Homework: Model order reduction
Clémence Duplat r0977801

January 7, 2024

1 Theory

In this document 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 where x represents the state vector, A the system matrux, b the input matrix:

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

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 (transform time-domain function into a complex frequency domain function to analyze the system) 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.
We will transform equation 3 which is a second order ODE (use of $\ddot{\mathbf{x}}$) into a first order ODE using a new state vector:

$$q(t) = [\mathbf{x}, \dot{\mathbf{x}}]^T \in \mathbb{R}^{2n}$$

(because combination of 2 states $\in \mathbb{R}^n$). The derivative of this new state vector is given by:

$$\dot{\mathbf{q}} = [\dot{\mathbf{x}}, \ddot{\mathbf{x}}]^T$$

and so, assuming M is invertible, we can rewrite the equation 3 as follows:

$$\ddot{\mathbf{x}} = M^{-1}(-D\dot{\mathbf{x}} - K\mathbf{x} + \mathbf{b} \cdot u(t))$$

If we take back the derivative of our new state we have :

$$\dot{\mathbf{q}} = [\dot{\mathbf{x}}, M^{-1}(-D\dot{\mathbf{x}} - K\mathbf{x} + \mathbf{b} \cdot u(t))]^T$$

which can be rewritten as follows:

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

where $A = \begin{bmatrix} 0_{nxn} & I_{nxn} \\ -M^{-1}K & -M^{-1}D \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} 0_{nxn} \\ M^{-1}\mathbf{b} \end{bmatrix}$

We can apply the Laplace transform to this new state, by knowing that the Laplace transform of the first derivative is $sQ(s) - q(0)$:

$$sQ(s) = AQ(s) + F(s)U(s)$$

by assuming that $q(0) = 0$. We can arrange it as follows

$$(sI - A)Q(s) = F(s)U(s)$$

If we use the abuse of notation , we find back:

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

Note that after the two transformations introduced before, these variable are linked, but not at all identical to their original counterparts.
We can generalize it, and instead looking 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. This will help us to model a larger class of systems, in particular when the mass matrix M is singular or non-invertible. This will lead to a more realistic and accurate representation of the system's behavior. Our generalized problem is formulated as follow:

$$E\dot{\mathbf{q}} = A\mathbf{q} + \mathbf{f} \cdot u(t) \tag{7}$$

where $A = \begin{bmatrix} 0_{nxn} & I_{nxn} \\ -K & -D \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} 0_{nxn} \\ \mathbf{b} \end{bmatrix}$ and $E = \begin{bmatrix} I_{nxn} & 0_{nxn} \\ 0_{nxn} & M \end{bmatrix}$

Typically, we are not interested in all of x, but only in some (scalar) function y of $\mathbf{x}$. This is written in two parts, the state equation (describes how the state of the system evolves over time in response to the input u) and the output equation(describes how the output is generated from the current state):

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{8}$$

Equation 19 is thus referred to as the *state space* model of the mechanical system.

We can demonstrate that under the assumptions made earlier about this model, it is equivalent to the model

$$\begin{aligned} s^2M\mathbf{x} + sD\mathbf{x} + K\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{9}$$

with $x \in \mathbf{R}^n$.
Indeed, by applying Laplace transformation on 4, we get:

$$[sx, s^2x]^T = A[x, s^2x]^T + \mathbf{f} \cdot u(t) \tag{10}$$

and we see that we retrieve on this equation. The output remains the same and show us how it is related to the state. 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* $H(s) := y(s)/u(s)$. Indeed, the transfer function represent the relationship between the input and the output of our LTI system in the frequency response and thus will explain us "how the output of the system responds to an input in terms of frequency". Thanks to this function we will be able to analyze the stability of the system (via the poles), the frequency response and the dynamic properties. If we take our model 9, we can derive the following transfer function H(s) by seeing that the state equation can be reformulated as $x = (sI - A)^{-1}bu$ and if we replace that in the output equation we find $y = c^T((sI - A)^{-1}bu)$. Thus we find that:

$$H(s) := y(s)/u(s) = c^T((sI - A)^{-1}b$$

If we take the eigendecomposition of A given by $A = P\Lambda Q^*$ and assume it exists (as well as that $P = [\mathbf{p}_1 \cdots \mathbf{p}_n]$, $Q = [\mathbf{q}_1 \cdots \mathbf{q}_n]$ and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$), we can express the transfer function as follows:

$$H(s) := c^T((sI - P\Lambda Q^*)^{-1}b$$

We would like to reexpress the matrix such that it is diagonal because calculating the inverse of a diagonal matrix is easy: we would just need to to invert each diagonal element.
Therefore, by assuming that $Q^*P = I$ (biorthogonality):

$$(sI - A) = (sPQ^* - P\Lambda Q^*) = Q(sI - \Lambda)P^*$$

Now it is easy to see that :

$$(sI - \Lambda)^{-1} = diag(\frac{1}{s-\lambda_1}, ..., \frac{1}{s-\lambda_n})$$

And thus, we know that the inverse of a matrix can be expressed as follows:

$$(sI - A)^{-1} = P(sI - \Lambda)^{-1}Q^* = \sum_{i=1}^{n} \frac{\mathbf{p}_i \mathbf{q}_i^*}{s - \lambda_i}. \tag{11}$$

If we substitute that in H(s):

$$H(s) := c^T(\sum_{i=1}^{n} \frac{\mathbf{p}_i \mathbf{q}_i^*}{s - \lambda_i} b)$$

Finally, we can express it as follow:

$$\begin{aligned} H(s) &= \mathbf{c}^T(sI - A)^{-1}\mathbf{b} \\ &= \sum_{i=1}^{n} \frac{(\mathbf{c}^T\mathbf{p}_i)(\mathbf{q}_i^*\mathbf{b})}{s - \lambda_i} \end{aligned} \tag{12}$$

