

Design of a Matlab toolbox and GUI for minimal realizations

Karim Cherifi, Kamel Hariche

The institute of electrical and electronic engineering, M'hamed Bougara University, Algeria

Article Info

Article history:

Received May 26, 2019

Revised Jul 27, 2019

Accepted Aug 11, 2019

Keywords:

Computer-aided control

Control engineering

Minimal realization

System modelling

ABSTRACT

Minimal realization for linear systems has been studied extensively in the literature. Techniques proposed differ in terms of the configuration of the state space form, the time efficiency of the algorithm and the reduction of memory storage. The minimal realization toolbox presented in this paper offers the possibility to test different methods and choose the most suitable form depending on the application. The given system can be represented by a transfer function or by raw (frequency or time domain) data. A graphical user interface (GUI) was implemented in order to ease the use of this toolbox and allow to rapidly test the methods and display the results.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Karim Cherifi,

The institute of electrical and electronic engineering,

M'hamed Bougara University,

Independence Avenue, 35000, Boumerdès, Algeria.

Email: cherifikarimd@gmail.com

1. INTRODUCTION

Minimal realization and model reduction [1, 2] are widely used to obtain mathematical representations of systems in state space form with the smallest possible number of states [3-7].

Many definitions have been given to realizations but generally speaking, a realization is the process of transforming exactly a system from its frequency based transfer function $G(s)$ to a time based state space representation (A, B, C, D) :

$$\begin{cases} \dot{X} = A X + B U \\ Y = C X + D U \end{cases} \quad (1)$$

Where U is the input vector regrouping m inputs. Y is the output vector regrouping with p inputs. X is the state vector with dimension n . A is an $n \times n$ state matrix. B is an $n \times m$ input matrix. C is a $p \times n$ output matrix. D is a $p \times m$ feedforward matrix.

A minimal realization is a realization with a minimal order n called the McMillian degree. All minimal realizations are equivalent [3]. It is worth noting that a state space representation is a minimal realization of a proper rational function $G(s)$ if and only if (A, B) is controllable and (A, C) is observable [3].

Beside time reduction in controller and observer designs [4, 5], minimal realizations in general have the advantage of reducing the cost of implementation of the state space equations in terms of electronic components. Different techniques exist for minimal realization. Each technique has its advantages and its disadvantages. Some techniques focused on obtaining the state space representation in a minimal time and other focused on the form of the matrices of the state space representation. Depending on the application, time efficiency or configuration of the resulting minimal realization may be preferred. This requires to have

tool that would allow researchers and engineers to easily compute minimal realization. Up to now such a complete tool could not be found in the literature.

Many toolboxes have been implemented to help researchers and engineers in their work [8-14]. This MATLAB toolbox for minimal realization can be combined to other tools to design controllers and observers in an efficient way. Although a toolbox for minimal realization was presented in the past [15], it is incomplete as it presents only one method and is not user friendly as it does not have a graphical user interface (GUI).

In this paper, first, the implemented minimal realization techniques are presented theoretically, then, the graphical user interface (GUI) is presented along with individual functions corresponding to the different minimal realization techniques. Finally, the GUI is tested using practical applications and different techniques are compared using the minimal realization toolbox.

In the broad sense, realizations aim to obtain systems in state space representations even for raw input-output data. One of the most promising methods for data from frequency domain and time domain, are the methods based on Lowener framework [6, 7]. These are also presented and implemented in this paper.

The presented toolbox is valid for single input single output (SISO) systems as well as multi input multi output (MIMO) systems. It is also valid for continuous time and discrete time systems.

2. PROPOSED PROCEDURES

There exist many minimal realization techniques [16-18]. In this paper we distinguish two main categories: realization from a transfer function and realizations from data. In this section, we present the theoretical basis for the methods implemented in the toolbox.

In 1963, Gilbert presented a method based on the partial fraction of the transfer function [19]. Although this method is interesting in terms of the resulting configuration, it is not general because it is not applicable to repeated poles. Thus, this method will not be detailed here. Another popular method is the Kalman decomposition method. The goal of this method is to obtain a minimal realization given a non-minimal realization. Other methods based on the Hankel matrix were presented in the following years. We present here two methods based on the Hankel matrix: the Singular value decomposition (SVD) method and the row searching method. Finally, methods based on coprime fractions of the transfer function are presented.

Concerning the realizations from data, we present two methods based on the Loewner matrix. One method is a realization from frequency domain data while the other one is a realization from time domain data.

2.1. Kalman Decomposition

The Kalman decomposition method [20] is an indirect method which requires to have first a non-minimal realization to be able to compute the minimal realization. Once a realization is obtained, Kalman decomposition is used to separate the system into four parts:

- A controllable and observable part (c-o).
- A non-controllable and observable part (nc-o).
- A controllable and non-observable part (c-no).
- A non-controllable and non-observable part (nc-no).

