

Case study: Approximations of the Bessel Function

D.S. Karachalios^{*}, I.V. Gosea^{*}, Q. Zhang[†], A.C. Antoulas[†].

January 11, 2018

Abstract

The purpose of this note is to compare various approximation methods as applied to the inverse of the Bessel function of the first kind $\frac{1}{J_0(s)}$, in a given domain of the complex plane.

1 Introduction

The Bessel function of the first kind and order $n \in \mathbb{N}$, is defined as

$$J_n(s) = \frac{1}{2\pi i} \oint e^{(\frac{s}{2})(t-\frac{1}{t})} t^{-n-1} dt. \quad (1)$$

In the sequel we will consider only the case $n = 0$. Our goal is to approximate

$$H(s) = \frac{1}{J_0(s)}, s \in \mathbb{C}. \quad (2)$$

In particular we will be interested in the approximation of $H(s)$ in the rectangle

$$\Omega = [0, 10] \times [-1, 1] \subset \mathbb{C}, \quad (3)$$

of the complex plane. The magnitude of $H(s)$, in Ω , is as follows:

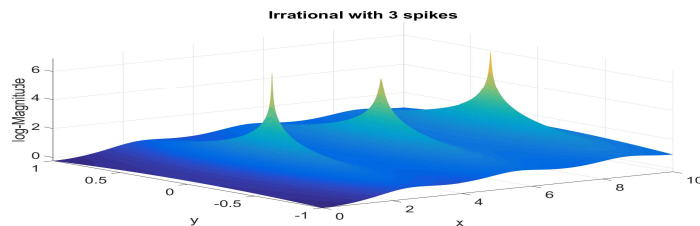


Figure 1: $\ln|H(s)|$, with $s \in \Omega = [0, 10] \times [-1, 1]$

^{*}Data-Driven System Reduction and Identification Group, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany e-mail: karachalios, gosea@mpi-magdeburg.mpg.de

[†]Department of Electrical and Computer Engineering, Rice University, Houston, e-mail: qz18, aca@rice.edu

The following interpolatory reduction methods will be used:

1. The Loewner Framework and (recursive) [1],
2. the AAA approach [8],
3. Vector Fitting [6, 3].

1.1 The Loewner Framework

The Loewner framework is a data-driven reduction method. It uses measured or computed data produced by an underlying dynamical system. The construction of reduced linear models is based on rank-revealing factorization. Here is a brief description of this method. For details, see [1]. Given is a set of pairs of complex numbers:

$$\{(s_k, \phi_k) : k = 1, \dots, N\},$$

with $s_k \in \mathbb{C}$, $\phi_k \in \mathbb{C}^{p \times m}$. We partition the data in two disjoint sets:

- left data: $(\mu_j, \ell_j, \mathbf{v}_j)$, $j = 1, \dots, q$
- right data: $(\lambda_i, \mathbf{r}_i, \mathbf{w}_i)$, $i = 1, \dots, k$

The objective is to find $\mathbf{H}(s) \in \mathbb{C}^{p \times m}$ such that:

$$\mathbf{H}(\lambda_i)\mathbf{r}_i \approx \mathbf{w}_i, \quad \ell_j^*\mathbf{H}(\mu_j) \approx \mathbf{v}_j^* \quad (4)$$

In this report, we consider approximation of a SISO (Single Input Single Output) system, i.e. $m = p = 1$, in which case the directions ℓ_j and \mathbf{r}_i can be chosen equal to 1. The associated Loewner and shifted Loewner matrices \mathbb{L} , \mathbb{L}_s , are defined as:

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 - \mathbf{w}_1}{\mu_1 - \lambda_1} & \dots & \frac{\mathbf{v}_1 - \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_q - \mathbf{w}_1}{\mu_q - \lambda_1} & \dots & \frac{\mathbf{v}_q - \mathbf{w}_m}{\mu_q - \lambda_k} \end{bmatrix}, \quad \mathbb{L}_s = \begin{bmatrix} \frac{\mu_1 \mathbf{v}_1 - \mathbf{w}_1 \lambda_1}{\mu_1 - \lambda_1} & \dots & \frac{\mu_1 \mathbf{v}_1 - \mathbf{w}_k \lambda_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q \mathbf{v}_q - \mathbf{w}_1 \lambda_1}{\mu_q - \lambda_1} & \dots & \frac{\mu_q \mathbf{v}_q - \mathbf{w}_k \lambda_k}{\mu_q - \lambda_k} \end{bmatrix}, \quad (5)$$

while $(W)_i = w_i$, $(V)_j = v_j$, $i = 1, \dots, q$, $j = 1, \dots, k$. In the case of just enough (minimal amount of) data, $(\mathbb{L}, \mathbb{L}_s)$, is regular. In this case $\{\hat{\mathbf{E}} = -\mathbb{L}, \hat{\mathbf{A}} = -\mathbb{L}_s, \hat{\mathbf{B}} = \mathbf{V}, \hat{\mathbf{C}} = \mathbf{W}\}$, is a minimal realization of the data. Hence, $\mathbf{G}(z) = \mathbf{W}(\mathbb{L}_s - z\mathbb{L})^{-1}\mathbf{V}$, is the required minimal solution of the problem. In the case of redundant data we use the rank revealing SVD decompositions

$$[\mathbb{L}, \mathbb{L}_s] = \mathbf{Y}\Sigma_\ell\tilde{\mathbf{X}}^*, \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \tilde{\mathbf{Y}}\Sigma_r\mathbf{X}^* \quad (6)$$

where Σ_ℓ and Σ_r are $r \times r$, with $r \ll N$.

The quadruple $\{\hat{\mathbf{E}} = -\mathbf{Y}^*\mathbb{L}\mathbf{X}, \hat{\mathbf{A}} = -\mathbf{Y}^*\mathbb{L}_s\mathbf{X}, \hat{\mathbf{B}} = \mathbf{Y}^*\mathbf{V}, \hat{\mathbf{C}} = \mathbf{W}\mathbf{X}\}$, is then, the realization of an approximate data interpolant of order r .

$$\mathbf{H}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} \quad (7)$$

1.1.1 Computation of poles/zeros and projected interpolation points Skelton [11], Van Dooren [9], Grimme [10]

Having a rational representation in state-space form, $H(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}}$ is crucial for computing its poles (roots of the denominator) and zeros (roots of the numerator), since these key quantities can be obtained, for example, by computing the finite eigenvalues resulting from the following generalized eigenvalue problems that involve the state-space matrices [12]

- poles: $\sigma(\hat{\mathbf{A}}, \hat{\mathbf{E}})$
- zeros: $\sigma\left(\begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hat{\mathbf{C}} & \hat{\mathbf{D}} \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{E}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right)$, where in this case $\hat{\mathbf{D}} = \mathbf{0}$.

Interpolatory projection matrices - (Skelton/Grimme/Van Dooren)

We introduce the following quantities which play the role of projectors in the sequel. For this we need the sets of the complex numbers namely $\mu_i, i = 1, \dots, q$, and $\lambda_j, j = 1, \dots, k$ which we will refer to as left and right interpolation points:

$$\mathcal{R} = [\phi(\lambda_1)\mathbf{B} \dots \phi(\lambda_k)\mathbf{B}] \in \mathbb{C}^{n \times k}, \quad \mathcal{O} = [\mathbf{C}\phi(\mu_1) \dots \mathbf{C}\phi(\mu_q)]^T \in \mathbb{C}^{q \times n}. \quad (8)$$

These are called the generalized **controllability** and the generalized **observability** matrices. With $\mathbf{A} = \text{diag}[\lambda_1, \dots, \lambda_k]$, $\mathbf{M} = \text{diag}[\mu_1, \dots, \mu_q]$, $\mathbf{e}_m = [1 \dots 1]^T \in \mathbb{R}^m$. Then, \mathcal{R} and \mathcal{O} satisfy the Sylvester equations:

$$\mathbf{E}\mathcal{R}\mathbf{A} - \mathbf{A}\mathcal{R} = \mathbf{B}\mathbf{e}_k^T, \quad \mathbf{M}\mathcal{O}\mathbf{E} - \mathcal{O}\mathbf{A} = \mathbf{e}_q\mathbf{C}. \quad (9)$$

By multiplying the above equations with \mathcal{O}, \mathcal{R} we obtain:

$$\mathbb{L}_s - \mathbb{L}\mathbf{A} = \mathbf{V}\mathbf{R} \text{ and } \mathbb{L}_s - \mathbf{M}\mathbb{L} = \mathbf{L}\mathbf{W}$$

By adding/subtracting appropriate multiples of these expressions it follows that the Loewner quadruple satisfies the Sylvester equations

$$\mathbf{M}\mathbb{L} - \mathbb{L}\mathbf{A} = \mathbf{V}\mathbf{R} - \mathbf{L}\mathbf{W}, \quad \mathbf{M}\mathbb{L}_s - \mathbb{L}_s\mathbf{A} = \mathbf{M}\mathbf{V}\mathbf{R} - \mathbf{L}\mathbf{W}\mathbf{A}. \quad (10)$$