The second form of equation 12 we will call the *residual form* of the transfer function where λ_i is the pole and $R_i = (c^T p_i)(q_i^* b)$ the associated residue. We can express H in terms of the quadratic model, described above, by expressing that state equation as follows: $x = (s^2M + sD + K)^{-1}bu$ and replace it in the output equation $y = c^T(s^2M + sD + K)^{-1}bu$. Finally the transfer function can be expressed as:

$$H(s) = c^T(s^2M + sD + K)^{-1}b$$

We see that we expressed H(s) in terms of his eigenvalues and eigenvectors of the system matrix A. Expressing it in this form, is a huge advantage because we can see how each mode, describes by the eigenvectors, contribute on the behaviour of the system. Because it is directly related to the eigenvalues, it will be easy to analyze the frequency response as well. Thanks to that, we can use it for model order reduction by analyzing which modes affects the most the system and thus which modes it is better to keep or not in order to still preserve the essential behaviour of the system.

1.3 State space models from simple ODEs

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

$$\begin{aligned} E\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + \mathbf{b} \cdot u(t) \\ y(t) &= \mathbf{c}^T\mathbf{x}(t), \end{aligned} \tag{13}$$

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

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

Note again the abuse in notation, and that in general E is not the identity. If we take the same steps as before, we can find the follow transfer function:

$$H(s) = c^T(sE - A)^{-1}b$$

It can still be written in some residual form as in equation 12 but under certain assumptions. To find those assumptions, we will look into the generalized eigenvalue decomposition of (A,E) defined as $A = EP\Lambda Q^*$. This time, we will have to make the assumption that $Q^*EP = I$. We will make the same steps as above:

$$(sE - A) = (sEPP^{-1} - EP\Lambda P^{-1}) = EP(sI - \Lambda)P^{-1}$$

and thus

$$(sE - A)^{-1} = P(sI - \Lambda)^{-1}P^{-1}E^{-1} = \sum_{i=1}^{n} \frac{\mathbf{p}_i\mathbf{q}_i^*}{s - \lambda_i}. \tag{15}$$

We find the same residual form. Note that E must be non-singular to be able to represent the transfer function in a residual form.

2 Applications

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 40μm wide, 15μm thick, with a separation of 40μm. The spiral is suspended 55μm 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 0.1μm plane of copper is added 45μm above the copper spiral. The overall extent of the suspended turns is 1.58mm $\times$ 1.58mm.
The model is discretized using a 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 representation**:

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

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. The **Laplace domain state space model** can be obtained as follow

$$LsI_m(s) = RI_m(s) + Nv_p(s)$$

and by arranging the terms we obtain:

$$I_m(s) = (LsI - R)^{-1}Nv_p(s)$$

$$i_p(s) = N^T I_m(s)$$

The transfer function is thus defined as

$$H(s) = N^T(LsI - R)^{-1}N$$

This Laplace domain transformation will provide a robust method for analyzing the electrical behavior of the spiral inductor. 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: $\omega \in [1, 10^{10}]$.

We can plot the transfer function in the frequency range $\omega \in [1, 10^{10}]$ using the matlab function bode_from_system that computes the frequency response of a linear system with transfer function $H(s) = c' * (sE - A)^{-1} * b$.

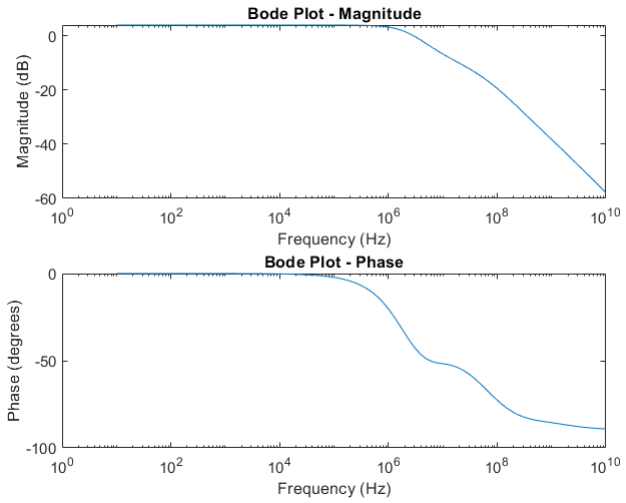

Figure 1: PEEC bode plot

This bode plot explains us the behavior of our system in the frequency domain. We can start by analyzing the Magnitude Plot of our system, to analyse the order of our system. At lower frequencies, the system has a consistent gain a little bit higher than 0dB. Between the frequencies of 10^6 and 10^8 the magnitude start to decrease to a value of -20dB. The magnitude then start to decrease at a different rate between 10^8 and 10^{10}. The slope of those decreases can help us determine the order of the system.
On the phase plot, we see that the begins to 0, indicating that there is no phase shift at lower frequencies. Then around 10^6 we see a first phase drop. This indicates a delay of the response to the input signal. We see that around

10^8 there is a second phase drop. We see that our system act as a low-pass filter.

We see that it indeed correspond to capturing the skin effect while also maintaining an accurate low frequency response. This knowledge is important to keep in mind when we will proceed the model-order reduction.

