# Homework: Model order reduction

November 21, 2022

This is an assignment for the course 'Numerical Linear Algebra'. The goal of this assignment is to explore model order reduction techniques as seen in the lectures and apply them to some interesting real-life models.
**Please note that the questions are deliberately open-ended! Unlike in the exercise sessions, you are encouraged to make your own analysis, provide your own results and construct your own report in your own way. The assignment is individual.** Don't be discouraged if not everything works in the end, but show your work and report what you tried and whether you know why something did not work.
**Your report should be a standalone text, should read pleasantly and should not refer to this assignment. Take special care of structure and visual presentation. You are allowed to copy text from this assignment document.**

## 1 Theory

In this assignment we will be analyzing large linear systems arising from a variety of applications, and the ways in which we can reduce the complexity of such systems. This can be seen as reducing the order of the models involved hence *Model Order Reduction*. In particular, we will look at three types of large state space models.

### 1.1 Basic state space models

Given a simple ODE

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b} \cdot u(t) \tag{1}$$

with solution $\mathbf{q}(t) \in \mathbb{R}^n$, and some input $u(t) \in \mathbb{R}$ we can transform this using the Laplace transform (**refresh your knowledge of the Laplace transform!**) and slightly abusing notation (recycling the variable $\mathbf{x}$) into

$$s\mathbf{x} - A\mathbf{x} = \mathbf{b}u.$$

We are not always interested in $\mathbf{x}$, but rather some functional applied to $\mathbf{x}$, which results in the state space model

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{2}$$

State space models of this kind will be referred to in this text as *basic state space models*.

## 1.2 State space models from mechanics

Typical ODEs for mechanical systems take the form

$$M\ddot{\mathbf{x}} + D\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{b} \cdot u(t). \tag{3}$$

Here $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{b} \cdot u(t) \in \mathbb{R}^n$ are vectors varying over time (note the special structure of the time variation of the right hand side!). Typically, $\mathbf{x}$ is a vector of displacements of degrees-of-freedom (dofs) arising from a Finite Element Method (FEM) discretization of the original continuous problem. The vector $\mathbf{b} \cdot u(t)$ then typically corresponds to a force exerted on the structure and is called the *input*. The matrices $M$, $D$ and $K$ are called the *mass*-, *damping*- and *stiffness matrix* respectively.

**Transform equation 3 into an equivalent ODE of the form**

$$\dot{\mathbf{q}} = A\mathbf{q} + \mathbf{f} \cdot u(t) \tag{4}$$

**with $\mathbf{q}(t) \in \mathbb{R}^{2n}$. NOTE THE CHANGE IN DIMENSION. Use the most natural transformation. You can assume that $M$ is invertible. Again, using the Laplace transform and a slight abuse of notation we re-write equation this as**

$$s\mathbf{x} - A\mathbf{x} = \mathbf{b}u \tag{5}$$

Note that after the two transformations introduced before, these variable are linked, but not at all identical to their original counterparts.

**What if we generalize, and instead look for a Laplace domain system of the form**

$$sE\mathbf{x} - A\mathbf{x} = \mathbf{b}u \tag{6}$$

**with $E \in \mathbb{R}^{2n}$ not necessarily the identity? What is then the most natural such formulation? What important advantage does it have? Show also that, under the corresponding assumptions, each of the above models is equivalent to the original model.**

Typically, we are not interested in all of $x$, but only in some (scalar) function $y$ of $\mathbf{x}$. This is written as

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{7}$$

Equation 14 is referred to as the *state space* model of the mechanical system. **Show that, under the assumptions made earlier about this model, it is equivalent to the model**

$$\begin{aligned} s^2M\mathbf{x} + sD\mathbf{x} + K\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{8}$$

**with $x \in \mathbf{R}^n$.** The model above is referred to as the *quadratic model* of the mechanical system.

The most important data associated to a state space model is the so-called *transfer function* $H(s) := y(s)/u(s)$. **Interpret the meaning of the transfer function. Show that it can be written as**

$$\begin{aligned} H(s) &= \mathbf{c}^T(sI - A)^{-1}\mathbf{b} \\ &= \sum_{i=1}^{n} \frac{(\mathbf{c}^T\mathbf{p}_i)(\mathbf{q}_i^*\mathbf{b})}{s - \lambda_i} \end{aligned} \tag{9}$$