Given the projectors $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{n \times k}$, let the reduced quantities be

$$\hat{\mathbb{L}} = \mathbf{X}^*\mathbb{L}\mathbf{Y}, \quad \hat{\mathbb{L}}_s = \mathbf{X}^*\mathbb{L}_s\mathbf{Y}, \quad \hat{\mathbf{V}} = \mathbf{X}^*\mathbf{V}, \quad \hat{\mathbf{L}} = \mathbf{X}^*\mathbf{L}, \quad \hat{\mathbf{W}} = \mathbf{W}\mathbf{Y}, \quad \hat{\mathbf{R}} = \mathbf{R}\mathbf{Y}. \quad (11)$$

The associated $\hat{\mathbf{A}}$, and $\hat{\mathbf{M}}$, must satisfy the projected equations above, so,

$$\hat{\mathbb{L}}_s - \hat{\mathbb{L}}\hat{\mathbf{A}} = \hat{\mathbf{V}}\hat{\mathbf{R}}, \quad \hat{\mathbb{L}}_s - \hat{\mathbf{M}}\hat{\mathbb{L}} = \hat{\mathbf{L}}\hat{\mathbf{W}}. \quad (12)$$

Let the associated generalized eigenvalue problem be:

$$\hat{\mathbf{A}} = \text{eig}(\hat{\mathbb{L}}_s - \hat{\mathbf{V}}\hat{\mathbf{R}}, \hat{\mathbb{L}}), \quad \hat{\mathbf{M}} = \text{eig}(\hat{\mathbb{L}}_s - \hat{\mathbf{L}}\hat{\mathbf{W}}, \hat{\mathbb{L}}). \quad (13)$$

From the above equations we are able to compute the **projected interpolation** points.

1.2 The recursive Loewner approach

A brief description is as follows. For details see [1]

First step:

Randomly choose two measurements. Put one of them into left measurement set and put another one into right measurement set. Generate the resulting approximant.

Second step:

- Calculate the error between the approximant and the measurement values at the sample points.
- Put the two measurements which has largest error into the left/right measurement sets respectively.
- Construct the new approximant by means of the recursive Loewner framework. Continue until all measurements are used.

1.3 The AAA algorithm

This algorithm [8] uses **barycentric representation** of interpolates :

$$\mathbf{r}(\mathbf{s}) = \frac{\mathbf{n}(\mathbf{s})}{\mathbf{d}(\mathbf{s})} = \frac{\sum_{j=1}^m \frac{\omega_j \mathbf{f}_j}{\mathbf{s} - \mathbf{s}_j}}{\sum_{j=1}^m \frac{\omega_j}{\mathbf{s} - \mathbf{s}_j}}. \quad (14)$$

First step:

Choose randomly one measurement from the measurement set. Construct the first order system $m = 1$, where \mathbf{s}_1 is the sample point and \mathbf{f}_1 is the measurement value.

Second step:

Define the error as $\mathbf{e}(\boldsymbol{\omega}) = \sum_{j=2}^n (\mathbf{f}_j \mathbf{d}(\mathbf{s}) - \mathbf{n}(\mathbf{s}))^2$, where $\boldsymbol{\omega}$, describes the weights. Using least square we can calculate $\boldsymbol{\omega}$.

Third step:

for $j = 2, \dots, k$

- Calculate the error between approximant and measurement value at sample points.
- Interpolate the measurement which has the largest error to increase the order of barycentric representations $m = j$ such that \mathbf{s}_j , is the sample point and \mathbf{f}_j , is the measurement value.
- Using least squares, calculate $\boldsymbol{\omega}$.

end

Return the barycentric representation approximation: $\mathbf{r}(\mathbf{s}) = \frac{\mathbf{n}(\mathbf{s})}{\mathbf{d}(\mathbf{s})} = \frac{\sum_{j=1}^k \frac{\omega_j \mathbf{f}_j}{\mathbf{s} - \mathbf{s}_j}}{\sum_{j=1}^k \frac{\omega_j}{\mathbf{s} - \mathbf{s}_j}}$. For more details see [8].

1.4 Vector Fitting

Consider the rational function approximation

$$f(s) = \sum_{n=1}^r \frac{\mathbf{c}_n}{s - \mathbf{a}_n} + \mathbf{d} + \mathbf{s}\mathbf{h}. \quad (15)$$

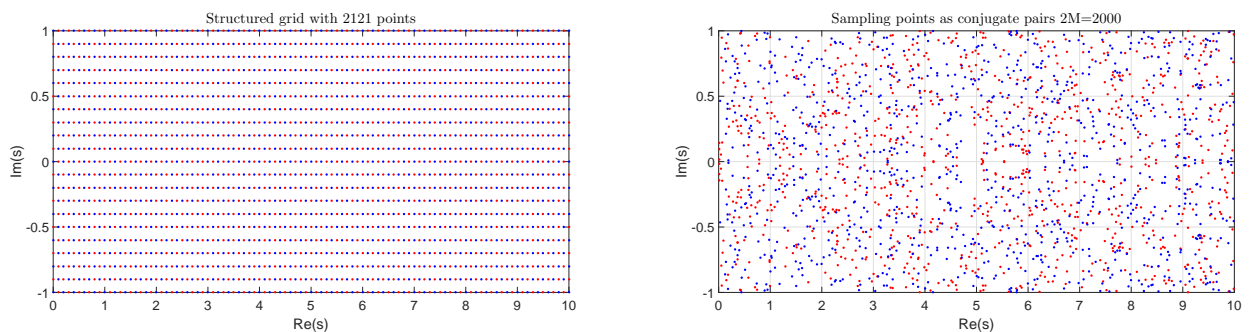
The residue \mathbf{c}_n , and poles are real or complex conjugate pairs, while \mathbf{d} and \mathbf{h} , are real. The problem is to estimate all coefficients in the above expression so that the least square error gets minimized. The above problem is a non linear problem, because \mathbf{a}_n , appears in the denominator. Vector fitting addresses this issue by solving a sequence of linear problems. The procedure is as follows. Fixing the quantities $\bar{\mathbf{a}}_i$, solve for $\bar{\mathbf{c}}_i$, \mathbf{c}_i , \mathbf{d} and \mathbf{h} , by evaluating the expression

$$\sum_{n=1}^r \left(\frac{\bar{\mathbf{c}}_n}{s - \bar{\mathbf{a}}_n} + 1 \right) f(s) = \sum_{n=1}^r \frac{\mathbf{c}_n}{s - \bar{\mathbf{a}}_n} + \mathbf{d} + \mathbf{s}\mathbf{h}. \quad (16)$$

at each interpolation point and using least squares to minimize the error. The procedure is repeated until convergence. For details, see [6, 3] or <http://www.sintef.no/projectweb/vectfit/>.

2 Approximation Example

Choosing the interpolation points. The aim is to approximate the irrational function $H(s)$, in Figure 1, in the domain Ω , which is sampled in two different ways. First using a structured grid with 2121 interpolation points (left pane below), and second, using a uniformly distributed grid with 2000 points (right pane below).



(a) Structured grid with 2121 points

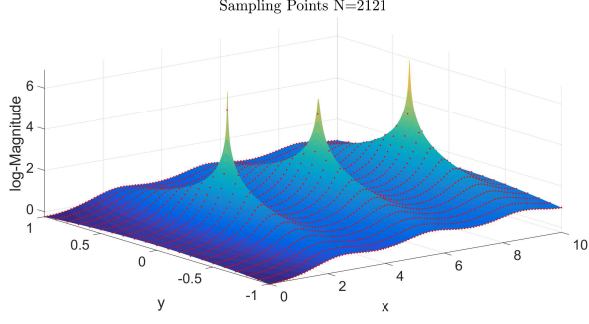
(b) Uniformly grid with 2000 sampling points.

Figure 2: In both cases we took under consideration conjugate pairs in order to enforce real symmetry.

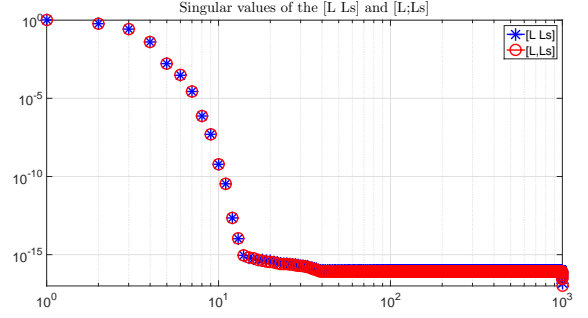
2.1 Interpolation points as in Figure 2a

2.1.1 The Loewner Framework applied to $H(s)$

First we will construct approximants $H_r(s)$, (r , order) of $H(s)$, using the interpolation points as shown in the left hand figure, section 2, page 5. Below is a description of these points with the corresponding samples of $H(s)$, (Figure 3a) and the (normalized) singular values of the augmented Loewner matrix (Figure 3b). We truncate at $r=11$ with corresponding accuracy $O(10^{-12})$.