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. By applying DC-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, 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

$$\begin{aligned} M\ddot{\mathbf{x}} + \beta K\dot{\mathbf{x}} + K\mathbf{x} &= B \cdot u(t). \\ y(t) &= C^T\mathbf{x}(t) \end{aligned} \tag{16}$$

We can obtain the Laplace state space model:

$$\begin{aligned} Ms^2X(s) + \beta KsX(s) + KX(s) &= BU(s). \\ Y(s) &= C^TX(s) \end{aligned} \tag{17}$$

We can thus obtain the following transfer function:

$$H(s) = \frac{c^TB}{Ms^2 + \beta Ks + K} \tag{18}$$

Using the Laplace transformation will simplify the analysis of the system's behavior in the frequency domain. Note that we are in a MIMO system

where we have one single input but 12 outputs. We are in a quadratic model. Therefore, we will plot the bode diagram of input related to a corresponding output by using the matlab function bode_from_function (note we have to take the transpose this result such that it match with our transfer function). We can analyse the system response of the first output when subjected to a single input on this Bode Diagram in the frequency range $\omega \in [1, 10000]$.

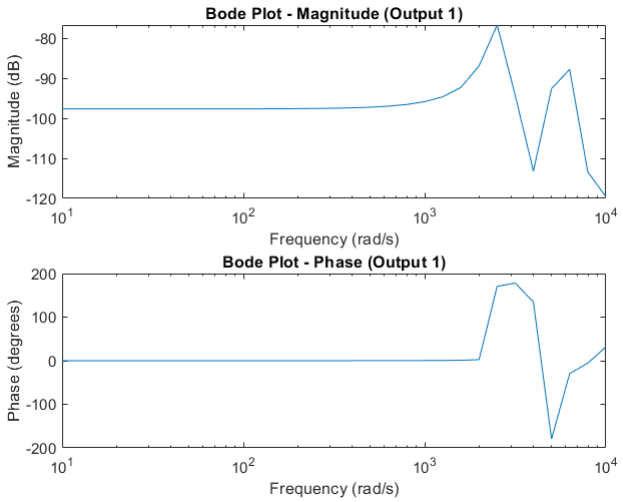

Figure 2: Gyroscope Bode Plot

On the magnitude plot, we see that for small frequencies the gain remains constant at -100dB. Then, we clearly see peaks indicating resonant frequencies. At those frequencies the system output will be very sensitive to the input. Every time that a peak occurs, we see a sudden decrease. This indicates the effect of damping that become more significant. Indeed $\beta = 1e-6$ the damping factor is very small. It is thus negligible at lower frequencies but become significant at higher frequencies and is thus able to damp the peaks.
On the phase plot we see that the there is initially no phase shift between the input and the output. At the frequencies where a peak occurs on the magnitude plot, we see a phase shift on the phase plot.
The presence of multiple peaks indicates that the Butterfly Gyroscope system has multiple modes of vibration, which seems logic based on the description of the system.

Here we analysed the response of one single output but each output can have different peaks.

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 maneuvering is done in 'stages' i.e. parts of the ISS. This stage leads to a large sparse system to be reduced 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 (MIMO system)! We can find the transfer function by arranging the terms as follow:

$$\begin{aligned} (sI - A)\mathbf{x} &= B \\ \mathbf{y} &= C^T(sI - A)^{-1}B \end{aligned}$$

The transfer function is thus given by:

$$H(s) = C^T(sI - A)^{-1}B$$

Each element of the matrix H(s) is a transfer function $H_{ij}(s)$ that relates the j-th input to the i-th output.
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. We will thus analyse the system response of the first output when subjected to the first input on this Bode Diagram in the frequency range of $\omega \in [10^{-2}, 10^2]$. We see the system is not quadratic so we will use the matlab function bode_from_system that computes the frequency response of a linear system with transfer function $H(s) = c' * (sE - A)^{-1} * b$.

On the magnitude plot, we see that our DC gain starts at -100 dB. This indicates a low sensitivity to steady-state input. We also see multiple clear

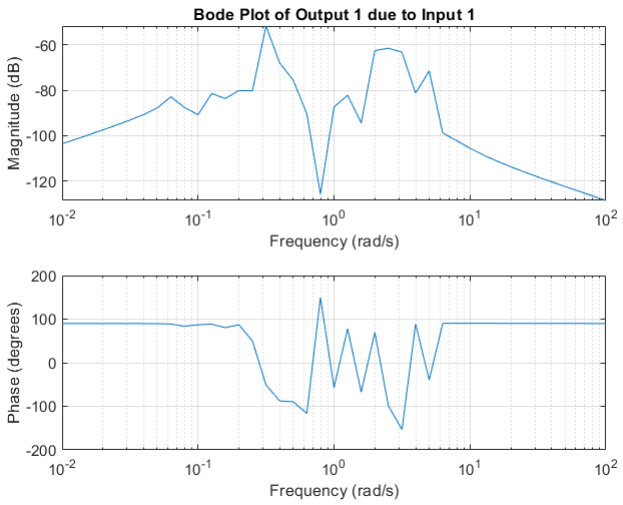

Figure 3: ISS Bode Plot

peaks indicating the resonant frequencies. The system output is significantly amplified in repsonde to inputs at those frequencies. We also see that those peaks are directly damped (important for the stability).
On the phase plot, we see that the phase shift of 90° indicating that the input and the output are in phase. Then, we see that the phase oscillates between 100 and -100. This indicate the presence of multiple mode of vibrations.

We see that we have a complex system representation. In addition, the peaks are significant for the ISS component and this has to be preserved when we will apply model order reduction.

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 we will construct such an $\hat{H}$ in three of ways.

3.1 Dominant pole methods

Modal truncation is 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.

Algorithm 1: DPA

input : System (E, A, b, c), initial pole estimate s_0 , tolerance ϵ
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

The DPA will approximate a high-order system by a low-order system by identifying the dominant poles (explain the most significant dynamic of the system). The DPA algorithm will thus find the dominant poles of the transfer function $H(s) = c^T (sI - A)^{-1} b$. We will then be able to construct a new transfer function $H^(s)$ with those poles.