Kalman decomposition is obtained using a similarity transformation. After Kalman decomposition, the obtained system is in the form:

$$\hat{A} = \begin{bmatrix} A_{c-no} & A_{12} & A_{13} & A_{14} \\ 0 & A_{c-o} & 0 & A_{24} \\ 0 & 0 & A_{nc-no} & A_{34} \\ 0 & 0 & 0 & A_{nc-o} \end{bmatrix} \hat{B} = \begin{bmatrix} B_{c-no} \\ B_{c-o} \\ 0 \\ 0 \end{bmatrix} \hat{C} = [0 \quad C_{c-o} \quad 0 \quad C_{nc-o}] \quad \hat{D} = D \quad (2)$$

Since by definition a minimal realization is controllable and observable, only that part is kept. Using the form (2), the minimal realization is given by (3):

$$\begin{cases} \dot{X} = A_{c-o} X + B_{c-o} U \\ Y = C_{c-o} X + D U \end{cases} \quad (3)$$

The main advantage of this form is the ability to use it when the given system is already in state space form but it is not of minimal order.

2.2. Hankel Matrix and SVD

The Hankel matrix is defined as:

$$T = \begin{bmatrix} H(1) & H(2) & \dots & H(r) \\ H(2) & H(3) & \dots & H(r+1) \\ \vdots & \vdots & \ddots & \vdots \\ H(r-1) & H(r) & \dots & H(2r-2) \\ H(r) & H(r+1) & \dots & H(2r-1) \end{bmatrix} \tag{4}$$

where H(i) are Markov parameters which can be computed from the transfer function as:

$$G(s) = H(0) + H(1)s^{-1} + H(2)s^{-2} + \dots \tag{5}$$

For SISO systems H(i) is just a scalar.

Theorem 1: A strictly proper rational function G(s) has degree n if and only if rank{T(n,n)}=rank{T(n+k,n+1)} for every k, l=1,2,.....

Proof of theorem 1: [3].

There exist many methods based on the Hankel matrix. We present here two methods: one based on the SVD and the other one on row searching.

In the first method, the singular value decomposition of the Hankel matrix (4) is computed:

$$T = K \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} L^T \tag{6}$$

By definition, K and L are orthogonal and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ where λ_i are the square roots of the positive eigenvalues of $T^T T$. The rank of T can be deduced from theorem 1.

Let \tilde{K} denote the first n columns of K and \tilde{L}^T denote the first n rows of L^T . Then T can be decomposed as:

$$T = \tilde{K} \tilde{A} \tilde{L}^T = \tilde{K} A^{1/2} A^{1/2} \tilde{L}^T := O \zeta \tag{7}$$

where $O = \tilde{K} A^{1/2}$ and $\zeta = A^{1/2} \tilde{L}^T$. ζ is the controllability matrix and is O the observability matrix.

The resulting minimal realization is given by:

$$A = O^+ \tilde{T} \zeta^+ \tag{8}$$

$$B = \text{First p columns of } \zeta \tag{9}$$

$$C = \text{First q columns of } O \tag{10}$$

$$D = H(0) \tag{11}$$

This minimal realization is also called balanced realization because the controllability matrix ζ and the observability matrix O satisfy:

$$\zeta^+ \zeta = O^+ O \tag{12}$$

A modified version of this method is used in model reduction.

2.3. Hankel Matrix and Row Searching

This method consists of searching for linearly dependent rows (columns) in the Hankel matrix. Because of the structure of the Hankel matrix and theorem 1, if one row in a block row is linearly dependent on its previous rows, then it is also linearly dependent on the subsequent rows in the same block row.

Let σ_i be the number of linearly independent rows in a block row i where $i=1, 2, \dots, a$. Each primary linearly dependent row in a given block row will be written as a linear combination of its previous block rows. A primary linearly dependent row is the first linearly dependent row in a given block row. At the end, the outcome of the row searching algorithm will be the coefficients corresponding to the q primary linearly dependent rows:

$$\begin{aligned}
 k_1 &= [a_{11}(1) \dots a_{11}(\sigma_1) 1 : 0 \dots 0 0 \dots 0 : \dots : 0 \dots 0 0 \dots 0] \\
 k_2 &= [a_{21}(1) \dots a_{21}(\sigma_1) 0 : a_{22}(1) \dots a_{22}(\sigma_2) 1 \dots 0 : \dots : 0 \dots 0 0 \dots 0] \\
 &\vdots \\
 k_q &= [a_{q1}(1) \dots a_{q1}(\sigma_1) 0 : a_{q2}(1) \dots a_{q2}(\sigma_2) 0 \dots 0 : \dots : a_{qq}(1) \dots a_{qq}(\sigma_q) 1 \dots 0]
 \end{aligned}
 \tag{13}$$

Such that $k_i T = 0$.

