer functions.

Interpolating Reduced Models obtained from Data

Interpolating the "Compressed" Representation

- "Original" representations numerically problematic and storage-intensive.
- The ultimate goal is to interpolate the "compressed" representation.
- The bases used to compress the "original" representation vary with the parameter (with additional freedom in s_i):
$$\boxed{s_i\mathbb{L}(p_i) - \mathbb{L}_\sigma(p_i) = Y_i\Sigma_i X_i^* \approx Y_{i,k}\Sigma_{i,k}X_{i,k}^*}$$
- **Idea 1**: To preserve the interpolation property of the original representation at $(-\mathbb{L}(p_i), -\mathbb{L}_\sigma(p_i), V(p_i), W(p_i))$, use common bases Y and X to reduce them.
- **Idea 2a**: Y should contain the dominant components of all "generalized observability" matrices Y_i. So we compute Y by the SVD (with $\mathbb{L}_i = \mathbb{L}(p_i)$ etc.):
$$\left[\, s_1\mathbb{L}_1 - \mathbb{L}_{\sigma\,1} \;\middle|\; s_2\mathbb{L}_2 - \mathbb{L}_{\sigma\,2} \;\middle|\; \dots \;\middle|\; s_{n_q}\mathbb{L}_{n_q} - \mathbb{L}_{\sigma\,n_p} \,\right] = Y_o\,\Sigma_o\,X_o^* \approx Y_K\Sigma_{o,K}X_{o,K}^*.$$
- **Idea 2b**: X should contain the dominant components of all "generalized controllability" matrices X_i. So we compute X by the SVD:
$$\begin{bmatrix} s_1\mathbb{L}_1 - \mathbb{L}_{\sigma\,1} \\ s_2\mathbb{L}_2 - \mathbb{L}_{\sigma\,2} \\ \dots \\ s_{n_q}\mathbb{L}_{n_q} - \mathbb{L}_{\sigma\,n_p} \end{bmatrix} = Y_c\Sigma_c X_c^* \approx Y_{c,K}\Sigma_{c,K}X_K^*.$$

Interpolating the "Compressed" Representation (Continued)

- Obviously, colspan$\{Y_i\} \subseteq$ colspan$\{Y_o\}$, rowspan$\{X_i\} \subseteq$ rowspan$\{X_c\}$.
- Therefore, Y_K and X_K computed from truncated SVD include the dominant components for all generalized observability and controllability matrices corresponding to the samples.

Algorithm 1 Interpolation of Loewner ROMs in the Compressed Representation

1: Build the common basis matrix Y_K by computing the truncated SVD

$$\left[\, s_1\mathbb{L}_1 - \mathbb{L}_{\sigma\,1} \;\middle|\; s_2\mathbb{L}_2 - \mathbb{L}_{\sigma\,2} \;\middle|\; \dots \;\middle|\; s_{n_q}\mathbb{L}_{n_q} - \mathbb{L}_{\sigma\,n_p} \,\right] = Y_o\,\Sigma_o\,X_o^*, \quad Y_K := Y_o(:, 1:K).$$

2: Build the common basis matrix X_K by computing the truncated SVD

$$\begin{bmatrix} s_1\mathbb{L}_1 - \mathbb{L}_{\sigma\,1} \\ s_2\mathbb{L}_2 - \mathbb{L}_{\sigma\,2} \\ \dots \\ s_{n_q}\mathbb{L}_{n_q} - \mathbb{L}_{\sigma\,n_p} \end{bmatrix} = Y_c\Sigma_c X_c^*, \quad X_K := X_c(:, 1:K).$$

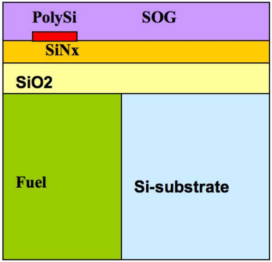
3: Build the "compressed" representation using the common bases:

$$\widehat{E}_l = -Y_K^*\mathbb{L}_l X_K, \quad \widehat{A}_l = -Y_K^*\mathbb{L}_{\sigma,l}X_K, \quad \widehat{B}_l = Y_K^* V_l, \quad \widehat{C}_l = W_l X_K, \quad l = 1, \dots, n_p.$$

4: Given an interpolation operator, the interpolated ROM at p_* is given by

$$\widehat{M}(p) = \sum_{l=1}^{n_p} \widehat{M}_l \phi_l(p), \qquad \widehat{M} \in \{\widehat{E}, \widehat{A}, \widehat{B}, \widehat{C}\}.$$

Numerical Results 1: Microthruster Model



- MEMS device for jet propulsion.
- Single input/single output (SISO) configuration.
- Assume that only input/output information is available.
- Parametrization by the film coefficient.