This is closely related to the Newton's method that iteratively find the roots of a function. We know that the poles of H(s) are the zeros of $Q(s) = H^{-1}(s)$. Thus if we use the Newton method to find the zeros of Q(s), it will be the

same as using the DPA method for finding the poles of H(s). The Newton-method is given by:

$$s_{k+1} = s_k - \frac{Q(s_k)}{Q'(s_k)} = s_k + \frac{H(s_k)}{H'(s_k)}$$

and has been adapted for the DPA.
The update setp given in our DPA algorithm is:

$$s_{k+1} = \frac{\mathbf{w}_k^* A \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}$$

where v_k and w_k are the solutions of the equations $(s_k E - A)\mathbf{v}_k = \mathbf{b}$ and $(s_k E - A)^* \mathbf{w}_k = \mathbf{c}$. They thus represent the right and left eigenvectors of the system for our current pole estimate s_k (when s tends to λ_1).
The convergence of the DPA is not guaranteed and even if it converges, DPA only finds one pole. To increase our chances to compute all the dominant poles, we will compute more than one pole. Therefore, we will introduce the Subspace accelerated DPS (SADPA) that will improve the DPA by combining it with subspace projection methods. It will thus explore subspaces that capture more of the system's dynamic behavior and improve the probability of finding the true dominant poles. This will increase the convergence chances of the algorithm.

The settings section of the preamble of the SADPA algorithm have to be chosen carefully in order to identify the dominant poles of our system with accuracy while having a computational efficient algorithm. The a priori knowledge of our Bode plots will help us decide for the optimal settings. The settings can be tuned if the convergence is not direclty met.

- **nwanted** indicate the number of dominant poles we want to compute. This can correspond to the number of resonant frequencies on our magnitude plot.
- **tol**: a classic tolerance of $1e-10$ like indicated or less for more sensitive problems.
- **strategy** will be choosen based on the expected behavior of our system. If we have clear peaks on the magnitude plot, we can choose LM.

- **kmin** and **kmax** will define the minimal and maximal search-space dimension. In other words the number minimal/maximal of pole we think is necessary to capture the essential dynamic of the system. This will be chosen based on the complexity of our system.
- **maxrestarts** indicate the maximum number of restarts allowed. A high number can improve our chances to find all the dominant poles.
- **Function Handles** for MV operations and solves: f_ax,f_ex, f_semax, f_semax_s
- **use_lu=0** and **dpa_bordered = 0** to start
- **Advanced options** those are there if you want to adapt this algorithm for large scale problems.

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

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

Proof:
We just need to replace the following terms in the equation $G'(s) = \frac{d}{ds}(s^2M + sD + K) = 2sM + D$ *and* $G^{-1}(s) = (s^2M + sD + K)^{-1}$.

This lemma explains us how the inverse of the system matrix $G(s)$ changes with respect to s. We can use it to construct the update step for a Newton method of a quadratic model (with a second-order transfer function). This is called the Quadratic DPA (QDPA) and is a special case of DPA. The transfer function for the quadratic model is given by $H(s) = c^T G(s)^{-1}b$ and the derivative is thus $H'(s) = -c^T(-G^{-1}(s)G'(s)G^{-1}(s))b$ The update step will then be as follow:

$$s_{k+1} = s_k + \frac{H(s_k)}{H'(s_k)}$$

$$s_{k+1} = s_k - \frac{c^T G(s)^{-1}b}{-c^T(-G^{-1}(s)G'(s)G^{-1}(s))b}$$

Algorithm 2: QDPA

input : System (M, D, K, b, c), initial pole estimate s_0, tolerance ε
output: Approximate dominant pole λ and corresponding eigenpair (x, y)

init $k := 0$, $err = \infty$
while $err > tol$ **do**
  Solve $(s_k^2 M + s_k D + K)v_k = b$
  Solve $(s_k^2 M + s_k D + K)^* w_k = c$
  Compute the new pole estimate
  $s_{k+1} = s_k - \frac{c^* v_k}{w_k^*(2s_k M + D)v_k} = \frac{w_k^* K v_k}{w_k^*(2s_k M + D)v_k}$
  $x := \frac{v_k}{\|v_k\|}$
  $y := \frac{w_k}{\|w_k\|}$
  err $:= \|Kx - s_{k+1}(2s_k M + D)x\|_2$
  $k := k + 1$
end

The algorithm of QDPA looks as follow
We can implement the subspace accelerated DPA specifically for quadratic systems, and this is called SAQDPA. The settings section of the preamble of the SAQDPA algorithm have to be chosen carefully in order to identify the dominant poles of our system with accuracy while having a computational efficient algorithm. The setting are quiet similar to those for SADPA. Here the main difference is that we have a second-order system that will add complexity to the pole search process, notamely the fact that we will have complex conjugate pairs of poles. SAQDPA also includes one additional function handle for the matrix M. So, when we tune the settings, we need to take into account the quadratic term.
We can apply the dominant pole based algorithms to the three models. Note that every time I used the theory from section 1 to reconstruct the reduced transfer functions. In general we will principally look if we were able to keep the important features described in the Magnitude Bode plot.

3.1.1 Spiral Inductor Model

The Spiral Inductor Model is a first-order system. Therefore, we will apply the DPA and the SADPA to this system to see which is the most efficient. By applying DPA, we only catch the influence of only one dominant pole. We see that it almost correspond to the true frequency response but that the initial gain is not good.