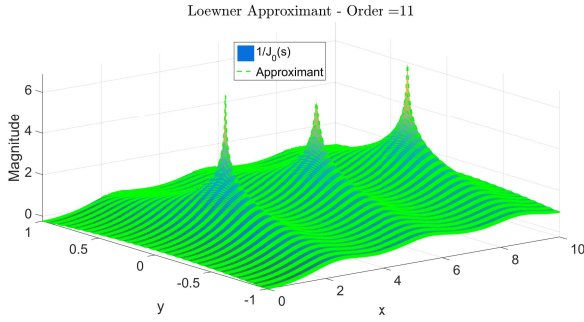
(a) Sampling points over $\Omega = [0, 10] \times [-1, 1]$



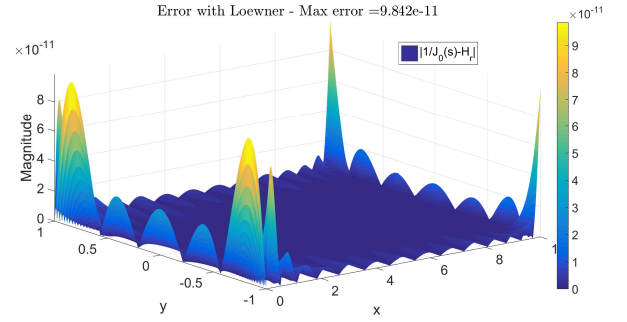
(b) Fast decay of the singular values.

Figure 3: Interpolation scheme of the irrational function and the corresponding rank revealing SVD of the augmented Loewner matrix.

The next plot (Figure 4) shows the 11th order approximant $H_r(s)$, superimposed on the plot of $H(s)$, in Ω , over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result we see the error plot in Figure 4b which shows for each point in the Ω_{grid} , the modulo $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-11})$.

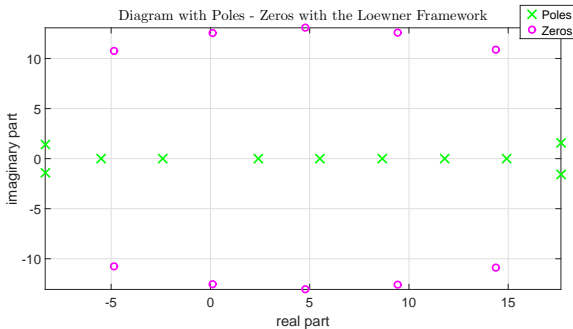


(a) $H(s)$, with Loewner Approximant

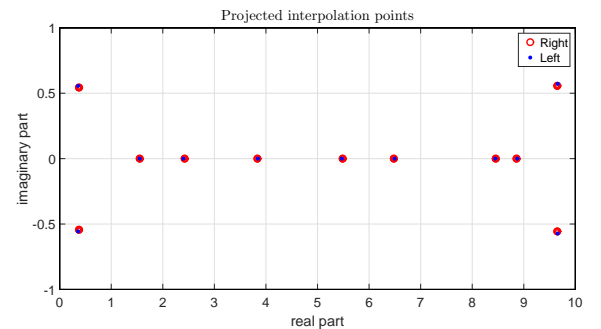


(b) Absolute error of $|H(s) - H_r|$, over the Ω , domain.

Figure 4: Approximant and the error over the dense domain Ω_{grid} .



(a) Poles \times , Zeros \circ .



(b) Right \circ and Left \bullet .

Figure 5: Poles/zeros diagram and projected interpolation points in Ω .

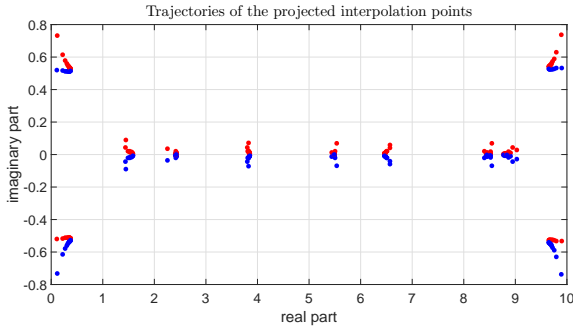
Last, the corresponding poles/zeros and the projected left/right interpolation points are shown in the Figure 5. The real part of the poles is shown with 15 digits accuracy, while the imaginary

part with 5. Also the poles marked in bold correspond to the original zeros of $J_0(s)$, in Ω , appendix 4. The ensuing left and right sets containing 11 projected interpolation points each (with 5 digits precision) are shown next.

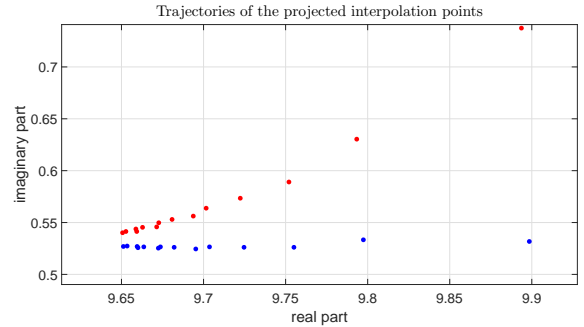
Remark: Thus, the initial 2121 interpolation points in Ω , are now compressed to 22.

$$\text{Poles} = \begin{pmatrix} -8.32213293322054 - 1.4252i \\ -8.32213289862456 + 1.4252i \\ -5.51461491999547 \\ -2.40481847965605 \\ \mathbf{2.40482555769577} \\ \mathbf{5.52007811028631} \\ \mathbf{8.65372791291101} \\ 11.7915356008908 \\ 14.9135964357538 \\ 17.6548692348549 - 1.561i \\ 17.654869354827 + 1.561i \end{pmatrix}, \text{Zeros} = \begin{pmatrix} -4.8491 - 10.766i \\ -4.8491 + 10.766i \\ 0.12013 - 12.537i \\ 0.12013 + 12.537i \\ 4.785 + 13.066i \\ 4.785 - 13.066i \\ 9.4384 + 12.591i \\ 9.4384 - 12.591i \\ 14.367 + 10.868i \\ 14.367 - 10.868i \end{pmatrix}, \text{Right} = \begin{pmatrix} 0.37181 + 0.54303i \\ 0.37181 - 0.54303i \\ 1.5504 \\ 2.424 \\ 3.8331 \\ 5.4898 \\ 6.479 \\ 8.4519 \\ 8.8561 \\ 9.6519 - 0.55566i \\ 9.6519 + 0.55566i \end{pmatrix}, \text{Left} = \begin{pmatrix} 0.36164 - 0.55711i \\ 0.36164 + 0.55711i \\ 1.5491 \\ 2.4005 \\ 3.8418 \\ 5.473 \\ 6.4935 \\ 8.4516 \\ 8.8759 \\ 9.6605 - 0.57111i \\ 9.6605 + 0.57111i \end{pmatrix}.$$

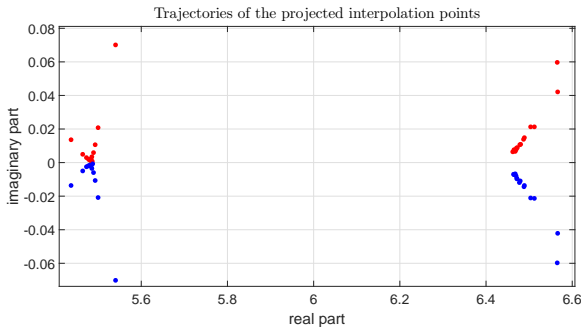
Trajectories of the projected interpolation points. If we increase the number of interpolation points in Ω , the **left and right projected interpolation points seem to converge**. If we choose a specific way to partition the measurements (left and right), i.e. $\max_{i,j} |\mu_i - \lambda_j| < \epsilon$, then we noticed that the corresponding pairs of projected points are ϵ order, close each other. The trajectories shown below use from 100 to almost 20000 initial interpolation points in Ω . By creating an increasingly dense Ω_{grid} , we observe the following convergence patterns. More specifically, we densify symmetrically Ω domain as follows: $a \times a, 2a \times 2a, \dots, na \times na$, with $a, n \in \mathbb{N}$, (i.e. $a = 10$, and $n = 14$) and for each grid we solve the set of equations (13). Only some pairs of trajectories out of 11 pairs are shown, as the remaining pairs have similar convergence behavior. Finally, it turns out that the projected interpolation points remain in Ω .



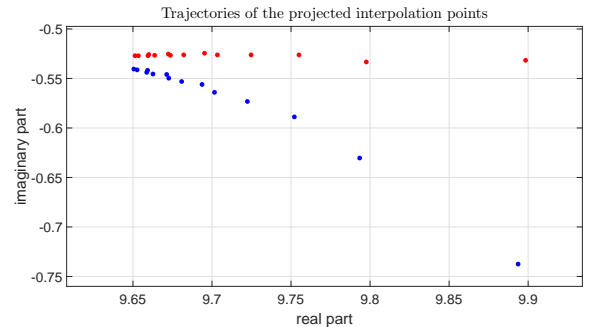
(a) Trajectories of the projected interpolation points



(b) Zoom on the upper right of Figure 6a.



(c) Convergence with real symmetry



(d) Conjugate trajectory of Figure 6b.

Figure 6: Pairs of trajectories of the projected interpolation points under increasingly denser grids. In some cases it seems that the convergence has oscillatory behavior. Also, left and right projected interpolation points seem to coincide when we select a particular way of separating the data (left-right).

2.1.2 The recursive Loewner algorithm to $H(s)$

The measurements are as before. The resulting approximant compared with the original is as follows.