Source: MOR Wiki: http://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Micropyros_Thruster

Numerical Results 1: Microthruster Model

Interpolating the "Compressed" Representation using Algorithm 1:

- $p_1, p_2, \ldots, p_{29}$ are uniformly distributed in the interval $[10, 7200]$.
- ROMs of size 10 are built by the Loewner Framework at p_1, p_8, p_{15}, p_{22}, p_{29}.
- The proposed method with cubic spline interpolation is used to get ROMs for the other p values.

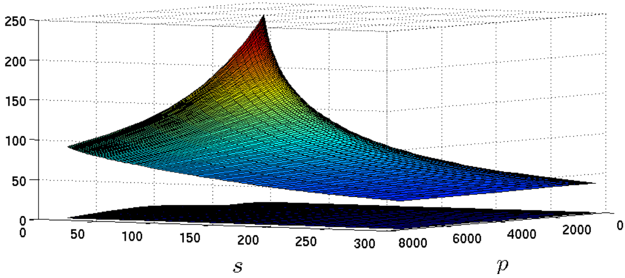


Figure: Response Surface and Absolute Error

Numerical Results 2: The “FOM” Model

We use a parametric version of the SLICOT “FOM” model to generate TF samples, which will be used by the Loewner Framework to build ROMs.

$$\Sigma(s;p) \equiv \begin{cases} (sI - A(p))\, x(s;p) & = & Bu(s), \\ y(s;p) & = & Cx(s;p), \end{cases}$$

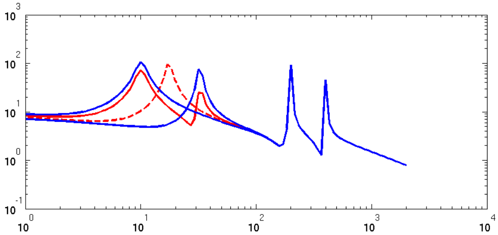
with

$$\begin{aligned} A(p) &= \operatorname{diag}(A_1(p), A_2, A_3, A_4), \\ A_1(p) &= \begin{bmatrix} -1 & p \\ -p & -1 \end{bmatrix} \quad A_2 = \begin{bmatrix} -1 & 200 \\ -200 & -1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} -1 & 400 \\ -400 & -1 \end{bmatrix}, \\ A_4 &= -\operatorname{diag}(1, 2, \dots, 1000), \end{aligned}$$

and $C = B^{\mathsf{T}} = [10, 10, 10, 10, 10, 10, 1, 1, \dots, 1]$.

Source: http://slicot.org/20-site/126-benchmark-examples-for-model-reduction

Numerical Results 2: The "FOM" model



- Blue curves: TFs at p_1 and p_2, where we build ROMs by the Loewner Framework.
- Solid red curve: TF of the interpolated ROM at p_*.
- Dashed red curve: true TF at p_*.

- These numerical results show the limitations of TF interpolation methods in general ("ghost poles"):
  - For the region far from the moving pole, the interpolation result is accurate.
  - However, the movement of a pole is not captured: the movement of a pole is "modeled" by waxing and waning of two poles.
  - Note that the poles of the interpolated system

$$\sum_{i=1}^{n_p} w(p)C(p_i)(sI - A(p_i))^{-1}B(p_i)$$

    is the union of the poles of all the systems used for interpolation, all poles are fixed in position!

Outline

1. Introduction
2. Interpolating Reduced Models obtained from Data
3. PMOR by Pole-Matching
4. Conclusions

PMOR by Pole-Matching

Motivation

- Capture the movements of poles by interpolating the poles explicitly.
- Use a “canonical realization” that is suitable for interpolation.
- Using the “canonical realization”, we should be able to interpolate ROMs built by different MOR methods.

PMOR by Pole-Matching

Motivation

- Capture the movements of poles by interpolating the poles explicitly.
- Use a "canonical realization" that is suitable for interpolation.
- Using the "canonical realization", we should be able to interpolate ROMs built by different MOR methods.

(Modified/Real) Modal Representation.