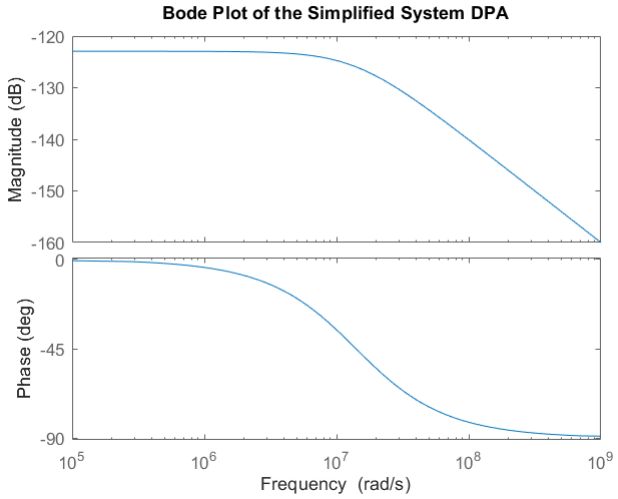

Figure 4: DPA for 1 dominant pole

By applying SADPA with the right options I obtain this Bode plot. I see that I have 2 visible peaks, so I will start by considering 4 dominant poles (because complex conjugate).

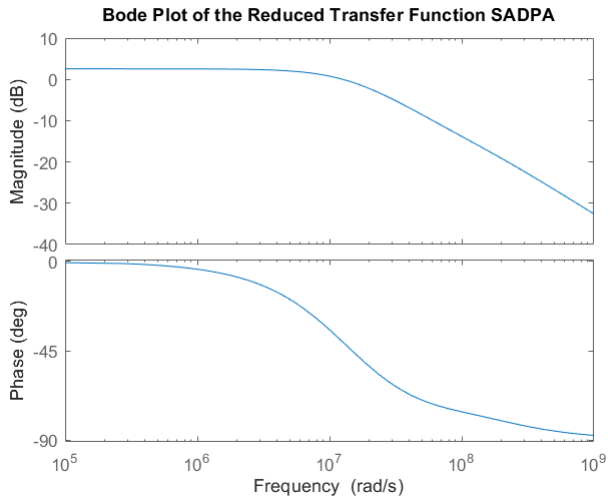

Figure 5: SADPA for 4 dominant poles

3.1.2 Butterfly Gyroscope

The Butterfly Gyroscope model is a second-order system wich make it a quadratic problem. Therefore, we will apply the QDPA and the SAQDPA to this system to see which is the most efficient.
However, we see that we have more than one dominant pole so applying QDPA doesn't really have a sense.
When I tried to apply SAQDPA, this is the only result I was able to obtain even after trying to change the options for hours and hours.

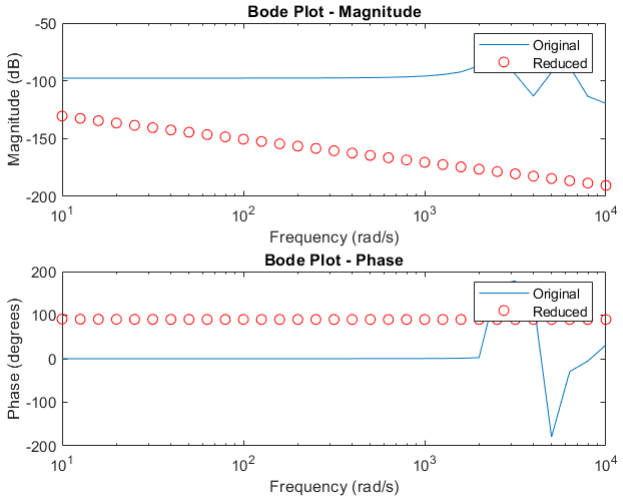

Figure 6: SADPA for PEEC

3.1.3 ISS component

The ISS component model is a MIMO first order system. Therefore, we will apply the DPA and the SADPA to this system to see which is the most efficient.
However, applying DPA here has not so much sens because we have a complex system which dynamic is represented by way more than one dominant poles. In addition, the algorithm will not converge. This shows the principal drawback of DPA.
We see that we have at least 10 important peaks thus I can start with 20 dominant poles (because complex conjugate) for SADPA we obtain this plot that is noy a good estimate of the dominant peaks.

When I increase the number of dominant poles I see that my model capture more of the system dynamic and that the important poles are captures. I think a choice of 50 dominant pole is sufficient to find a good balance between accuracy and efficiency.

We can even increase the number of dominant poles to 70 to see the result.