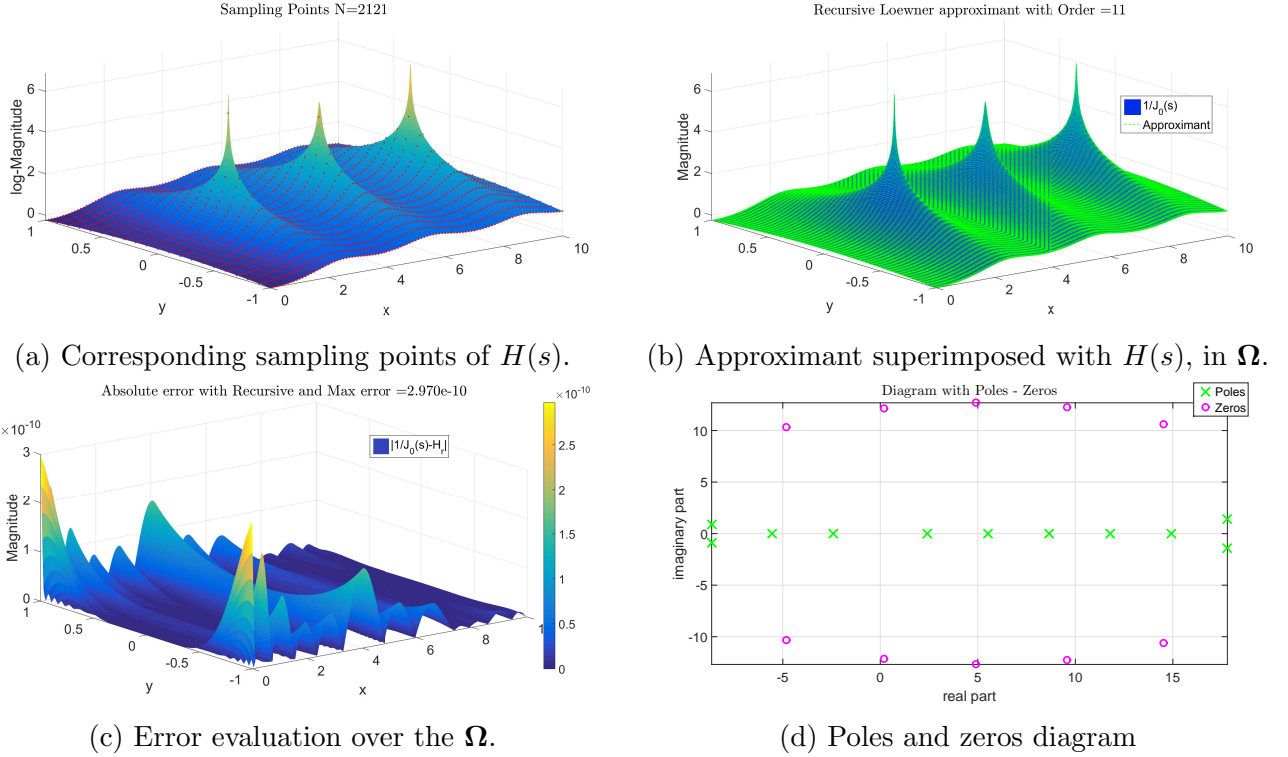


Figure 7: Approximation results of the $H(s)$, over Ω , with the recursive Loewner approach.

In more detail the Figure 7a describes the interpolation points as shown in the left hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 7b shows the 11th order approximant with the recursive Loewner superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result is the error plot in Figure 7c which shows for each point in the Ω_{grid} , the distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-10})$. Last, the corresponding poles/zeros are shown in the Figure 7d and for the exact values of them, someone can see Appendix 4 number 2.

Remark 2.1.2.1: With the recursive Loewner approach algorithm we need to choose the order of the approximant.

Remark 2.1.2.2: With the recursive Loewner approach algorithm usually we notice that the error is distributed more uniformly over Ω .

2.1.3 The AAA algorithm applied to $H(s)$

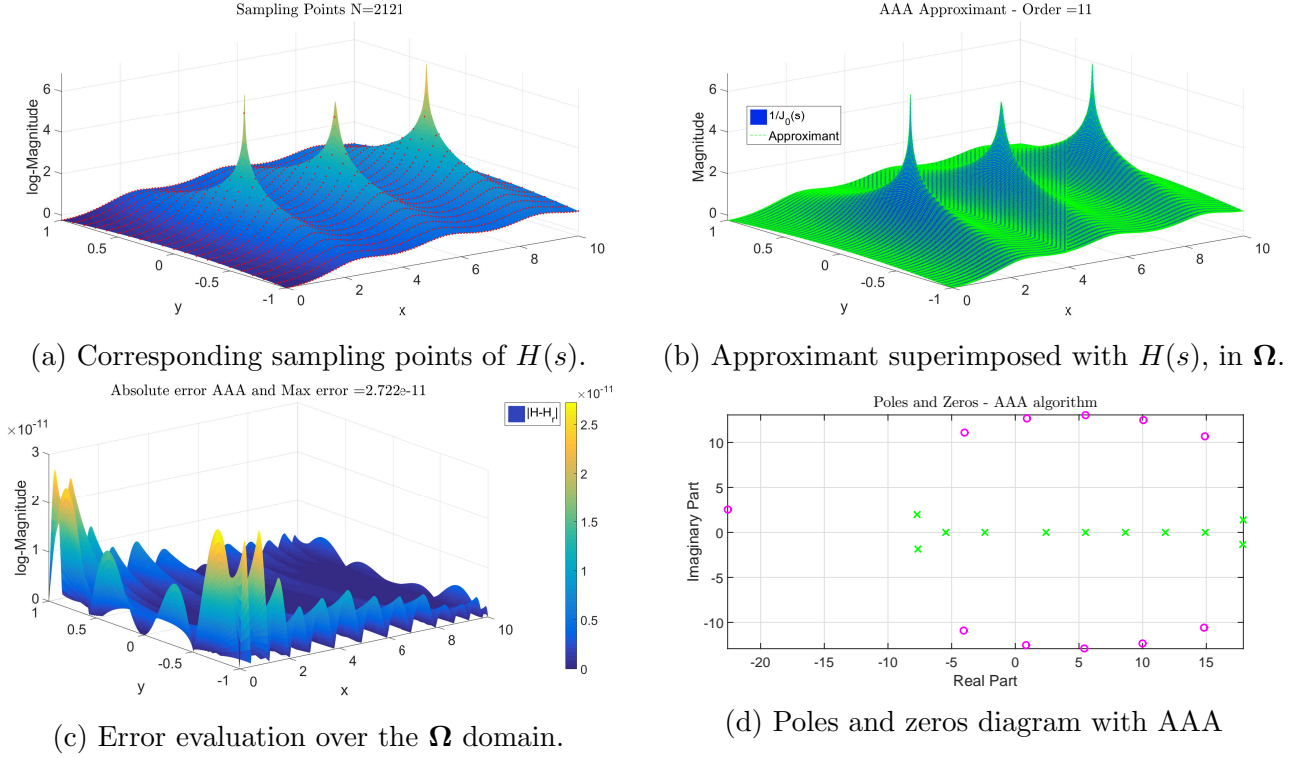


Figure 8: Approximation results of the $H(s)$, over Ω , with the AAA algorithm approach.

In more detail the Figure 8a describes the interpolation points as shown in the left hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 8b shows the 11th order approximant with the AAA approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result is the error plot in Figure 8c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-11})$. Last, the corresponding poles/zeros are shown in the Figure 8d and for the exact values, someone can see Appendix 4 number 3.

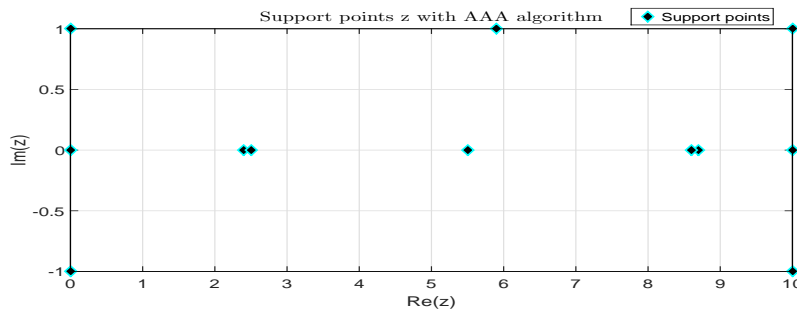
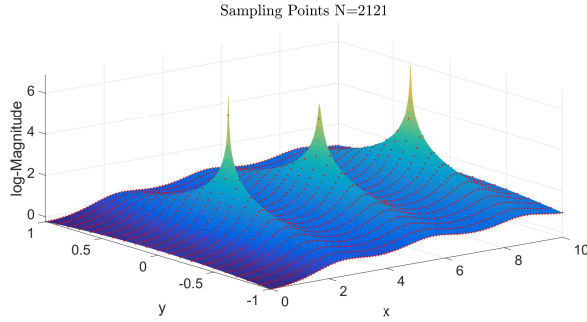


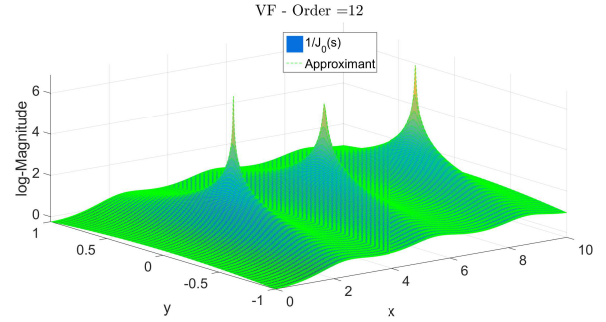
Figure 9: Support points from AAA algorithm in Ω .