A minimal realization is given by:

$$A = \begin{bmatrix} A_{11} & 0 & \dots & 0 \\ A_{21} & A_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{q1} & A_{q2} & \dots & A_{qq} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} \quad D = \begin{bmatrix} h_1(0) \\ h_2(0) \\ \vdots \\ h_q(0) \end{bmatrix}
 \tag{14}$$

$$\text{for } i = j \quad A_{ij} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_{ij}(1) & -a_{ij}(2) & \dots & -a_{ij}(\sigma_i) \end{bmatrix}, \quad \text{for } i \neq j \quad A_{ij} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ -a_{ij}(1) & -a_{ij}(2) & \dots & -a_{ij}(\sigma_j) \end{bmatrix}
 \tag{15}$$

$$b_i = \begin{bmatrix} h_{i1}(1) \\ h_{i1}(2) \\ \vdots \\ h_{i1}(\sigma_i) \end{bmatrix}
 \tag{16}$$

For all i .

The matrix C is computed as:

$$C = \left[\begin{array}{cccc} \overbrace{1 \ 0 \ \dots \ 0}^{\sigma_1} & \overbrace{0 \ 0 \ \dots \ 0}^{\sigma_2} & \overbrace{0 \ 0 \ \dots \ 0}^{\sigma_q} & \\ 0 \ 0 \ \dots \ 0 & 1 \ 0 \ \dots \ 0 & 0 \ 0 \ \dots \ 0 & \\ \vdots \ \vdots \ \ddots \ \vdots & \vdots \ \vdots \ \ddots \ \vdots & \dots \ \vdots \ \vdots \ \ddots \ \vdots & \\ 0 \ 0 \ \dots \ 0 & 0 \ 0 \ \dots \ 0 & 1 \ 0 \ \dots \ 0 & \end{array} \right] \Bigg\} q
 \tag{17}$$

for $i=1, 2, \dots, q$, and if $\sigma_i \neq 0$.

If $\sigma_i = 0$ for some i then the i th row in the $(i-1)$ first block columns will be constituted from $-a_{ij}(v)$ where j is the number of the block column and $v = 1, 2, \dots, \sigma_j$.

2.4. Coprime Fractions

Given a coprime right fraction $G(s)$:

$$G(s) = N_R(s) D_R(s)^{-1} \tag{18}$$

If the fraction is column reduced then $D(s)$ and $N(s)$ can be written as:

$$D(s) = D_{hc} H(s) + D_{lc} L(s) \tag{19}$$

$$N(s) = N_{lc} L(s) \tag{20}$$

where

$$H(s) = \begin{bmatrix} s^{\mu_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & s^{\mu_p} \end{bmatrix} L(s) = \begin{bmatrix} 1 \\ s \\ \vdots \\ s^{\mu_1-1} \\ \vdots \\ \vdots \\ 1 \\ s \\ \vdots \\ s^{\mu_p-1} \end{bmatrix} \tag{21}$$

D_{hc} , D_{lc} and N_{lc} are constant matrices. μ_i Is the column degree of $D(s)$ for $i=1, 2, \dots, p$. It was proven [21] that if a system is controllable and observable then:

$$(\lambda I_n - A)^{-1} B = L(s) [D_{rh} \delta_r]^{-1} \tag{22}$$

$$B_r = D_{hc}^{-1} \tag{23}$$

And by definition:

$$G(s) = C(\lambda I_n - A)^{-1} B = N_R(s) D_R(s)^{-1} \tag{24}$$

Replacing (22) and (23) results in:

$$D_R(s) = D_{rh} \delta_r \tag{25}$$

$$N_R(s) = CL(s) \tag{26}$$

Where:

$$\delta_r = H(s) - A_r L(s) \tag{27}$$

Replacing (24) in (25):

$$D_R(s) = D_{hc} H(s) - D_{hc} A_r L(s) \tag{28}$$

This results in:

$$A_r = -D_{hc}^{-1}D_{lc}, \quad B_r = D_{hc}^{-1}, \quad C = N_{lc} \quad (29)$$

A_r and B_r are the nontrivial rows of A and B.

The resulting minimal realization is in controller canonical form:

- a. A is in general controller canonical form where

$$\begin{cases} \mu_1^{th} \text{ row of A is the } 1^{st} \text{ row of } -D_{hc}^{-1}D_{lc} \\ (\mu_1 + \mu_2)^{th} \text{ row of A is the } 2^{nd} \text{ row of } -D_{hc}^{-1}D_{lc} \\ \vdots \end{cases}$$

- b. B is in general controller canonical form where

$$\begin{cases} \mu_1^{th} \text{ row of B is the } 1^{st} \text{ row of } D_{hc}^{-1} \\ (\mu_1 + \mu_2)^{th} \text{ row of B is the } 2^{nd} \text{ row of } D_{hc}^{-1} \\ \vdots \end{cases}$$

- c. $C = N_{lc}$

The controller canonical form was obtained from a right coprime fraction. This is particularly interesting for controller design using pole placement [22]. If a left coprime fraction is used, the result is a minimal realization in observer canonical form. Both methods were implemented in the minimal realization toolbox.