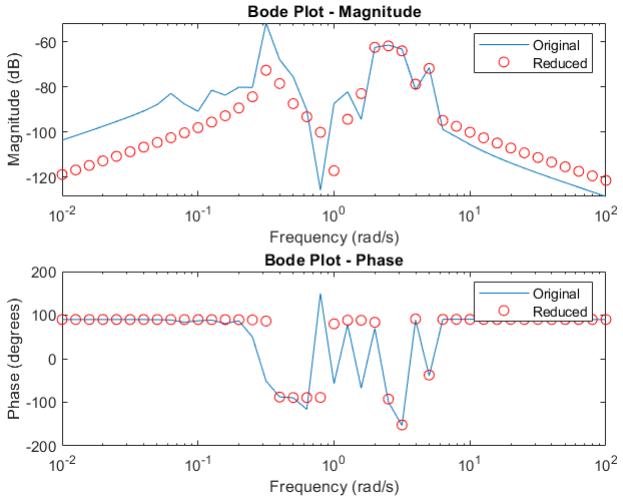

Figure 7: SADPA with 20 dominant poles

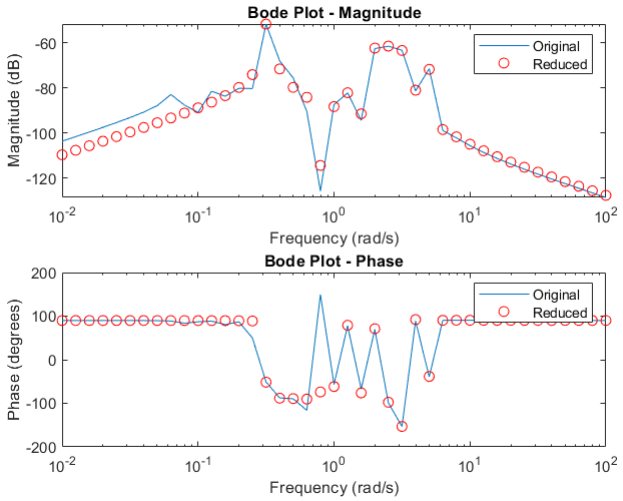

Figure 8: SADPA with 50 dominant poles

3.2 Iterative Rational Krylov

Suppose we have a system of the form

$$\begin{aligned} s\mathbf{x} - A\mathbf{x} &= \mathbf{b}u \\ y &= \mathbf{c}^T\mathbf{x}. \end{aligned} \tag{19}$$

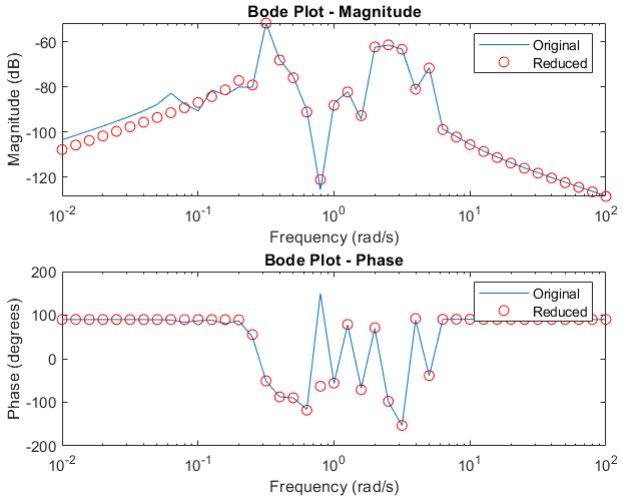

Figure 9: SADPA with 70 dominant poles

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

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

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.
Indeed, each term in the reduced-order model $\hat{H}(s)$ is characterized by d poles $\hat{\lambda}_i$ and d residues $\hat{R}_i$ and thus by 2d terms. The 2d variables will be adjusted during the IRKA in order to best approximate the behavior of the full-order system $H(s)$.

We can apply the Lemma 2 on $\mathcal{E}(s)$.

$$\mathcal{E}(s) = \sum_{i=1}^{n} R_i(H(-\lambda_i) - \hat{H}(-\lambda_i)) + = \sum_{i=1}^{d} \hat{R}_i(H(-\hat{\lambda}_i) - \hat{H}(-\hat{\lambda}_i))$$

To minimize it over the set of approximate model $\hat{H}$, we do it with respect to the poles and the residues of the reduced order model.

$$\frac{\partial \mathcal{E}}{\partial \hat{\lambda}_i} = 0 \quad \text{and} \quad \frac{\partial \mathcal{E}}{\partial \hat{R}_i} = 0$$

We can thus state the following Meier-Luenberger conditions for optimality:

$$H(-\hat{\lambda}_i) = \hat{H}(-\hat{\lambda}_i)$$
$$H'(-\hat{\lambda}_i) = \hat{H}'(-\hat{\lambda}_i)$$

Those conditions states that their values must match at the poles (moment matching) and that their derivative match at the poles (interpolation). This ensure that that the reduced-order model is the best approximation of the full-order model in the $\mathcal{H}_2$ norm sense. These matching conditions form the basis of the IRKA algorithm for updating iteratively the poles and residues of the reduced-order model.

We will try to construct the reduced-order transfer function that interpolate and tangentially matches the full-order transfer function at selected points

in the complex plane. This can be view as a rational Hermite interpolation, a generalization of Lagrange interpolation for rational functions.
Directly satisfying the above conditions is not possible due to the nonlinear and complex nature of the problem. Indeed, the poles λ_i and residues R_i appears in the transfer function in such a way that no explicit formula directly satisfy those conditions. In other words, we can say that the λ_i are not known a priori. Instead we will thus use an iterative process, IRKA, that will adjust the poles and residues of the reduced-order model and thus improve the approximation. It will use intermediate interpolation points until the required optimality criterion is met.

The algorithm will begin with an initial state space model (defined by the matrices) and an initial set of poles $\{\sigma_i\}_{i=1}^{k}$ (determine initial rational Krylov Subspace). To capture the essential dynamic behavior around the chosen poles, the algorithm will construct two orthogonal bases for the Krylov subspaces (V and W) associated with the original system. The vectors z and x are the solutions of the projected systems defined as $(\sigma_i E - A)^{-1}b$ and $(\sigma_i E - A)^{-T}c$. The system matrices A and E are then projected onto the subspaces spanned by the orthogonal bases V and W to form the reduced-order model matrices. This will reduce the dimensionality of our problem by keeping the main characteristics. We find new estimates for the poles by solving the generalized eigenvalues problem of the reduced-order model matrices. Note that we take the negative of the computed eigenvalue as new estimate to keep stability and accuracy. This algorithm will iteratively update the pole estimates until the change in two consecutive pole estimate falls below a certain tolerance. The final Krylov subspaces and the reduced-order model are then computed.