**when $A = P\Lambda Q^*$ is the eigendecomposition of $A$. You can assume that this eigendecomposition exists as well as that $P = [\mathbf{p}_1 \cdots \mathbf{p}_n]$, $Q = [\mathbf{q}_1 \cdots \mathbf{q}_n]$ and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$. HINT: show that**

$$(sI - A)^{-1} = \sum_{i=1}^{n} \frac{\mathbf{p}_i\mathbf{q}_i^*}{s - \lambda_i}. \tag{10}$$

The second form of equation 9 we will call the *residual form* of the transfer function. **What does $H$ look like in terms of the quadratic model? Describe the advantages of this formulation over the above (theoretical) descriptions.**

### 1.3 State space models from simple ODEs

Naturally we can consider any linear ODE to obtain a state space model i.e.

$$\begin{aligned} E\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + \mathbf{b} \cdot u(t) \\ y(t) &= \mathbf{c}^T \mathbf{x}(t), \end{aligned} \tag{11}$$

with $\mathbf{x}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ and $E, A \in \mathbb{R}^{n \times n}$ leading by way of the Laplace transform to the model

$$\begin{aligned} sE\mathbf{x} &= A\mathbf{x} + \mathbf{b}u \\ y &= \mathbf{c}^T \mathbf{x}. \end{aligned} \tag{12}$$

**Note again the abuse in notation, and that in general $E$ is not the identity. What does the transfer function look like now? Can it still be written in some residual form as in equation 9? If so, under what assumptions? HINT: this requires looking into the generalized eigenvalue decomposition.**

## 2 Applications

In this section three applications are outlined. You have been given code `bode_from_system` and `bode_from_function` that can produce bode plots for transfer functions given in descriptor form or in the form of a transfer function definition, as well as a script that produces these plots for the relevant frequency ranges. **Study the systems and these functions carefully.**

**All matrices and vectors have been provided in Matrix Market format or `.mat` format. In addition you have been provided a file called '`mmread.m`' which contains a function for reading in these matrices in MatLab.**

### 2.1 Spiral inductor

We look here at an integrated RF passive inductor, of the type 'spiral inductor'. Spiral inductors, which vary in structure, are among the most common types of on-chip inductors. Spiral inductors are usually characterized by the diameter, the line width, the number of turns, and the line-to-line space. In this case, the inductor has turns that are 40$\mu$m wide, 15$\mu$m thick, with a separation of $40\mu$m. The spiral is suspended $55\mu$m over a substrate by posts at the corners and centers of the turns in order to reduce the capacitance to the substrate. To make it also a proximity sensor, a $0.1\mu$m plane of copper is added $45\mu$m above the copper spiral. The overall extent of the suspended turns is 1.58mm $\times$ 1.58mm. The spiral inductor, including part of the overhanging copper plane, is shown in figure 1. The model is discretized using a

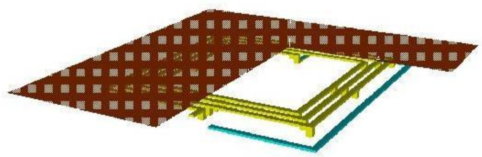

Figure 1: Spiral inductor

FEM-like technique called PEEC, which results in mesh specific inductance and resistance matrices $L$ and $R$ respectively, corresponding to the following differential state-space form:

$$L\frac{di_m}{dt} = Ri_m + Nv_p$$
$$i_p = N^T i_m$$

Here $i_m$ is the mesh current and $v_p$ and $i_p$ are the voltage and current at nodes of interest, with $N$ a 'natural' matrix mapping between these nodes and the mesh. **You do not need to understand this in detail! What does the associated Laplace domain state space model look like?** In the multi-gigahertz frequency range, the so-called 'skin effect' causes current to flow only at the surface of conductors (i.e. the copper coil and plane), leading to a decrease of wire inductance and an increase of resistance. Capturing the skin effect while also maintaining an accurate low frequency response is a challenge for many model reduction algorithms. For that reason the frequency range considered for this model is very wide: $\omega \in [1, 10^{10}]$. **You have been given a MatLab file containing all the necessary data for this system. Analyze the system in detail and report your findings. Use the function `bode_from_system` or `bode_from_function` to evaluate and plot transfer function in the frequency range $\omega \in [1, 10^{10}]$. Recall that $s = 2\pi\imath\omega$.**