2.5. Realization from Frequency Domain Data:

The first results for realizations based on frequency domain data using the Loewner matrix were presented in [6]. In this subsection we present the main results and a brief explanation of the procedure used to construct the numerical minimal realization. The resulting system is a descriptor system or also known as differential algebraic equations (DAE). It has the form:

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

If E is invertible, the standard state space form can be easily recovered. If E is singular some techniques exist to transform it to a standard state space provided that some conditions are met. More details can be found in [23-26].

Given right frequency data: $(\lambda_i; r_i; w_i), i = 1, \dots, k$ and left frequency data $(\mu_j; \ell_j^*; v_j^*), j = 1, \dots, q$. The goal is to find a transfer function H(s) that interpolates this data such that:

$$H(\lambda_i)r_i = w_i, \quad \ell_j^* H(\mu_j) = v_j^* \quad (30)$$

Then one can regroup the data in matrices such that for right data:

$$\begin{aligned} \Lambda &= \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{pmatrix} \in \mathbb{C}^{k \times k}, \\ R &= [r_1 \quad r_2 \quad \dots \quad r_k] \in \mathbb{C}^{m \times k} \\ W &= [w_1 \quad w_2 \quad \dots \quad w_k] \in \mathbb{C}^{p \times k} \end{aligned} \quad (31)$$

and for left data:

$$M = \begin{pmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_q \end{pmatrix} \in \mathbb{C}^{q \times q}, L = \begin{bmatrix} l^*_1 \\ \vdots \\ l^*_q \end{bmatrix} \in \mathbb{C}^{q \times p}, V = \begin{bmatrix} v^*_1 \\ \vdots \\ v^*_q \end{bmatrix} \in \mathbb{C}^{q \times m} \tag{32}$$

Then from this data, one can construct the well known Loewner matrix $L \in \mathbb{C}^{q \times k}$ as follow:

$$L = \begin{bmatrix} \frac{v_1 r_1 - l_1 w_1}{\mu_1 - \lambda_1} & \dots & \frac{v_1 r_k - l_1 w_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{v_q r_1 - l_q w_1}{\mu_q - \lambda_1} & \dots & \frac{v_q r_k - l_q w_k}{\mu_q - \lambda_k} \end{bmatrix} \tag{33}$$

This L matrix satisfies the Sylvester equation:

$$L\Lambda - ML = VR - LW \tag{34}$$

Then one also needs to construct the shifted Loewner matrix $L_s \in \mathbb{C}^{q \times k}$ such that:

$$L_s = \begin{bmatrix} \frac{\mu_1 v_1 r_1 - l_1 w_1 \lambda_1}{\mu_1 - \lambda_1} & \dots & \frac{\mu_1 v_1 r_k - l_1 w_k \lambda_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q v_q r_1 - l_q w_1 \lambda_1}{\mu_q - \lambda_1} & \dots & \frac{\mu_q v_q r_k - l_q w_k \lambda_k}{\mu_q - \lambda_k} \end{bmatrix} \tag{35}$$

This L_s matrix satisfies the Sylvester equation:

$$L_s \Lambda - ML_s = MVR - LW\Lambda \tag{36}$$

Then one can construct a realization:

$$E = -L, A = -L_s, B = V, C = W, D = 0 \tag{37}$$

If the numerical minimal realization is to be computed and the numerical $rank L = k$, compute the rank revealing SVD:

$$L = Y \Sigma X^* \approx Y_k \Sigma_k X_k^* \tag{38}$$

Then the minimal realization is given by:

$$E = -Y_k^* L X_k, A = -Y_k^* L_s X_k, B = Y_k^* V, C = W X_k, D = 0 \tag{39}$$

The resulting system may be complex. A procedure was developed to recover the underlying real system. This procedure and the proofs for the results given above can be found in [27].

2.6. Realization from Time Domain Data:

The goal in this procedure is to interpolate the time domain input-output data by a descriptor system:

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (40)$$

To achieve this objective, a modified Loewner can be used [7]. This algorithm constructs a model for an LTI system directly from a single time domain input/output snapshot without prior knowledge of frequency response data. Of course, the more knowledge about the characteristics of the system, the more accurate the model will be.

The input u and the output y of the system are measured with respect to time and recorded as vectors u and y with length K which corresponds to the number of samples taken. K_{min} is selected as a quarter of K such that to ensure that the outputs have entered a steady state after k_{min} . n is an integer corresponding to the dimension of the model.