We can adapt this code into a new function that works for quadratic models by taking into account the quadratic term in the state-space representation. Multiple options exist in the case of the quadratic models. First, we can convert our quadratic model into a first order system but this will double the state-space dimension (because the state vector will be expanded). Then, we can truncate our quadratic system.
However, we will choose the second-order IRKA algorithm that will try to match the moments of second-order transfer function. Therefore, we will need to modify the Krylov subspace construction in order to take into account the mass matrix. We choose this option because it direct adress the

Algorithm 3: IRKA

input : System (E, A, b, c), init. pole estimates $\{\sigma_i\}_{i=1}^k$, tolerance ϵ
output: Approximate model $(\hat{E}, \hat{A}, \hat{b}, \hat{c})$

while *norm of* σ*-update* $>$ *tol* **do**
  $V = \text{orth}\{\mathbf{x}(\sigma_1), \ldots, \mathbf{x}(\sigma_k)\}$
  $W = \text{orth}\{\mathbf{z}(\sigma_1), \ldots, \mathbf{z}(\sigma_k)\}$
  $\hat{E} = W^T E V, \hat{A} = W^T A V, \hat{\mathbf{b}} = W^T \mathbf{b}, \hat{\mathbf{c}} = V^T \mathbf{c}$
  calculate generalized eigenvalues $\{\hat{\lambda}_1, \ldots, \hat{\lambda}_k\}$ of $(\hat{A}, \hat{E})$
  set $\sigma_i := -\hat{\lambda}_i$ for all $i = 1, \ldots, k$
end
$V = \text{orth}\{\mathbf{x}(\sigma_1), \ldots, \mathbf{x}(\sigma_k)\}$
$W = \text{orth}\{\mathbf{z}(\sigma_1), \ldots, \mathbf{z}(\sigma_k)\}$
$\hat{E} = W^T E V, \hat{A} = W^T A V, \hat{\mathbf{b}} = W^T \mathbf{b}, \hat{\mathbf{c}} = V^T \mathbf{c}$

second-order system leading to a more accurate performance. It has been implemented in matlab.

We can apply IRKA to the three models on the next pages.

3.2.1 Spiral Inductor

We can first apply the IRKA with one dominant pole. Then we can increase

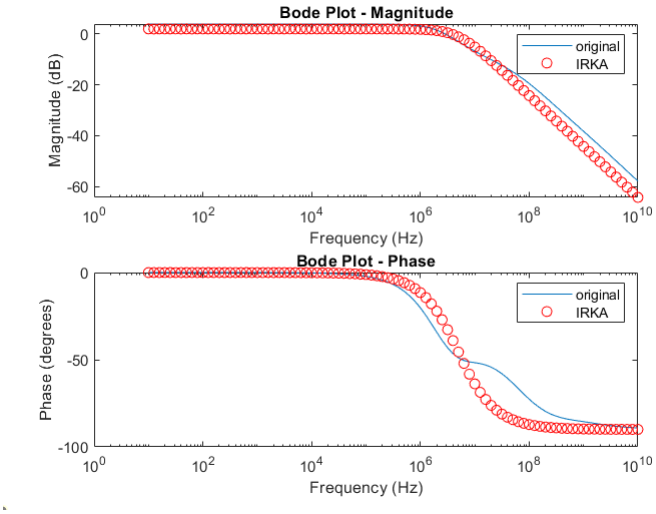

Figure 10: IRKA reduction k=1

it to 2 dominant poles. We see that the reduced system almost entirely capture the frequency response of our problem. This better capture the system transfer function than SADPA.

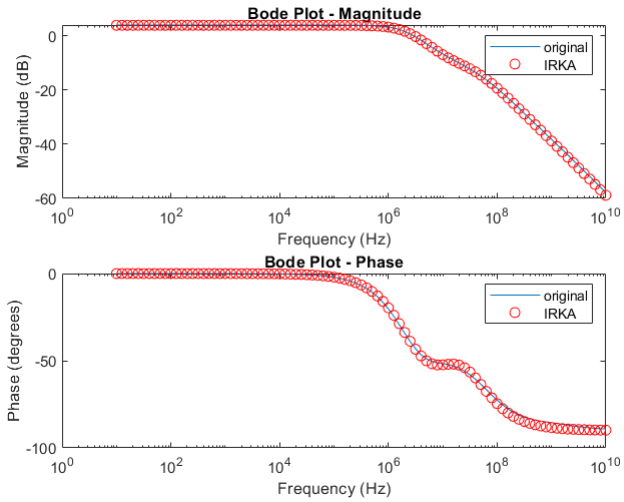

Figure 11: IRKA reduction k=2

3.2.2 Butterfly Gyroscope

Here I used QIRKA with 10 dominant poles and I have considerably a good approximation.

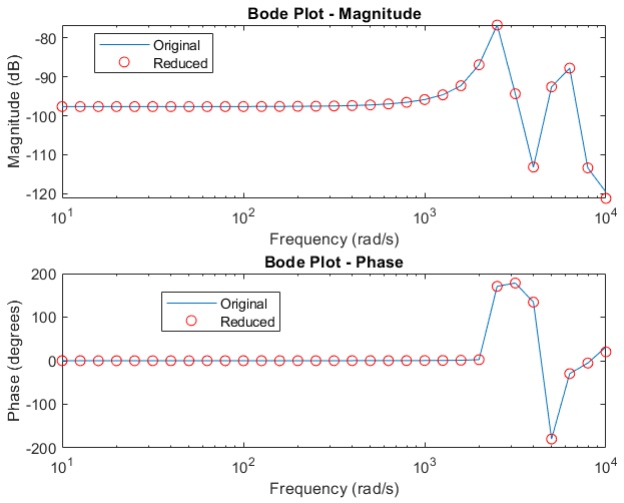

Figure 12: QIRKA reduction k=10

3.2.3 ISS component

I see that my results are less satisfying than those found with SADPA but that with k=10 I still catch some important peaks.