## 2.2 Butterfly Gyroscope

In inertial navigation, rotation sensors, also called gyroscopes, are indispensible. One such sensor, at the high-end of the quality range is the Butterfly gyroscope. You can see a diagram of this gyro in figure 2 By applying DC-

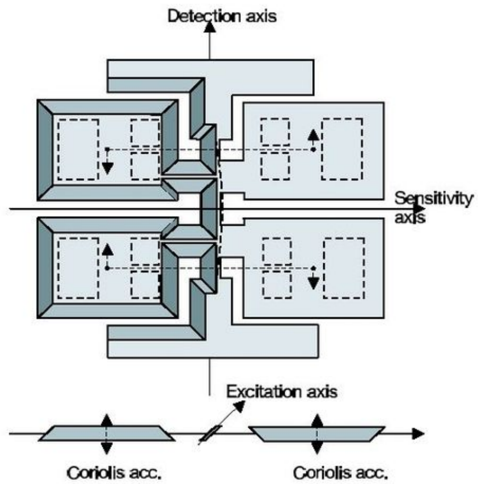


Figure 2: Butterfly Gyroscope

biased AC-voltages, the Butterfly wings are forced to vibrate in anti-phase in the wafer plane. This vibrational mode is called the excitation mode. As the structure containing the gyro rotates about the axis of sensitivity (see Figure 2), each of the masses will be affected by a Coriolis acceleration. The Coriolis force induces an anti-phase motion of the wings out of the wafer plane, which can be measured via different electrodes. The external angular velocity can be related to the amplitude of this vibrational mode. Since this gyro must be fast and precise, it should be clear to you that a high-precision, but efficiently computable structural model of the gyro must be implemented.
The Gyro is discretized using FEM, resulting in a model of the form

$$\begin{aligned} M\ddot{\mathbf{x}} + \beta K\dot{\mathbf{x}} + K\mathbf{x} &= B \cdot u(t). \\ y(t) &= C^T\mathbf{x}(t) \end{aligned} \tag{13}$$

**Note that multiple input and output states are considered! You do not have to report on all input-output pairs, but your methods**

**should work for each of them! You have been given a MatLab file containing all the necessary data for this system. Take $\beta = 1e - 6$ Analyze the system in detail and report your findings. Use the function `bode_from_system` or `bode_from_function` to evaluate and plot transfer function in the frequency range $\omega \in [1, 10000]$. Recall that $s = 2\pi\imath\omega$.**

## 2.3 International space station component

Control is another important source of model order reduction, since the complexity of a controller is dominated by the complexity of the corresponding system. In the International Space Station (ISS), assembly, testing and maneuvring is done in 'stages' i.e. parts of the ISS. One such stage, stage 12A, is depicted in figure 2.3. This stage leads to a large sparse system to be re-

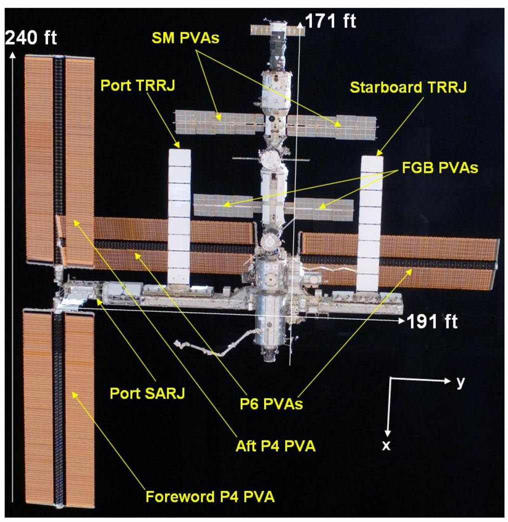


Figure 3: Stage 12A of the ISS. Source: *First Ever Flight Demonstration of Zero Propellant Maneuver(TM) Attitude Control Concept*, Bedrossian, N. & Bhatt, S.,2007

duced in order. This system was determined using techniques from standard systems theory, and is of the form

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= B \\ \mathbf{y} &= C^T\mathbf{x} \end{aligned}$$