The choice of the input u , the interpolation points and the number of samples K can be changed to best correspond to the system at hand including which range of frequencies is of importance. Specifically, the range of the possible interpolation frequencies that can be chosen is:

$$\left[\frac{2\pi}{K}, \frac{2\pi(K-1)}{K} \right] \subset \mathbb{R} \quad (41)$$

Therefore, increasing the number of samples K , increases the range of possible frequencies.

We used a modified version of that algorithm in our implementation to meet our requirements. For more details on the algorithm itself and its proof, please refer to [7].

3. RESEARCH METHOD

There exist different algorithms each with its advantages and drawbacks. The best method depends on each specific application. This requires having a tool to be able to compare between the different methods and choose the most appropriate method for that specific application. This tool (or commonly called toolbox) is the minimal realization toolbox implemented in MATLAB and presented in this paper. MATLAB was chosen because it is used by a lot of researchers and engineers [28, 29]. All the methods presented in Section 2 were implemented in functions: Gilbert Method, Kalman decomposition, Hankel and SVD, Hankel and row searching, Left coprime fraction (LCF), Right coprime fraction (RCF), Interferq (from frequency domain data), Intertime (from time domain data).

All the programmed functions are grouped in a toolbox for minimal realization. This toolbox can be easily used using the graphical user interface (GUI) shown in Figure 1.

The GUI is divided into two main sections: The first one on the top left is the “interpolation and minimal realization” section where one can obtain a minimal realization from frequency or time domain data by pressing the appropriate button. In the second section, one can compute a minimal realization from a transfer function (TF). Six buttons corresponding to six minimal realization methods are available. Choosing a specific method will display the result of that method directly on the same view (Figure 1). Choosing another method will display the results of that method discarding the previous results. Finally on the top right, in the options section, one can load data, save the results to a matlab file or compare the different minimal realization methods available with a bode plot. Under these sections, we have five text fields displaying the matrices of the minimal realization. In the next section, the GUI is used to compute the minimal realization of practical problems.

4. RESULTS AND DISCUSSION

In order to test the methods for the minimal realization from a transfer function, a distillation column system is used [30]. The matrix transfer function is as follow:

$$G(s) = \begin{bmatrix} \frac{2.581s - 6.746}{s^3 + 4s^2 + 5s} & \frac{5.949s + 26.64}{s^3 + 4s^2 + 5s} \\ \frac{4.087s + 59.48}{s^3 + 4s^2 + 5s} & \frac{2.622s - 54.94}{s^3 + 4s^2 + 5s} \end{bmatrix} \quad (42)$$

Entering this matrix transfer function into the GUI and choosing for example left coprime fraction results in the minimal state space realization shown in Figure 1. This can be done for all the other methods by selecting the appropriate button.

Using the button “Methods’ comparison”, the methods can be compared in terms of their Bode plot where the transfer function is plotted in dots and the other methods as continuous line. For the example used, the resulting Bode plot is shown in Figure 2.

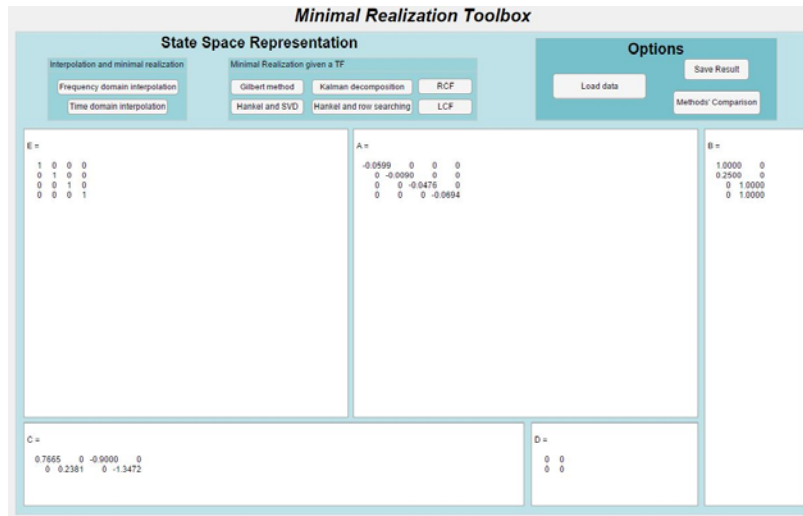


Figure 1. Minimal realization of a distillation column using the presented GUI

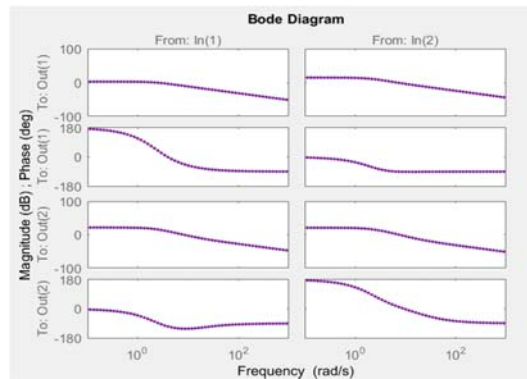


Figure 2. Bode plot comparison of the TF (dotted blue) and state space form (line red)

The previous example allowed us to test the capabilities of the GUI. After few clicks, a minimal realization can be obtained using the chosen minimal realization method. This also permitted to determine the McMillian degree. Since the size of the matrix A is clearly 4 then the McMillian degree is 4. Also, based on the comparison of the methods, the Bode plot of the original system perfectly superposes with the resulting methods. This proves that these minimal realizations are indeed exact.