2.1.4 The Vector Fitting (VF) method applied to $H(s)$

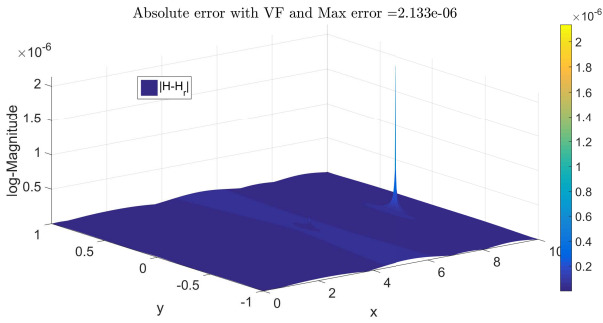
The measurements are as before:



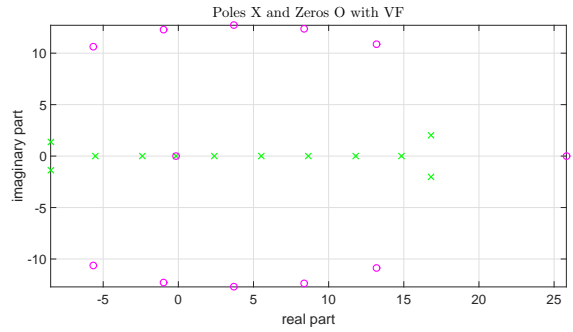
(a) Corresponding sampling points of $H(s)$.



(b) Approximant superimposed with $H(s)$, in Ω .



(c) Error evaluation over the Ω .



(d) 12 poles/zeros with one pole/zero cancellation

Figure 10: Approximation results of the $H(s)$, over Ω , with the Vector Fitting algorithm approach.

In more detail the Figure 10a describes the interpolation points as shown in the left hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 10b shows the 11th order approximant with the VF approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result is the error plot in Figure 10c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$ with $s \in \Omega_{grid}$. The order of error is $O(10^{-6})$. Last, the corresponding poles/zeros are shown in the Figure 10d and for the exact values, someone can see Appendix 4 number 4.

Remark 2.1.4.1 Since the 5th pole and the first zero above are equal up to 10^{-9} , (Figure 8d) or Appendix 4 number 4, they can be eliminated. After this **zero/pole** cancellation we obtain the same order of accuracy with an approximate of order $r = 11$.

Remark 2.1.4.2 We notice that the biggest error appears in the vicinity of the 3rd pole. By changing the weights distribution we can improve the error locally.

Remark 2.1.4.3 VF method also needs as an input the order of the approximant.

2.2 Interpolation points as in figure 2b

2.2.1 The Loewner Framework applied to $H(s)$.

We now make use of the uniformly distributed interpolation points as in Figure 2b.

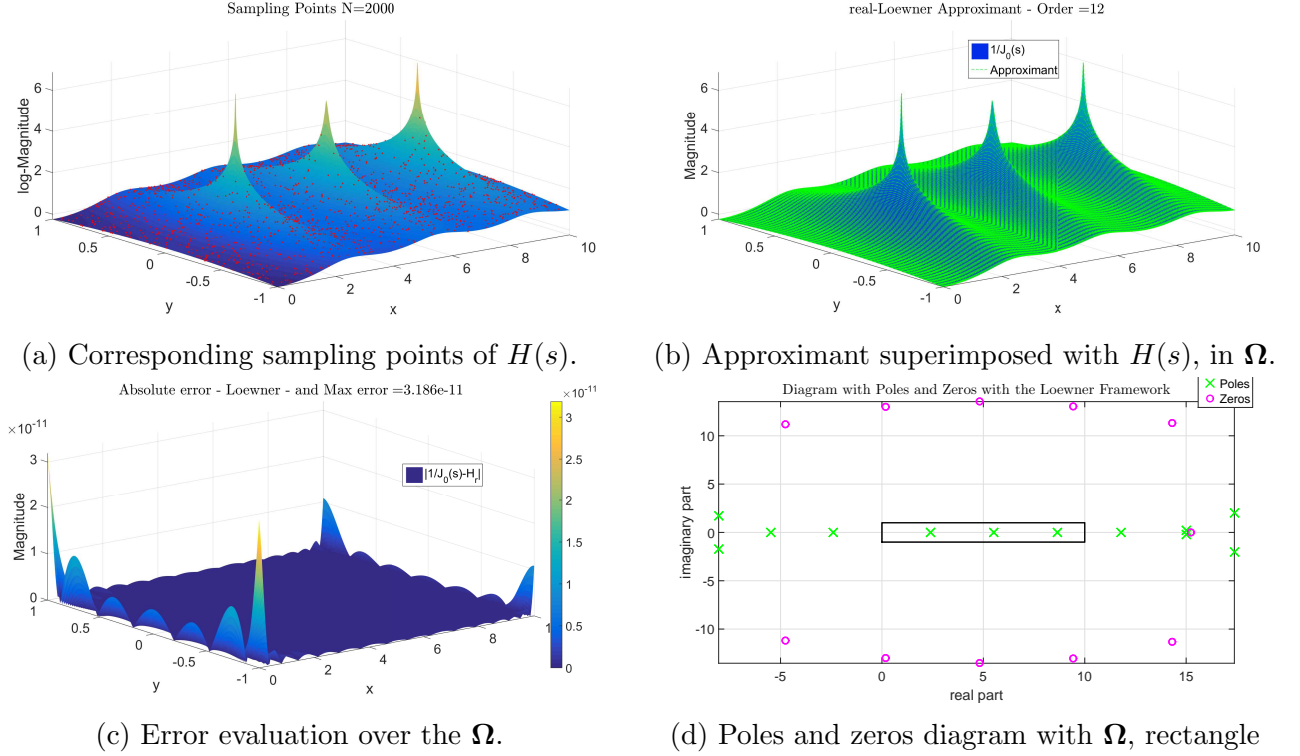


Figure 11: Approximation results of the $H(s)$, over Ω , with the Loewner framework approach.

In more detail the Figure 11a describes the interpolation points as shown in the right hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 11b shows the 12th order approximant with the Loewner framework approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result we see the error plot in Figure 11c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-11})$. Last, the corresponding poles/zeros are shown in the Figure 11d and for the exact values, someone can see Appendix 4 number 5.

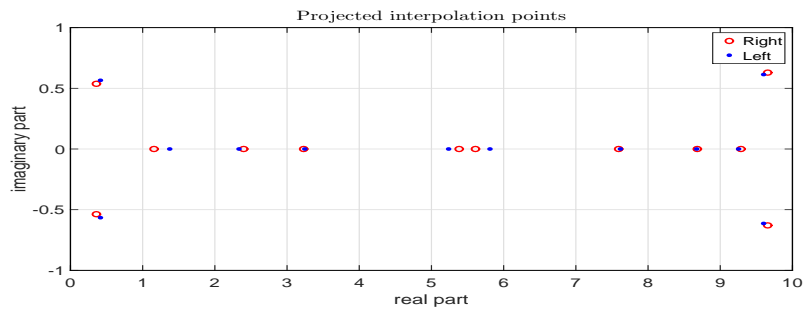


Figure 12: Left and right projected interpolation points in Ω .