**Note that multiple input and output states are considered! You do not have to report on all input-output pairs, but your methods should work for each of them!** Due to the large size of stage 12A and the absense of oscillatory air forces in space, we are mainly interested in the low-frequency behavior. **You have been given a MatLab file containing all the necessary data for this system. Analyze the system in detail and report your findings. Use the function `bode_from_system` or `bode_from_function` to evaluate and plot transfer function in the frequency range $\omega \in [10^{-2}, 10^{2}]$. Recall that $s = 2\pi\imath\omega$.**

# 3 Approximations of the transfer function

Now that we know our models, we can start reducing them. This means approximating the actual transfer function by some approximant $\hat{H}$. In this section you are asked to construct such an $\hat{H}$ in three of ways. Keep the following questions in mind: in what way is the transfer function approximated? How can I evaluate this approximation? How do we measure its accuracy? Analyze thoroughly!
**Compare and contrast the properties of these approximizations, the (typical) quality of the approximation reached, the overall performance,...**

## 3.1 Dominant pole methods

In class you have seen modal truncation as a method for model order reduction. One example of this is the *Dominant Pole Algorithm* (DPA). The general form of DPA is given in algorithm 1.
**Explain DPA and its relation to Newton's method. Implement it in `Matlab`**
**Simple DPA is not a very good or useful algorithm. Explain why. Use this to motivate *subspace accelerated DPA* (SADPA) as you have seen it in the slides.** You have been provided an algorithm that implements SADPA. This is an extremely complicated algorithm. You do not need to understand the full algorithm, but analyse the settings section of the preamble carefully. For convenience, you can set `options.use_lu=0` and `opt.dpa_bordered = 0` to start. However the other options require more

**Algorithm 1:** DPA

**input :** System $(E, A, b, c)$, initial pole estimate $s_0$ , tolerance $\epsilon$
**output:** Approximate dominant pole $\hat{\lambda}$ and corresponding eigenpair $(\mathbf{x}, \mathbf{y})$

**init** $k := 0$, $err = \infty$
**while** $err > tol$ **do**
  Solve $(s_k E - A)\mathbf{v}_k = \mathbf{b}$
  Solve $(s_k E - A)^* \mathbf{w}_k = \mathbf{c}$
  Compute the new pole estimate
  $s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k} = \frac{\mathbf{w}_k^* A \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}$
  $\mathbf{x} := \mathbf{v}_k / \|\mathbf{v}_k\|$
  $\mathbf{y} := \mathbf{w}_k / \|\mathbf{w}_k\|$
  err $:= \|A\mathbf{x} - s_{k+1} E\mathbf{x}\|_2$
  $k := k + 1$
**end**

careful understanding. Play around a bit and pick heuristically good choices. **You do not have to provide a proof of optimality, but justify your choices.**
We can also write a version of SADPA specifically tailored to the quadratic model of a mechanical system. For this we use lemma 1.

**Lemma 1** *Suppose* $(s^2 M + sD + K) := G(s)$ *is invertible. Then*

$$\begin{aligned}\frac{d}{ds} G^{-1}(s) &= -G^{-1}(s) G'(s) G^{-1}(s) \\ &= -(s^2 M + sD + K)^{-1}(2sM + D)(s^2 M + sD + K)^{-1}\end{aligned}$$

**Prove this lemma. Use it to construct a Newton method for a quadratic model like in the case of (regular) DPA. Call this algorithm QDPA and write down what this algorithm looks like. Implement this in `Matlab`**
You have been provided code that implements the subspace accelerated DPA specifically for quadratic systems, called SAQDPA. Again, this is an extremely complicated algorithm. You do not need to understand the full algorithm, but similarly to the descriptor case, analyse the settings section of the preamble carefully. Play around a bit and pick heuristically good choices.