Since the algorithms have similar time complexities between ranging between $O(n^3)$ and $O(n^4)$, the engineer or the researcher will generally choose the method based on the usefulness of the configuration of the state space representation.

In order to test the Loewner method given frequency domain data, we use the beam example of order 348 from [31]. The Loewner method allows to choose the numerical minimal order based on the decay of singular values. In order to capture the behaviour of the system, the order is chosen to be 14. The results are shown in Figure 3. The original and resulting bode plots are depicted in Figure 5.

Finally in order to test the time domain Loewner method, we consider Penzl's time-continuous LTI system introduced in [32], which is investigated in the context of the Loewner approach. The data given is in form of input-output time domain data. Using the the button “time domain interpolation”. We obtain the system shown in Figure 4. A comparison between the bode plots of the original system and the interpolated system in descriptor form are shown in Figure 5.

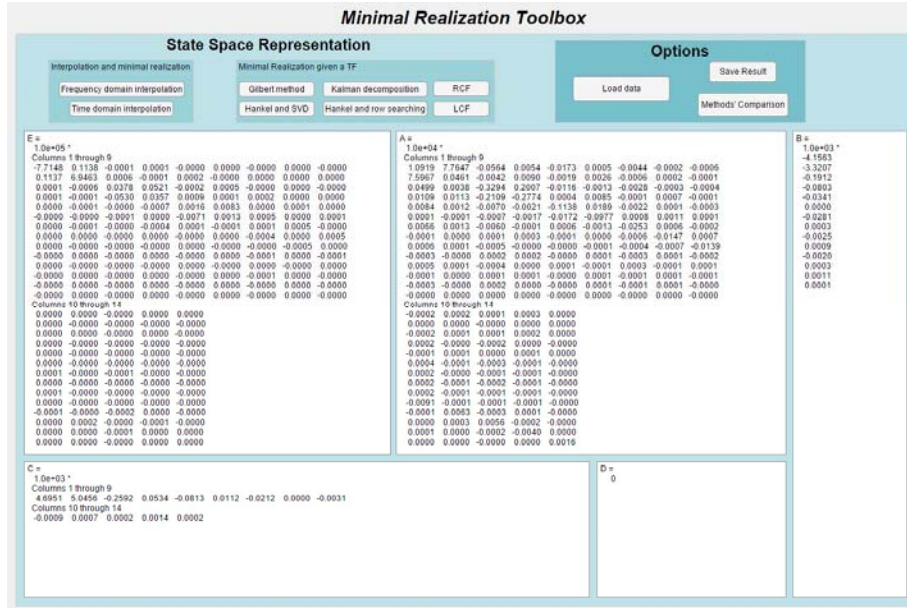


Figure 3. Result of the interpolation of the beam example as shown in the GUI

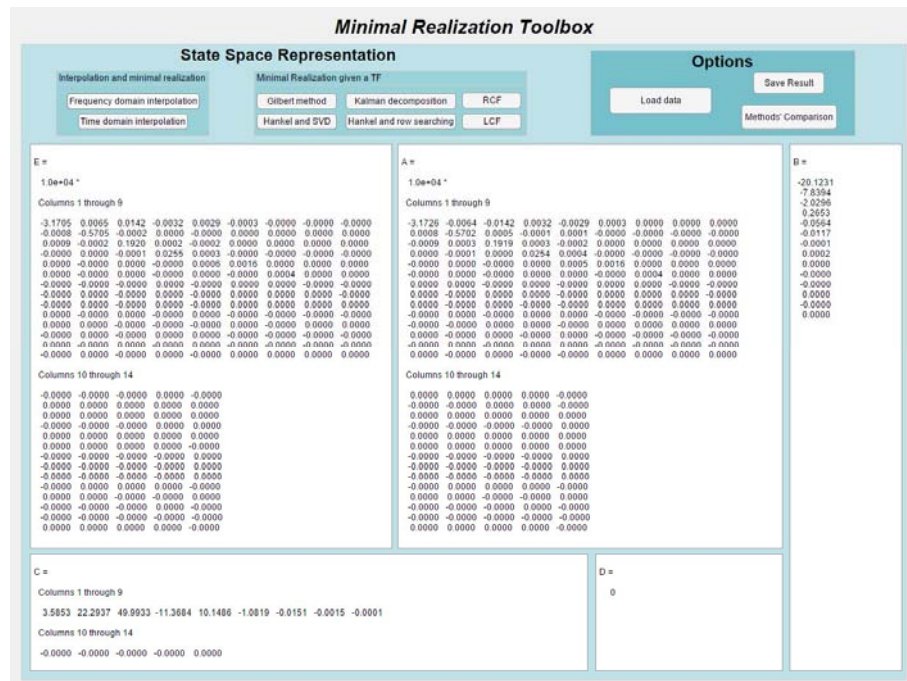


Figure 4. Result of the Penzl example's interpolation as shown in the GUI

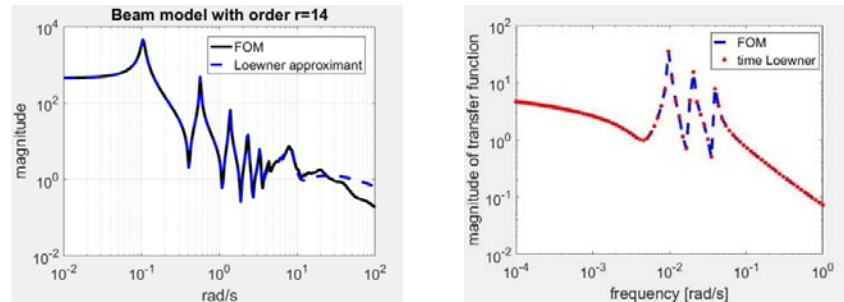


Figure 5. Bode plot comparison for the beam example (left) and Penzl example (right)

5. CONCLUSION

A new Minimal realization MATLAB toolbox was implemented. Minimal realization is an important tool used to compute the state space representation of a system with a minimal number of states. The presented GUI and toolbox allows to compute and compare between different minimal realizations techniques for linear systems. Transforming the system to state space allows to use the rich theory available for systems in state space representation especially controller and observer design using pole placement.

This toolbox works for both SISO and MIMO systems. The presented toolbox can be enhanced by incorporating controller and observer design algorithms. Some of these algorithms are already available as functions in MATLAB.

REFERENCES

- [1] C. B. Vishwakarma, "Modified hankel matrix approach for model order reduction in time domain," *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, vol. 8, pp. 7, 2014.
- [2] Y. Shamash., "Model reduction using minimal realisation algorithms," *Electronics Letters*, vol. 11(16), pp. 385-387, 1975, Available: http://digital-library.theiet.org/content/journals/10.1049/el_19750293
- [3] C.-T. Chen, "Linear System Theory and Design," Oxford University Press, Inc., 1995.