2.2.2 The recursive Loewner Framework applied to $H(s)$.

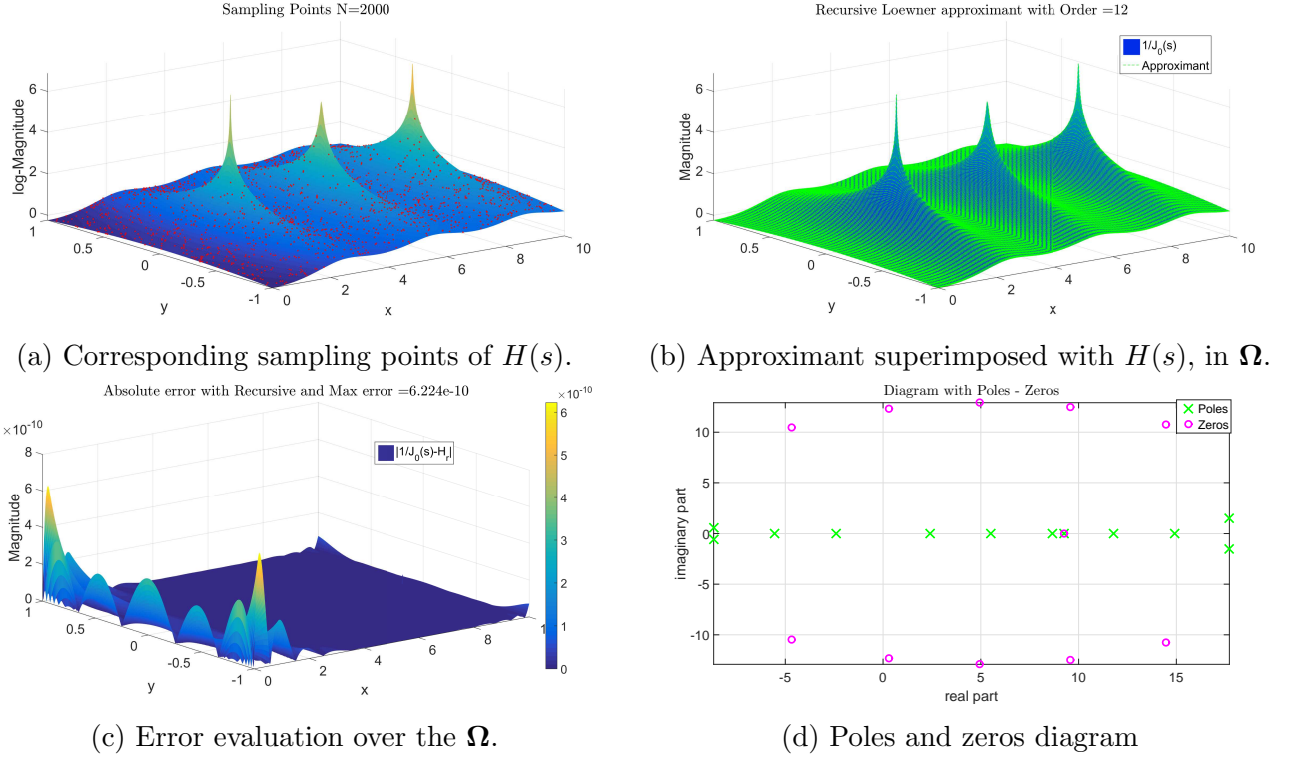


Figure 13: Approximation results of the $H(s)$, over Ω , with the recursive Loewner algorithm approach.

In more detail the Figure 13a describes the interpolation points as shown in the right hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 13b shows the 12th order approximant with the recursive Loewner framework approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result we see the error plot in Figure 13c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-10})$. Last, the corresponding poles/zeros are shown in the Figure 13d and for the exact values, someone can see Appendix 4 number 6.

Remark 2.2.2.1 Since the 8th pole and the last zero above are equal up to 10^{-5} , (Figure 13d or Appendix 4 number 6, they can not be eliminated because it is less than the error. So in this case recursive algorithm produce 12th order approximant with almost a pole/zero cancellation.

2.2.3 The AAA Algorithm over Uniformly Distributed Points

In this subsection we present the approximation with AAA algorithm where the input uses 2000 points as in figure 2b. After many realizations with uniformly distributed grid (2000 points), we noticed that the error varies between $O(1e - 11)$, to $O(1e - 13)$. We kept one of the best.

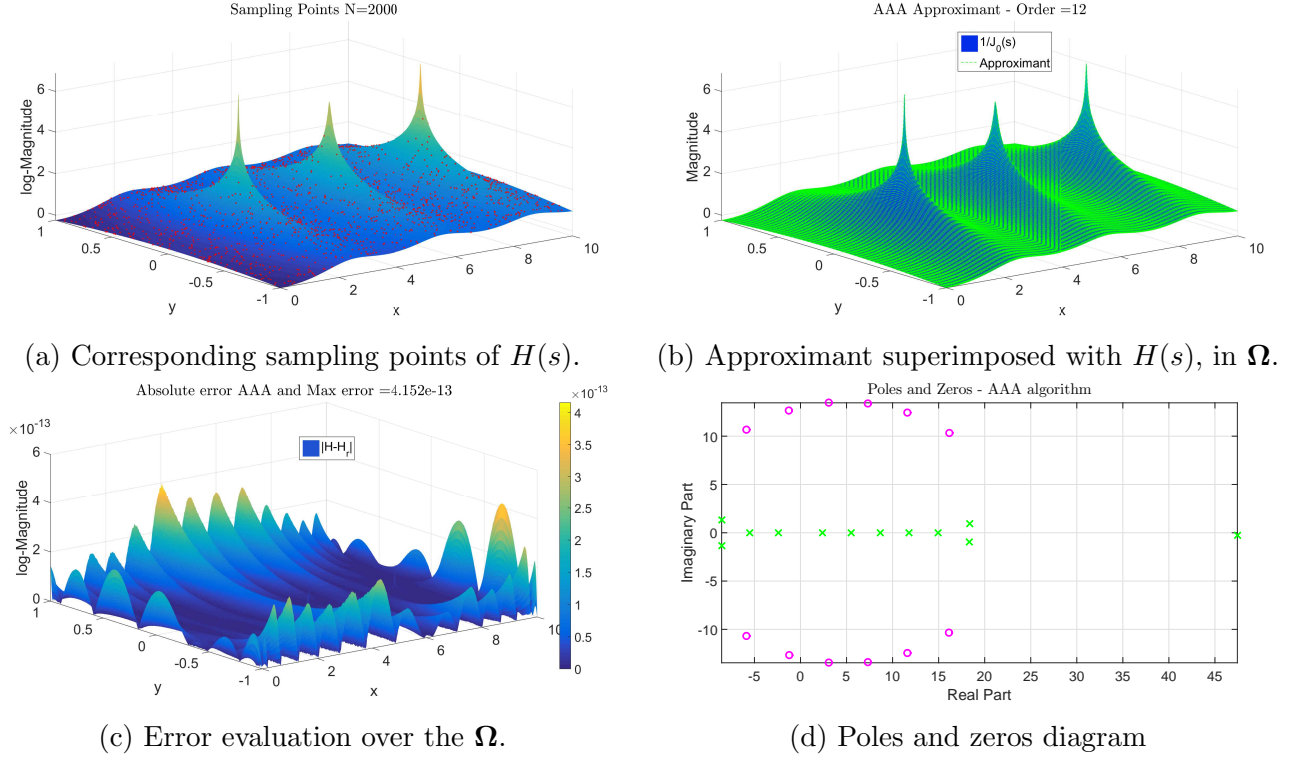


Figure 14: Approximation results of the $H(s)$, over Ω , with the AAA algorithm approach.

In more detail the Figure 14a describes the interpolation points as shown in the right hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 14b shows the 12th order approximant with the AAA algorithm approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result we see the error plot in Figure 14c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-13})$. Last, the corresponding poles/zeros are shown in the Figure 14d and for the exact values, someone can see Appendix 4 number 7.

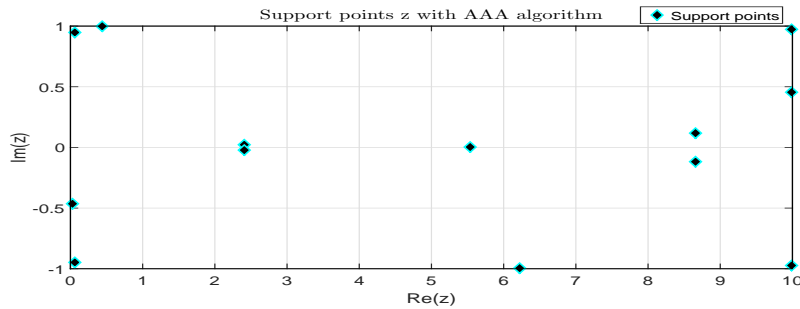


Figure 15: Support points from AAA algorithm in Ω .

2.2.4 Vector Fitting for uniformly distributed points

In this subsection we present the approximation with VF algorithm where we use 2000 (conjugate pairs) uniformly distributed over the domain Ω , as in figure 2b.