Assume reduced-order model is in modal coordinates,

$$P^{-1}\hat{A}P = \Lambda = \begin{bmatrix} \Lambda_1 & & & \\ & \Lambda_2 & & \\ & & \ddots & \\ & & & \Lambda_m \end{bmatrix}, \qquad P = [P_1,\, P_2,\, \ldots,\, P_m]\,,$$

where for a real eigenpair (λ_j, v_j),

$$\Lambda_j = [\lambda_j] \qquad \text{and} \qquad P_j = \left[v_j\right],$$

while for a complex eigenpair $(a_j \pm bw_j, r_j \pm iq_j)$,

$$\Lambda_j = \begin{bmatrix} a_j & b_j \\ -b_j & a_j \end{bmatrix} \quad \text{and} \quad P_j = [r_j \quad q_j].$$

PMOR by Pole-Matching

The Modified Modal Representation (Pole-Residue Form)

Let

$$G(s) = C(sI - A)^{-1}B = C(sI - P\Lambda P^{-1})^{-1}B = CP(sI - \Lambda)^{-1}P^{-1}B =: C^{\mathrm{I}}(SI - \Lambda)^{-1}B^{\mathrm{I}},$$

and partition

$$C^{\mathrm{I}} = [C_1^{\mathrm{I}}, C_2^{\mathrm{I}}, \ldots, C_m^{\mathrm{I}}], \quad B^{\mathrm{I}} = [B_1^{\mathrm{I}}, B_2^{\mathrm{I}}, \ldots, B_m^{\mathrm{I}}]^T,$$

where the vectors C_j^{I} and B_j^{I} are of size 1 (if the j-th eigenvalue is real) or 2 (if the j-th eigenvalue is complex). Then

$$G(s) = \sum_{j=1}^{m} C_j^{\mathrm{I}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{I}}$$

For a real eigenpair (λ_j, v_j), we define

$$C_j^{\mathrm{II}} = C_j^{\mathrm{I}}B_j^{\mathrm{I}} \quad \text{and} \quad B_j^{\mathrm{II}} = 1$$

and derive

$$C_j^{\mathrm{I}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{I}} = \frac{C_j^{\mathrm{I}}B_j^{1}}{s - \lambda_j} = C_j^{\mathrm{II}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{II}},$$

PMOR by Pole-Matching

Modified Modal Representation (Pole-Residue Form), continued

For a complex eigenpair $(a_j \pm bw_j, r_j \pm iq_j)$, we first define $C_j^{\mathrm{I}} = [C_{j,1}^{\mathrm{I}}, C_{j,2}^{\mathrm{I}}]$ and $B_j^{\mathrm{I}} = [B_{j,1}^{\mathrm{I}}, B_{j,2}^{\mathrm{I}}]^{\mathrm{T}}$, and then derive

$$
\begin{aligned}
C_j^{\mathrm{I}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{I}} &= \frac{C_j^{\mathrm{I}} \begin{bmatrix} s - a_j & b_j \\ -b_j & s - a_j \end{bmatrix} B_j^{\mathrm{I}}}{(s-a_j)^2 + b_j^2} \\
&= \frac{(C_{j,1}^{\mathrm{I}}B_{j,1}^{\mathrm{I}} + C_{j,2}^{\mathrm{I}}B_{j,2}^{\mathrm{I}},\ C_{j,2}^{\mathrm{I}}B_{j,1}^{\mathrm{I}} - C_{j,1}^{\mathrm{I}}B_{j,2}^{\mathrm{I}}) \begin{bmatrix} s - a_j & b_j \\ -b_j & s - a_j \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{(s-a_j)^2 + b_j^2} \\
&= C_j^{\mathrm{II}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{II}},
\end{aligned}
$$

where we define $C_j^{\mathrm{II}} = (C_{j,1}^{\mathrm{I}}B_{j,1}^{\mathrm{I}} + C_{j,2}^{\mathrm{I}}B_{j,2}^{\mathrm{I}},\ C_{j,2}^{\mathrm{I}}B_{j,1}^{\mathrm{I}} - C_{j,1}^{\mathrm{I}}B_{j,2}^{\mathrm{I}})$ and $B_j^{\mathrm{II}} = [1, 0]^{\mathrm{T}}$. Therefore,

$$
y = \sum_{j=1}^{m} C_j^{\mathrm{I}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{I}} = \sum_{j=1}^{m} C_j^{\mathrm{II}}(sI - \Lambda_j)^{-1}B_j^{\mathrm{II}} = C^{\mathrm{II}}(sI - \Lambda)^{-1}B^{\mathrm{II}},
$$