- [4] E. C. Abana, M. Pacion, R. Sordilla, D. Montaner, D. Agpaoa, and R. M. Allam, "Rakebot: a robotic rake for mixing paddy in sun drying," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 14, 2019.
- [5] R. Pilla, S. Killari, and K. B. M. Sahu, "Design and simulation of the control system for inverter-fed permanent magnet synchronous motor drive," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 12, 2018.
- [6] A. J. Mayo and A. C. Antoulas, "A framework for the solution of the generalized realization problem," *Linear algebra and its applications*, vol. 425, pp. 634-662, 2007.
- [7] B. Peherstorfer, S. Gugercin, and K. Willcox, "Data-driven reduced model construction with time-domain loewner models," *SIAM Journal on Scientific Computing*, vol. 39, pp. A2152-A2178, 2017.
- [8] Y. Yuan, A. Rai, E. Yeung, G.-B. Stan, S. Warnick, and J. Goncalves, "A minimal realization technique for the dynamical structure function of a class of LTI systems," *IEEE Transactions on Control of Network Systems*, pp. 1-1, 2015.
- [9] A. V. Evseev and V. N. Chetverikov, "Realizations and minimal realizations of input-output mappings of general form," *Differential Equations*, vol. 49, pp. 1609-1618, 2014.
- [10] C. W. Scherer and İ. E. Köse, "From transfer matrices to realizations: Convergence properties and parametrization of robustness analysis conditions," *Systems & Control Letters*, vol. 62, pp. 632-642, 2013.
- [11] P. Corke, "Robot manipulator capability in MATLAB: A tutorial on using the robotics system toolbox [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 24, pp. 165-166, 2017.
- [12] H. Qin, Z. Liu, Y. Liu, and H. Zhong, "An object-oriented MATLAB toolbox for automotive body conceptual design using distributed parallel optimization," *Advances in Engineering Software*, vol. 106, pp. 19-32, 2017.
- [13] J. R. Raj, S. M. K. Rahman, and S. Anand, "Microcontroller USB interfacing with MATLAB GUI for low cost medical ultrasound scanners," *Engineering Science and Technology, an International Journal*, vol. 19, pp. 964-969, 2016.
- [14] B. S. Ahmed, M. A. Sahib, and M. Y. Potrus, "Generating combinatorial test cases using Simplified Swarm Optimization (SSO) algorithm for automated GUI functional testing," *Engineering Science and Technology, an International Journal*, vol. 17, pp. 218-226, 2014.
- [15] X.-D. Sun, T. Clarke, and M. A. Maiza., "A toolbox for minimal state space model realisation," *IEE Proceedings-Control Theory and Applications*, vol. 143(2), pp. 152-158, 1996 Available: http://digital-library.theiet.org/content/journals/10.1049/ip-cta_19960360
- [16] B. D. Schutter, "Minimal state-space realization in linear system theory: An overview," *Journal of Computational and Applied Mathematics*, pp. 24, 2000.

