International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 1, February 2021, pp. 272~283
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i1.pp272-283

Fully symbolic-based technique for solving complex state-space control systems

Amera M. Abd-Alrahem[1], Hala M. Elhadidy[2], Kamel A. Elserafi[3], Hassen T. Dorrah[4]
[1,2,3]Department of Electrical Engineering, Port-Said University, Egypt
[2]Department of Computer Engineering and Networks, Jouf University, Kingdom of Saudi Arabia
[4]Department of Electrical Engineering, Cairo University, Egypt

Article Info

Article history:

Received Dec 24, 2019
Revised Jun 7, 2020
Accepted Aug 5, 2020

Keywords:

Liquid container
Partition matrix theory
Ship autopilot
State-space solution
Symbolic algebra
Wind turbine

ABSTRACT

Despite the superiority of symbolic approaches over the purely numerical approaches in many aspects, it does not receive the proper attention due to its significant complexity, high resources requirement and long drawn time which even grows significantly with the increase of model dimensions. However, its merits deserve every attempt to overcome the difficulties being faced. In this paper, a fully generic symbolic-based technique is proposed to deal with complex state space control problems. In this technique, depending on the model dimension if exceeds a predefined limit, the state space is solved using the partitioned matrices theory and block wise inversion formula. Experimental results demonstrate that the proposed technique overcomes all the previously mentioned barriers and gives the same results when compared to numerical methods (Simulink). Moreover, it can be used to gain useful information about the system itself, provides an indication of which parameters are more important and reveals the sensitivity of system model to single parameter variations.

Corresponding Author:

Amera M. Abd-Alrahem,
Department of Electrical Engineering,
Port-Said University,
Faculty of Engineering, Port-Foad 42526, Egypt.
Email: amera.gamea@eng.psu.edu.eg

1. INTRODUCTION

Symbolic algebra is a generic technique exploited to produce symbolic expressions for any system regarding its parameters and variables. In a field like control systems where complex mathematical formulas are commonly used, generic solution can be extremely valuable. Using symbolic manipulation and simplifications, the control engineer can obtain simple formulas for a complex looking problem. In general, it is conceivable to express that using symbolic techniques brings two fundamental advantages among numerous others: using exact calculations and direct manipulation of symbols [1].

Numerical technique is based on analyzing the system for several sample points and it is assumed that the results also hold for all the values among them [2]. It is considered to be more approximate solutions compared to symbolic solutions. Although it is usually much faster, it shows some weaknesses especially with parameters varying systems. It may lead to imprecise conclusions, the variables that actually matter may be difficult to agree upon, be measured, or control [3]. Therefore, this causes loads of researches where each proposes different set of values assumptions to be the perfect choice for the problem under investigation.

Although theoretically symbolic analysis can be considered as a good complement to numerical analysis, it is still facing some difficulties such as the cost of hardware implementation measured by

computational and communication requirements. Also, there is a problem of increasing complexity of modern control problems to be dealt with while the symbolic-based assistant programming is still under development. It is believed that the advantages are much more important than the difficulties which make it worth all our efforts to defeat the difficulties being confronted.

Historically, the use of symbolic based techniques witness periods of rise and decay. Although the early ideas of symbolic computations arise in the 1960s, but attention was turned by the time toward numerical techniques as systems were getting more complex. Then, it can be considered that the real leap for exploiting symbolic algebra in control engineering applications was in the late nineties by Professor Neil Munro. He gathered many valuable applications of symbolic algebra to control theory [4], which drew intense attention to the use of symbolic techniques as a new powerful tool for solving many problems in control systems engineering. This led to the development of many symbolic toolboxes in control system design [5, 6].

In recent years, as the cyber-physical systems increase, researchers return again to focus on symbolic algebra in control applications including fields like system modelling, analysis and controller design [7]. Regarding the system modelling, Pola *et al*. [8] derived a symbolic model as an approximation for the plant and then exploited to obtain the control problem solution. The results were utilized in the symbolic control of a car-like robot. Fakhroleslam *et al*. [9] constructed approximately bisimilar symbolic models for nonlinear control systems. It shows how the chemical process control can be managed via the usage of methods that is based on symbolic models. Girard *et al*. [10] proposed multiscale symbolic models construction for switched systems. It was demonstrated that these models are approximately bisimilar to the genuine systems. Then, it was used for controller computation and applied to the safety automata. Radaydeh and Mothafar [11] developed a symbolic state-space-based model for current-mode controlled modular DC-DC converters. Then the control-to-output voltage transfer functions are derived in a symbolic form.

Regarding controller design, Mizoguchi and Ushio [12] designed a deadlock-free symbolic output feedback controller. Borri *et al*. [13] proposed a symbolic method to the control design of NCS, taking into account the most significant non-idealities in the communication channel with application to robot motion planning. Shamgah *et al*. [14] developed a novel symbolic technique for controlling autonomous attacking vehicles in feuding environments. Dorrah *et al*. [15] proposed a new generic methodology for derivation and realization of feedback gains for automatic control systems based on symbolic representation of feedback stabilization functions.