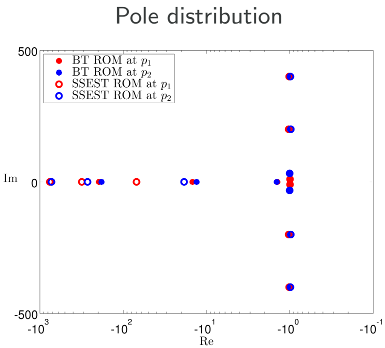
where $C^{\mathrm{II}} = [C_1^{\mathrm{II}}, C_2^{\mathrm{II}}, \ldots, C_m^{\mathrm{II}}]$ and $B^{\mathrm{II}} = [B_1^{\mathrm{II},\mathrm{T}}, B_2^{\mathrm{II},\mathrm{T}}, \ldots, B_m^{\mathrm{II},\mathrm{T}}]^{\mathrm{T}}$.

PMOR by Pole-Matching

The Modified Modal Representation — some remarks

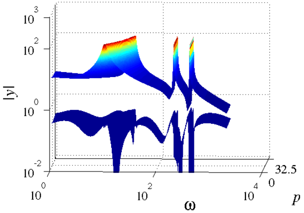
- In the modified modal representation, interpolation of ROMs can be done in two steps:
  1. Pole matching—in practice, this is achieved by just reordering the blocks in C^{II}, Λ and B^{II}.
  2. Interpolating the positions and amplitudes of the poles, equivalent to interpolating Λ and C^{II}.
- If for a system, the positions of poles change little when the parameter(s) change, pole matching is easy.
  $\rightsquigarrow$ Some effort in algorithmic eigenvalue continuation to match the right poles!
- Semisimple eigenvalues can essentially be treated like simple eigenvalues.
- Systems with defective eigenvalues do not have a modified modal realization.
  In practice, we check the condition number of P and return a warning when it gets too large.

Numerical Results 1: the "FOM" Model

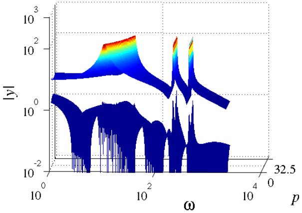


- Two MOR methods are used: balanced truncation (BT) and a method based on system identification (`ssest` in MATLAB).
- All ROMs are of size 10.
- It is clear how poles should be paired for ROMs built by the same method.

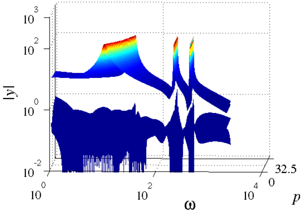
Numerical Results 1: the “FOM” Model



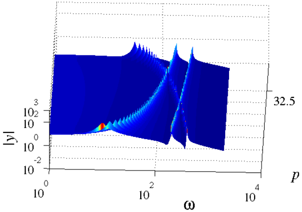
Interpolation of BT ROMs



Interpolation of ssest ROMs



Interpolation of a BT and a ssest ROM

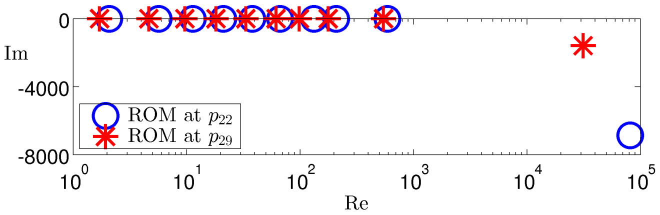


Mismatched poles

PMOR by Pole-Matching

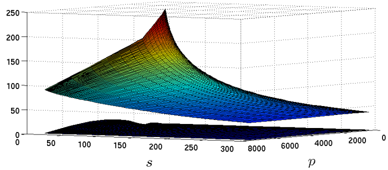
Numerical Results 2: the Microthruster Model

The poles of the ROMs for p_{22} and p_{29}, respectively.

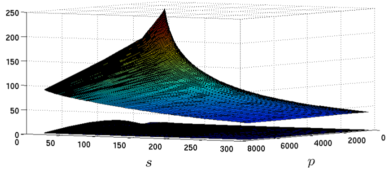


- In this example, A is complex and none of the eigenvalues are paired.
- Therefore, Λ is diagonal. The form of the modified model realization is the same as the case when all eigenvalues are real.