- [17] K. Cherifi and K. Hariche, "Review of MIMO minimal realization techniques and case study on SCARA robot manipulator," *International review of automatic control*, vol. 10, 2017.
- [18] Y. Yuan, A. Rai, E. Yeung, G. B. Stan, S. Warnick, and J. Gonçalves, "A minimal realization technique for the dynamical structure function of a class of LTI systems," *IEEE Transactions on Control of Network Systems*, vol. 4, pp. 301-311, 2017.
- [19] E.G.Gilbert, "Controllability and observability in multi-variable control systems," *SIAM J. Control* vol. 1, pp. 128-151, 1963.
- [20] R.E.Kalman, "Irreducible realizations and the degree of a rational matrix," *J. Soc. Ind. Appl. Math*, vol. 13, pp. 24, 1965.
- [21] S. Tsay, Barnett, "Structural analysis and design of multivariable control systems : An algebraic approach: Berlin," New York : Springer-Verlag, 1988.
- [22] M. Sababha, M. Zohdy, and M. Kafafy, "Robust pole placement using firefly algorithm," *International Journal Of Electrical and Computer Engineering (IJECE)*, vol. 9, 2019.
- [23] P. Benner, P. Losse, V. Mehrmann, and M. Voigt, "Numerical linear Algebra Methods for linear Differential-Algebraic equations," in *Surveys in Differential-Algebraic Equations III*, A. Ilchmann and T. Reis, Eds., ed Cham: Springer International Publishing, pp. 117-175, 2015.
- [24] S. Campbell, P. Kunkel, and V. Mehrmann, "Chapter 2: Regularization of linear and nonlinear descriptor systems," in *Control and Optimization with Differential-Algebraic Constraints*, ed: Society for Industrial and Applied Mathematics, pp. 17-36, 2012.
- [25] R. K. H. Galvão, K. H. Kienitz, and S. Hadjiloucas, "Conversion of descriptor representations to state-space form: an extension of the shuffle algorithm," *International Journal of Control*, pp. 1-15, 2017.
- [26] J. Sjöberg, "Descriptor systems and control theory," Linköpings universitet, Department of Electrical Engineering, 2005.
- [27] A. Antoulas, S. Lefteriu, and A. Ionita, "Chapter 8: A tutorial introduction to the loewner framework for model reduction," in *Model Reduction and Approximation*, ed: Society for Industrial and Applied Mathematics, pp. 335-376, 2017.
- [28] A. Parvaresh and S. M. A. Mohammadi, "A Multi-Zone HVAC system for a typical building for MATLAB/SIMULINK platform," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 2, pp. 83-87, 2013.
- [29] A. Akinrinde, A. Swanson, and R. Tiako, "Effect of ferroresonance on wind turbine: Comparison of Atp/Emp and Matlab/Simulink," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 14, 2019.
- [30] Q. G. Wang, "Decoupling control," *Springer*, Berlin Heidelberg, 2003.
- [31] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga, "SLICOT-a subroutine library in systems and control theory," in *Applied and Computational Control, Signal and Circuits*. vol. 1, pp. 499-539, 1999.
- [32] T. Penzl, "Algorithms for model reduction of large dynamical systems," *Linear algebra and its applications*, vol. 415, pp. 322-343, 2006.

BIOGRAPHIES OF AUTHORS



Karim Cherifi is a PhD student in the institute of electrical and electronic engineering of the University of Boumerdes, Algeria. He received from the same institute a B.S. degree in 2013 and a master degree in 2015. He has received several distinctions and grants from SIAM and DAAD. He recently finished a research stay with a research grant from DAAD in Max Planck institute, Magdeburg, Germany. His research interests include control theory and engineering, minimal realization, model reduction and robotics.



Kamel Hariche received a B.S. degree in physics from the University of Algiers, Algeria (1973), a master's degree in 1978 and a PhD degree in electrical engineering from the University of Houston (Texas-U SA). Currently, he is a professor in electrical engineering at the institute of electrical and electronic engineering of the University of Boumerdes, Algeria. He wrote numerous papers in control systems and applied mathematics.