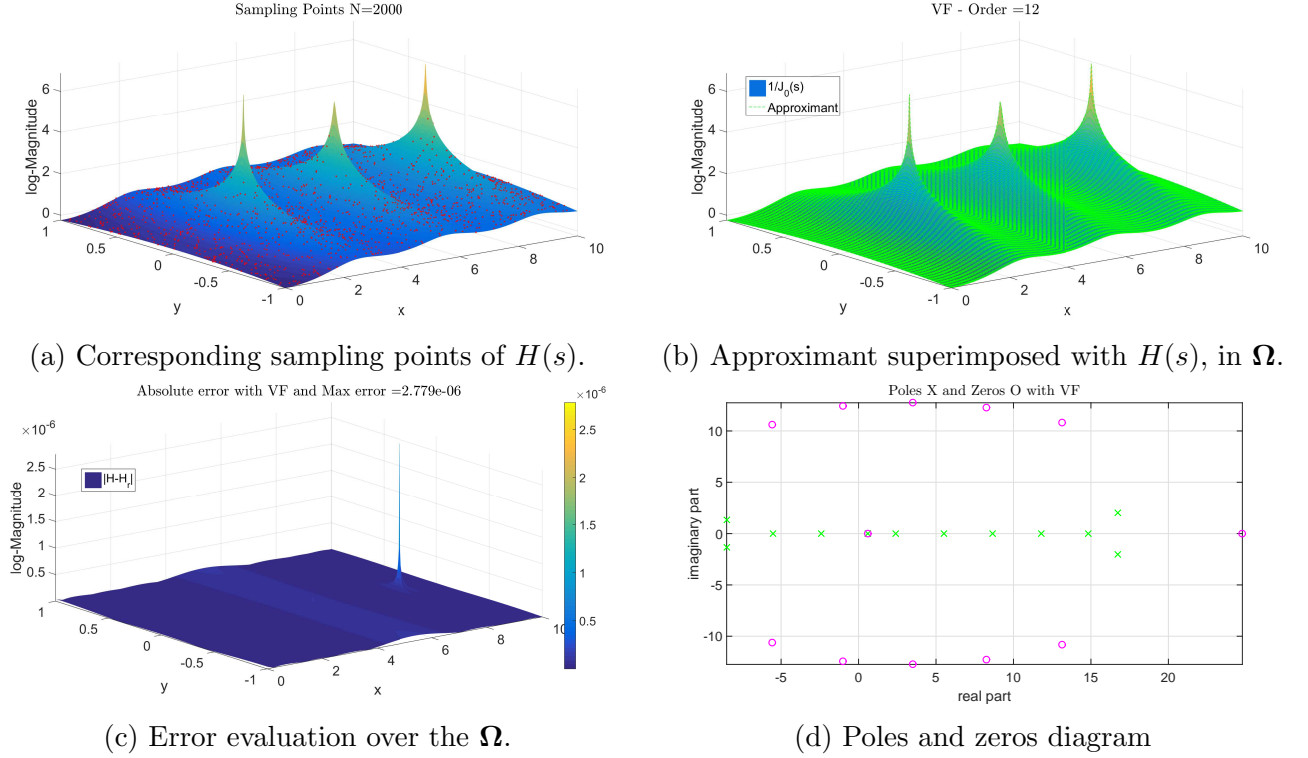


Figure 16: Approximation results of the $H(s)$, over Ω , with the Vector Fitting algorithm approach.

In more detail the Figure 16a describes the interpolation points as shown in the right hand figure, section 2, page 5 with a depiction of these points on the corresponding samples of $H(s)$. The Figure 16b shows the 12th order approximant with the AAA algorithm approach superimposed on the plot of $H(s)$, over the dense grid $\Omega_{grid} = [x_1, \dots, x_{500}] \times [y_1, \dots, y_{500}] \subset \Omega$. As a result is the error plot in Figure 16c which shows for each point in the Ω_{grid} , the absolute distance $|H_r(s) - H(s)|$, with $s \in \Omega_{grid}$. The order of error is $O(10^{-6})$. Last, the corresponding poles/zeros are shown in the Figure 16d and for the exact values, someone can see Appendix 4 number 8.

Remark: Since the one pole/zero pair above are equal up to 10^{-9} , (Figure 16d) or Appendix 4 number 8, they can be eliminated. After this **zero/pole** cancellation we obtain the same order of accuracy with an approximate of order $r = 11$.

3 Conclusion

3.1 Error comparison

Case/Method	Loewner	recursive Loewner	AAA	VF
1st: 2121 structured points	$O(10^{-11})$	$O(10^{-10})$	$O(10^{-11})$	$O(10^{-6})$
2nd: 2000 uniformly points	$O(10^{-11})$	$O(10^{-10})$	$O(10^{-13})$	$O(10^{-6})$

The above results show that all methods constructed low order models with very good accuracy. Among them, the methods that built models with the highest accuracy are the direct method with the Loewner Framework and the iterative AAA algorithm. It is worthwhile to note that these two methods, in contrast to (recursive Loewner Framework and Vector Fitting), do not require the a-priori assignment of the order of the reduced system. The Loewner framework method resolves an entire SVD and from the way the singular values decay one can choose the truncation order. This gives a trade off between accuracy and complexity. On the other hand, the AAA algorithm, constructs the model by selecting on each step one more support point and solving a SVD on each iteration step. When the error drops down, AAA fullfills its exit criterion (i.e. $\text{tol}=1e-13$). Another point that is worth mentioning is that the Loewner framework method constructs models $(r - 1, r), r \in \mathbb{N}$, with real symmetry as opposed to the AAA algorithm which makes approximants of (r, r) , and with no real symmetry. The first number of the previous parenthesis stands for the order of the numerator and the second for the denominator order. This approach gives the opportunity for the Loewner framework method to introduce an arbitrary parameter D (parametrize the system, details see [1] **theorem 1.13 b**) where its calculation can reduce the error as in [7]. Another point is, that if we wanted to have the best possible approximation with the Loewner Framework, we could choose order 13th in both cases (Figure 2). The error would be: $O(10^{-13})$. We maintained a lower-order model (11th for Figure 2a and 12th for Figure 2b) in order to do a comparison with the AAA algorithm.

Last but not least, as we feed the system with conjugate measurements in both cases, someone should be able to reveal the underlying real model. The AAA algorithm does not impose real symmetry. For all the above methods we build a real approximant except for the AAA aproximant. Introducing some changes to the AAA alorithm, now we are able to construct a reduced model with real symmetry. In this particular case, we noticed that the order of the approximant becomes higher and produce many poles/zeros cancellations (Froissart doublets) up to precision. In order to tackle this problem, it is needed to enable the function "cleanup" as explained in [8]. Both functions increase the complexity for the construction of a real approximant.

3.2 Complexity comparison

The Complexity of the Loewner Framework alorith and AAA algorithm.

The main computational cost, for both methods comes from SVD calculation. For a matrix A with dimension $m \times n$, the SVD cost is $O(m^2n + mn^2 + n^3)$. Since the Loewner Framework involves SVDs of dimension $(\frac{N}{2} \times \frac{N}{2})$, the resulting complexity is $2O(\frac{3N^3}{8}) \approx O(N^3)$, flops. Algorithm AAA involves SVDs of dimensions $(M - j) \times j$, with $j = 1, 2, \dots, m$, (m , stands for the order) its complexity is $O(Mm^3)$, flops [8] instead of the Loewner framework which solves only one SVD with complexity $O(M^3)$. For this problem on Bessel approximation, two methods succeed to construct reduced models with low order approximants. The Loewner framework would produce better computational results in the case that we have a slow decay of the singular values, so it would be

needed to truncate in a higher order. This is something that will increase the computational cost of AAA algorithm.

4 Appendix

The Bessel function zeros of the first kind. The first 6 solutions of the equation $J_0(s) = 0$, with 15 digits precision are:

$$\text{Bessel original Zeros with 15 digits} = \begin{pmatrix} \mathbf{2.40482555769577} \\ \mathbf{5.52007811028631} \\ \mathbf{8.65372791291101} \\ 11.7915344390142 \\ 14.9309177084877 \\ 18.0710639679109 \end{pmatrix}.$$

1. Poles/zeros from the Loewner Framework with 2121 points grid

$$\text{Poles} = \begin{pmatrix} -8.32213293322054 - 1.4252 i \\ -8.32213289862456 + 1.4252 i \\ -5.51461491999547 \\ -2.40481847965605 \\ \mathbf{2.40482555769577} \\ \mathbf{5.52007811028631} \\ \mathbf{8.65372791291101} \\ 11.7915356008908 \\ 14.9135964357538 \\ 17.6548692348549 - 1.561 i \\ 17.654869354827 + 1.561 i \end{pmatrix}, \text{Zeros} = \begin{pmatrix} -4.8491 - 10.766 i \\ -4.8491 + 10.766 i \\ 0.12013 - 12.537 i \\ 0.12013 + 12.537 i \\ 4.785 + 13.066 i \\ 4.785 - 13.066 i \\ 9.4384 + 12.591 i \\ 9.4384 - 12.591 i \\ 14.367 + 10.868 i \\ 14.367 - 10.868 i \end{pmatrix}.$$

Remark: With **bold** are the poles inside the Ω , domain.

2. Poles/zeros from the recursive Loewner approach with 2121 points grid

$$\text{PolesRec} = \begin{pmatrix} -8.64763053774167 - 0.874672 i \\ -8.64763053774167 + 0.874672 i \\ -5.53736536596127 \\ -2.40481506507015 \\ \mathbf{2.40482555769639} \\ \mathbf{5.52007811028556} \\ \mathbf{8.65372791291091} \\ 11.791534687103 \\ 14.9207741074572 \\ 17.7972322878303 - 1.42773 i \\ 17.7972322878303 + 1.42773 i \end{pmatrix}, \text{ZerosRec} = \begin{pmatrix} -4.81774 - 10.3428 i \\ -4.81774 + 10.3428 i \\ 0.196012 - 12.134 i \\ 0.196012 + 12.134 i \\ 4.89894 - 12.6902 i \\ 4.89894 + 12.6902 i \\ 9.58028 - 12.2575 i \\ 9.58028 + 12.2575 i \\ 14.5169 - 10.5921 i \\ 14.5169 + 10.5921 i \end{pmatrix}.$$

3. Poles/zeros from the AAA approach with 2121 points grid

$$\text{PolesAAA} = \begin{pmatrix} -7.70989384927742 + 1.98333 i \\ -7.64706106530066 - 1.84221 i \\ -5.46205580369784 + 0.00647 i \\ -2.40483752963882 \\ \mathbf{2.40482555769577} \\ \mathbf{5.52007811028632} \\ \mathbf{8.65372791291101} \\ 11.7915347392982 \\ 14.9221430048388 + 0.00059 i \\ 17.8709511796541 - 1.35084 i \\ 17.9082708199292 + 1.39389 i \end{pmatrix}, \text{ZerosAAA} = \begin{pmatrix} -4.05365 - 10.9078 i \\ -3.99775 + 11.0922 i \\ 0.832526 - 12.5068 i \\ 0.905811 + 12.6642 i \\ 5.42377 - 12.9188 i \\ 5.49875 + 13.0562 i \\ 9.99811 - 12.3601 i \\ 10.0689 + 12.476 i \\ 14.8259 - 10.5757 i \\ 14.892 + 10.666 i \\ -22.5906 + 2.52206 i \end{pmatrix}.$$

Remark: Poles and Zeros do not appear as conjugates.