Numerical Results 2: the Microthruster Model



Linear Interpolation
(Loewner Representation)



Linear Interpolation
(Modified Modal Representation)

PMOR by Pole-Matching

Numerical Results 3: A Footbridge Model

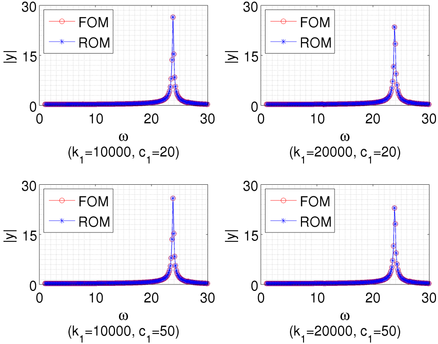
The full model is a second-order system:

$$\left(K_0 + i\omega C_0 + (k_1 + i\omega c_1)K_i - \omega^2 M_0\right)x = f,$$
$$y = \ell^* x,$$

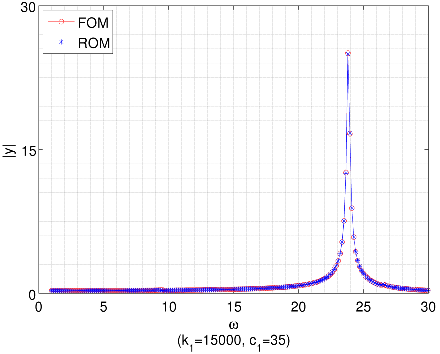
- The footbridge is located over the Dijle river in Mechelen (Belgium).
- The size of the discretized system is 25,963.
- The two parameters k_1 and c_1 represent the stiffness and the viscosity of a tuned mass damper, respectively.
- We use a Krylov method to reduce the equivalent first-order system.
- The ROMs are of order 10.

Numerical Results 3: A Footbridge Model

The ROMs Used for Interpolation



The Interpolated ROM



Outline

Conclusions

- A ROM interpolation method for data-driven modeling (in particular the Loewner Framework) is proposed. However, the ghost pole problem is present in the case of poles moving with parameter variation.
- A pole-matching method is proposed based on a modified modal representation (pole-residue form).
  - It does not need explicit or even affine parameter dependence in the FOM.
  - It does not assume the existence of the FOM and works well also with data-driven ROMs.
  - It can even interpolate ROMs resulting from different MOR methods.
  - It is relatively insensitive to the number of parameters and the complexity of (e.g., nonlinear and non-affine) parameter dependence.
  - If we only use linear interpolation, stability is preserved. In other case (easily detected by checking the poles), we can return to linear interpolation when the interpolated ROM is not stable.
  - The method proves to interpolate ROMs built by different methods: balanced truncation, Krylov methods, the Loewner Framework, the `ssest` method.
- Realization is important for ROM interpolation:
  - In Loewner representation, we interpolate the transfer function;
  - In the modified modal representation, we interpolate the positions and amplitudes of the poles.

References

N. Banagaaya, P. Benner, L. Feng, P. Meuris, and W. Schoenmaker.
An index-aware parametric model order reduction method for parametrized quadratic differential-algebraic equations.
Applied Mathematics and Computation, 319:409–424, 2018.

U. Baur, C. Beattie, P. Benner, and S. Gugercin.
Interpolatory projection methods for parameterized model reduction.
SIAM Journal on Scientific Computing 33(5):2489–2518, 2011.

U. Baur and P. Benner.
Model reduction for parametric systems using balanced truncation and interpolation.
at-Automatisierungstechnik 57(8):411–419, 2009.

U. Baur, P. Benner, A. Greiner, J.G. Korvink, J. Lienemann, and C. Moosmann.
Parameter preserving model order reduction for MEMS applications.
Mathematical and Computer Modelling of Dynamical Systems 17(4):297–317, 2011.

P. Benner and A. Bruns.
Parametric model order reduction of thermal models using the bilinear interpolatory rational Krylov algorithm.
Mathematical and Computer Modelling of Dynamical Systems, 21(2):103–129, 2015.

P. Benner, L. Feng, and Y. Yue.
Parammetric model order reduction in data-driven frameworks.
In preparation.

P. Benner, S. Gugercin, and K. Willcox.
A survey of model reduction methods for parametric systems.
SIAM Review 57(4):483–531, 2015.