**You do not have to provide a proof of optimality, but justify your choices.**
**Apply the dominant pole based algorithms to the three models. What works best for the quadratic model?**

## 3.2 Iterative Rational Krylov

Suppose we have a system of the form

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{14}$$

with associated transfer function

$$H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i}.$$

Theoretically, the Iterative Rational Krylov Algorithm (IRKA) aims at minimizing

$$\|H - \hat{H}\|_{\mathcal{H}_2} := \left( \int_{\infty}^{\infty} |H(is) - \hat{H}(is)| ds \right)^{\frac{1}{2}}$$

over the set $\mathcal{V}_d$ of approximants

$$\hat{H}(s) = \sum_{i=1}^{d} \frac{\hat{R}_i}{s - \hat{\lambda}_i}$$

of fixed Macmillan degree $d$. This lemma is useful:

**Lemma 2** *If*

$$G(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i}$$

*then it holds that*

$$\|G\|_{\mathcal{H}_2}^2 = \sum_{i=1}^{n} R_i G(-\lambda_i)$$

*under mild technical assumptions that you can assume satisfied.*

Now define the error function

$$\mathcal{E}(s) := \|H(s) - \hat{H}(s)\|^2_{\mathcal{H}_2}$$

and observe that in the context of minimizing $\mathcal{E}$ over the set of approximate models $\hat{H}$ of fixed Macmillan degree $d$, $\mathcal{E}$ is a function of $2d$ variables. **Explain this. Which variables are this specifically? Use this together with the lemma to show that a necessary condition for $\mathcal{E}$ to be minimized over the set of models $\mathcal{V}_d$ is that**

$$\begin{aligned} H(-\hat{\lambda}_i) &= \hat{H}(-\hat{\lambda}_i) \\ H'(-\hat{\lambda}_i) &= \hat{H}'(-\hat{\lambda}_i) \end{aligned}$$

**In the context of Krylov methods for model order reduction, what kind of scheme do these conditions make IRKA?**
**Is it possible to directly make this above condition be satisfied? Why (not)?**
**Use the above reasoning and definitions to explain the basic mechanism behind IRKA, given in algorithm 2 (Observe that this version of IRKA is specifically for descriptor systems)**

---

**Algorithm 2:** IRKA

---

**input :** System $(E, A, b, c)$, init. pole estimates $\{\sigma_i\}_{i=1}^k$ , tolerance $\epsilon$
**output:** Approximate model $(\hat{E}, \hat{A}, \hat{b}, \hat{c})$
**while** *norm of $\sigma$-update $>$ tol* **do**
  $V = \mathrm{orth}\{\mathbf{x}(\sigma_1), \dots, \mathbf{x}(\sigma_k)\}$
  $W = \mathrm{orth}\{\mathbf{z}(\sigma_1), \dots, \mathbf{z}(\sigma_k)\}$
  $\hat{E} = W^T E V, \hat{A} = W^T A V, \hat{\mathbf{b}} = W^T \mathbf{b}, \hat{\mathbf{c}} = V^T \mathbf{c}$
  calculate generalized eigenvalues $\{\hat{\lambda}_1, \dots, \hat{\lambda}_k\}$ of $(\hat{A}, \hat{E})$
  set $\sigma_i := -\hat{\lambda}_i$ for all $i = 1, \dots, k$
**end**
$V = \mathrm{orth}\{\mathbf{x}(\sigma_1), \dots, \mathbf{x}(\sigma_k)\}$
$W = \mathrm{orth}\{\mathbf{z}(\sigma_1), \dots, \mathbf{z}(\sigma_k)\}$
$\hat{E} = W^T E V, \hat{A} = W^T A V, \hat{\mathbf{b}} = W^T \mathbf{b}, \hat{\mathbf{c}} = V^T \mathbf{c}$

---

**You have been provided code that implements IRKA for descriptor systems. Adapt this code into a new function that works for**

**quadratic models. What options do you have in case of quadratic models? Which one seems the best? Justify your choice. Apply IRKA to the three models. Compare IRKA to the DPA based approximation schemes.**

## 3.3 Greedy Rational Krylov

In the slides on model order reduction you can also find the Greedy Rational Krylov method. **Explain the principles of Greedy rational Krylov. You have been provided a version of greedy rational Krylov for descriptor sytems. Again, as for IRKA, adapt this code into new code for quadratic models. Test this greedy algorithm on the three systems provided and report your results. Compare and contrast as before. Interpret your findings carefully.**

## 3.4 Quotation and questions

Points are awarded based on the correctness of your results and the quality of your code (40%), the insight demonstrated in your report (40%) and the organization and presentation of your report (20%). Don't hesitate to email me at simon.dirckx@kuleuven.be if anything is unclear or if you suspect something is wrong. If you are stuck on something you can always ask for a hint. This will be taken into account in the evaluation but will not severely impact your grade.