4. Poles/zeros from the Vector Fitting algorithm approach with 2121 points grid

$$\text{PolesVF} = \begin{pmatrix} -8.49629987353267 - 1.3597 i \\ -8.49629987353267 + 1.3597 i \\ -5.51126493255679 \\ -2.40482600718294 \\ \mathbf{-0.157905555129157} \\ \mathbf{2.40482555759485} \\ \mathbf{5.52007811166818} \\ \mathbf{8.65372790279672} \\ 11.7915516528236 \\ 14.8446346140717 \\ 16.8161497628161 - 2.0178 i \\ 16.8161497628161 + 2.0178 i \end{pmatrix}, \text{ZerosVF} = \begin{pmatrix} \mathbf{-0.157905555129157} \\ -5.6578 - 10.624 i \\ -5.6578 + 10.624 i \\ -0.9698 + 12.28 i \\ -0.9698 - 12.28 i \\ 3.683 - 12.71 i \\ 3.683 + 12.71 i \\ 8.3717 + 12.366 i \\ 8.3717 - 12.366 i \\ 13.196 - 10.889 i \\ 13.196 + 10.889 i \\ 25.816 \end{pmatrix}.$$

Remark: red color pole/zero pair shows cancellation

5. Poles/zeros from the Loewner Framework with 2000 uniformly points

$$\text{PolesRL} = \begin{pmatrix} -8.05158320855723 - 1.7303 i \\ -8.05158320855723 + 1.7303 i \\ -5.48778972998639 \\ -2.40483192639586 \\ \mathbf{2.40482555769577} \\ \mathbf{5.52007811028632} \\ \mathbf{8.65372791291101} \\ 11.7915339297017 \\ 15.0015209902362 - 0.222 i \\ 15.0015209902362 + 0.222 i \\ 17.377289961194 - 2.03 i \\ 17.377289961194 + 2.03 i \end{pmatrix}, \text{ZerosRL} = \begin{pmatrix} -4.7536 - 11.207 i \\ -4.7536 + 11.207 i \\ 0.17552 - 12.998 i \\ 0.17552 + 12.998 i \\ 4.8076 - 13.534 i \\ 4.8076 + 13.534 i \\ 15.242 \\ 9.4292 - 13.06 i \\ 9.4292 + 13.06 i \\ 14.319 - 11.333 i \\ 14.319 + 11.333 i \end{pmatrix}.$$

6. Poles/zeros from the recursive Loewner with 2000 uniformly points

$$\text{polesREC} = \begin{pmatrix} -8.66473644838372 - 0.55511 i \\ -8.66473644838372 + 0.55511 i \\ -5.55846358891533 \\ -2.40478725513252 \\ 2.40482555769579 \\ 5.52007811028631 \\ 8.65372791291094 \\ \mathbf{9.25763698485788} \\ 11.7915351867464 \\ 14.9164167045242 \\ 17.7116227515344 - 1.513 i \\ 17.7116227515344 + 1.513 i \end{pmatrix}, \text{ZerosREC} = \begin{pmatrix} 14.465 + 10.772 i \\ 14.465 - 10.772 i \\ -4.6782 + 10.478 i \\ -4.6782 - 10.478 i \\ 9.5668 + 12.474 i \\ 9.5668 - 12.474 i \\ 0.29543 + 12.334 i \\ 0.29543 - 12.334 i \\ 4.9411 + 12.915 i \\ 4.9411 - 12.915 i \\ \mathbf{9.25764464786048} \end{pmatrix}.$$

7. Poles/zeros from the AAA with 2000 uniformly points

$$\text{PolesAAA} = \begin{pmatrix} -8.56002747182546 - 1.3357 i \\ -8.54868176358933 + 1.3357 i \\ -5.51154958824584 \\ -2.40482617183982 \\ \mathbf{2.40482555769577} \\ \mathbf{5.52007811028630} \\ \mathbf{8.65372791291101} \\ 11.7915344794608 \\ 14.9286212759068 + 0.000044838 i \\ 18.349207798415518.349 - 0.93836 i \\ 18.3564686110828 + 0.94294 i \\ 47.4348759491114 - 0.25113 i \end{pmatrix}, \text{ZerosAAA} = \begin{pmatrix} -5.8598 - 10.679 i \\ -5.8806 + 10.684 i \\ -1.2407 - 12.66 i \\ -1.2625 + 12.659 i \\ 3.062 + 13.453 i \\ 3.0802 - 13.46 i \\ 7.3122 + 13.387 i \\ 7.3233 - 13.396 i \\ 11.605 + 12.456 i \\ 11.608 - 12.463 i \\ 16.139 - 10.328 i \\ 16.142 + 10.327 i \end{pmatrix}.$$

Remark: Poles and Zeros do not appear as conjugates.

8. Poles/zeros from the VF with 2000 uniformly points

$$\text{PolesVF} = \begin{pmatrix} -8.51428135857578 - 1.3424 i \\ -8.51428135857578 + 1.3424 i \\ -5.51184166025835 \\ -2.40482593907193 \\ \mathbf{0.598123905184851} \\ \mathbf{2.40482555782458} \\ \mathbf{5.52007810888635} \\ \mathbf{8.65372792599631} \\ 11.7915548184731 \\ 14.8352178424186 \\ 16.7433767283711 - 2.0413 i \\ 16.7433767283711 + 2.0413 i \end{pmatrix}, \text{ZerosVF} = \begin{pmatrix} \mathbf{0.598123905506776} \\ -5.5835 - 10.635 i \\ -5.5835 + 10.635 i \\ -1.0088 + 12.445 i \\ -1.0088 - 12.445 i \\ 3.4938 + 12.744 i \\ 3.4938 - 12.744 i \\ 8.2407 - 12.29 i \\ 8.2407 + 12.29 i \\ 13.127 + 10.826 i \\ 13.127 - 10.826 i \\ 24.775 \end{pmatrix}.$$

Remark: red color pole/zero pair shows cancellation

References

- [1] A.C. ANTOULAS, S. LEFTERIU AND A.C. IONITA, *A tutorial introduction to the Loewner Framework for Model Reduction*, in *Model Reduction and Approximation: Theory and Algorithms*, edited by: P. Benner, M. Ohlberger, A. Cohen & K. Willcox, Series: Computational Science Engineering, pages 335-376, SIAM, Philadelphia, 2017,
- [2] A.C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, *Advances in Design and Control*, <https://doi.org/10.1137/1.9780898718713>, SIAM, Philadelphia, 2005,
- [3] Z. DRMAC, S. GUGERCIN AND C.A. BEATTIE, *Vector Fitting for Matrix-valued Rational Approximation*, SIAM JOURNAL ON SCIENTIFIC COMPUTING, VOL.37, ISSUE:5, PP. A2151-S626, 2015
- [4] G. FLAGG, C.A. BEATTIE AND S. GUGERCIN, *Interpolatory H_{∞} model reduction*, SYSTEMS AND CONTROL LETTERS, **62**: 567-574 (2013)
- [5] S. Gugercin, A.C. Antoulas and C.A. Beattie, *H_2 model reduction for large-scale linear dynamical systems* SIAM Journal on Matrix Analysis and Applications Vol. 32, Issue: 2, pp. 609-638, 2008
- [6] B. GUSTAVSEN AND A. SEMLYEN, *Rational Approximation of frequency domain responses by vector fitting*, IEEE Trans. Power Delivery, vol. 14, no. 3, pp. 1052-1061, July 1999
- [7] C. MAGRUDER, C.A. BEATTIE AND S. GUGERCIN, *Rational Krylov methods for optimal L_2 model reduction*, PROC. 49TH IEEE CONFERENCE ON DECISION AND CONTROL, PP. 6797-6802, 2010
- [8] Y. NAKATSUKASA, O. SETE AND L.N. TREFETHEN, *The AAA Algorithm for rational approximation*, arXiv: 1612.00337 [math.NA], 29 pages, 2016
- [9] K. GALLIVAN, A. VANDENDORPE AND P. VAN DOOREN, *Model Reduction of MIMO Systems via Tangential Interpolation*, SIAM J. Matrix Anal. Applic. Vol.26, No.2, pp.328-349, 2004
- [10] ERIC JAMES GRIMME, *Krylov Projection Methods For Model Reduction*, Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 1997
- [11] SKELTON, R.E., *Observability Measures and Performance Sensitivity in the Model Reduction Problem*, International Journal of Control, Vol. 29, No. 4, pp. 541-556, 1979.
- [12] A.C. IONITA, *Lagrange Rational Interpolation and its Applications to Approximation of Large-Scale Dynamical Systems*, PhD thesis, Rice University, 2013