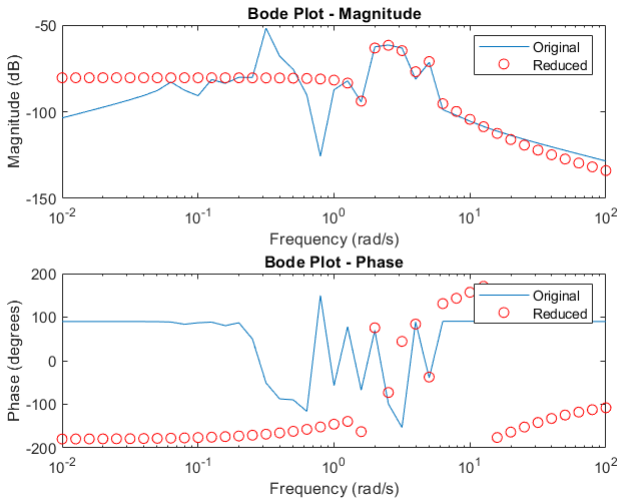

Figure 13: IRKA reduction k=5

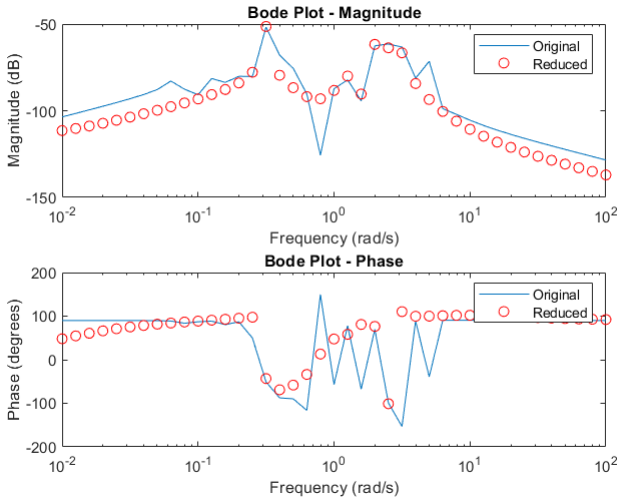

Figure 14: IRKA reduction k=10

3.3 Greedy Rational Krylov

The goal of this algorithm is to find a set of interpolation points $\sigma_1, ...\sigma_k$ such that the residual norm

$$||(sI - A)V\hat{x} - b||_2$$

is minimized with $V = [x(\sigma_1), ...x(\sigma_k)]$. This method is very efficient because it selects the most impactful points for model reduction. It thus captures the essential dynamics of the system with fewer computations.

Algorithm 4: Greedy Rational Krylov Algorithm

input : System (A, b), initial interpolation point σ_1, tolerance ϵ
output: Reduced order model matrices $(\hat{A}, \hat{b}, \hat{c})$

init *Set* $k = 1$ **while** *not converged with tolerance* ϵ **do**
  for $i = 1$ **to** k **do**
    Compute $x(\sigma_i) = (sI - A)^{-1}b$ at σ_i
  end
  $V = \text{orth}\{x(\sigma_1), \ldots, x(\sigma_k)\}$
  Build reduced model of order k: $\hat{A}_k, \hat{b}_k, \hat{c}_k$
  Compute residual norms for s on the imaginary axis:
  $\|(sI - A)V\hat{x}_k - b\|_2$
  Choose σ_{k+1} with largest residual norm
  $k = k + 1$
end

I could apply it for the PEEC model. The result is satisfying.

For my ISS model I obtained those results still less satisfying than SADPA.

Unfortunately I wasn't able to finish the homework due to a lack of time.

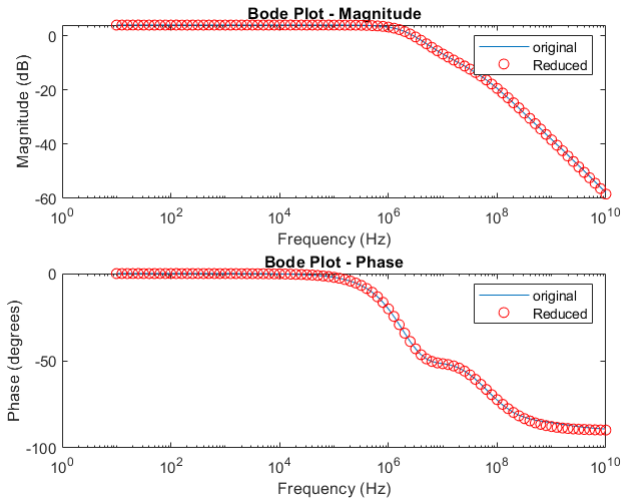

Figure 15: GRKA PEEC with scount = 2

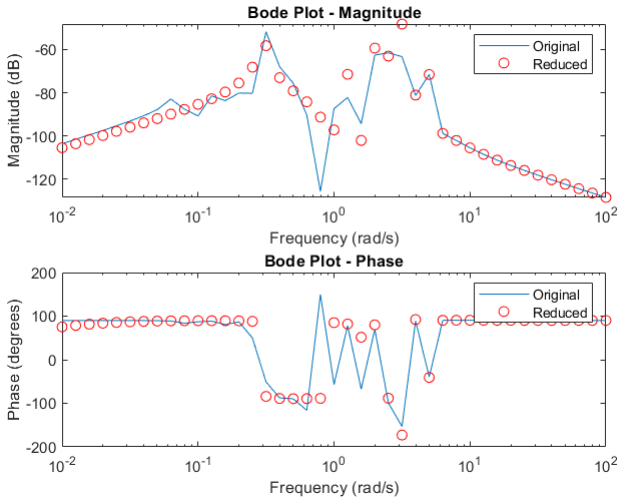

Figure 16: GRKA ISS with scount = 50

4 References

DPA:
https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.200700524
https://www.researchgate.net/publication/

IRKA:
https://arxiv.org/pdf/1911.05804.pdf