Regarding the system analysis, Mladenović *et al*. [16] analyzed the expectation-maximization algorithm via symbolic processing and computer algebra tools. Then, the obtained results are employed for further optimization. Setiawan *et al*. [17] proposed a steady state symbolic analysis for buck converter which gives the exact calculation of steady state output with no need for transient response evaluation. Abd-Alrahem *et al*. [2] examined the problem of fully symbolic representation and analysis for parameter varying systems (PVS). The simulation results were carried out on the drug concentration pharmacokinetics and proportional-derivative action in the retina. The generated output was reported to be neater than and as close as algebraic, and the effect of parameter variation on state variables was analyzed. Nevertheless, it is inconvenient for high order state space models, as it suffers from high computational complexity.

Therefore, in this paper, a fully symbolic-based technique is proposed for solving n^{th} order state-space models and thus utilized in the analysis of parameter variations. The proposed technique has the ability to deal with large complex systems through partitioning the original matrices, then each sub-matrix is treated separately with the appropriate methods and assembled eventually altogether. Hence, speeding up the process and reduce the computational complexity problem found in [2].

The rest of this paper is organized as follows: The conventional criterion for solving state space model is formulated in Section 2. The proposed symbolic-based technique via partition matrix theory for solving state models is described in Section 3. The simulation results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. PROBLEM STATEMENT

As modern systems become inherently complex, the state space representation of a dynamic system replaces an n^{th} order differential equation with a single first order matrix differential equation as follows [18]:

$$\dot{x} = A\,x + B\,u \tag{1}$$

$$y = C\,x + D\,u \tag{2}$$

where (1) is called the state equation and (2) is called the output equation. u is the input function and *y* is the output function. *A(n×n), B(n×p), C(q× n), D*(*q*× *p*) represents the state matrix, the input matrix, the output matrix and the feed through matrix respectively.

The complete system state response x(t) consists of two parts: a zero-input response X_{ZI} that represents the response due to arbitrary initial conditions x(0) with no inputs at all, and a zero state response X_{ZS} that represents the response due to arbitrary input with zero initial conditions. The two parts are then joined to produce the complete response as follows,

$$X_{ZI} = \Phi(t).\ x(0) \tag{3}$$

$$X_{ZS} = \int_0^t \Phi(t-\tau) B\ U(\tau)\ d\tau \tag{4}$$

$$X(t) = \Phi(t).\ x(0) + \int_0^t \Phi(t-\tau) B\ U(\tau)\ d\tau \tag{5}$$

where Φ(t) is identified to be the state transition matrix, which is computed by:

$$\Phi(t) = L^{-1}[SI - A]^{-1} \tag{6}$$

According to [2], a straightforward systematic method for symbolic derivation of state space solution is presented. An attempt to exploit the method for solving large state space models unfortunately shows severe shortcomings in speed and resources usage due to high dimension matrix manipulations with its related prospects of running out of available memory [19]. Symbolic solutions of higher dimension matrices are inevitably large. For instance, computing the inverse matrix in symbolic form is a cumbersome process. 5×5 inverse matrix exceeds 200000 characters long, 6 x 6 inverse matrix exceeds 2000000, the inverse becomes 10 times larger in length for each additional increase of matrix dimension. In general, for a symbolic inverse of n×n matrix, the output size is about $2.2x10^N$. So, multiple gigabytes are needed to calculate the inverse other than the other processes. Also, the current software often generates highly complex raw expressions which are too difficult to be directly used [20].

In response to these problems, a fully symbolic based technique for solving complex state space models is proposed. The proposed technique tries to overcome the barriers of high resources, time consumption and computational complexity using the partitioned matrix theory. All the state, input and output matrices of the original system are broken up into blocks and all the subsequent processing is made on these blocks, then to be assembled at the end. Although the state space model may be of any dimension either odd or even, the partition process may lead to rectangular blocks which will be considered a problem as the processing includes inverse matrix operations. This barrier is overcome by blockwise inversion formula which insures that only either the top-left or bottom-right matrices should only be square blocks.

3. THE PROPOSED TECHNIQUE

In order to overcome the aforementioned drawbacks, a procedure that is based on partition matrix theory along with blockwise inverses is proposed. Partition matrix theory is considered to be the best method to deal with high dimension matrices. It relies on partitioning a large matrix into compatible smaller sections referred to as sub-matrices or blocks. Hence, manipulations can be performed on the smaller blocks which in turn leads to significant computational benefits [21]. The complete flow chart of the proposed symbolic-based technique for solving complex state space models is depicted in Figure 1. For low dimension systems, it is convenient to use the straightforward systematic procedure for solving state space model directly on the original matrices without any additional processing. Meanwhile for high dimension systems, it is desirable to partition the original system into lower dimensional blocks. So that any further processing is made on these blocks instead, which makes the processing much powerful and easier to be applied.

The proposed technique begins with deciding whether technique to be used according to the original model dimension. If the matrix dimensions exceed a predefined limit, all subsequent operations will be performed on sub-matrix level, otherwise on full matrix level. The value of the dimension limit is set to 7 as it was found empirically by our experiments. As the dimensions get larger, the difference between the two methods gets further entirely in favours of the partitioned matrix technique.

Partitioning occurs according to a certain criterion as shown in Figure 2. If *n* (the state matrix dimension) is an even number, the state matrix is divided into four equal quarters of size *n*/2. In case that n is an odd number, the partition process ensures that only A_{11} and A_{22} are square blocks so that they can be inverted blockwise while A_{12} and A_{21} are maintained as rectangular block matrices. In case of very high dimension matrix, the so-called nested partitions can be exploited if needed.

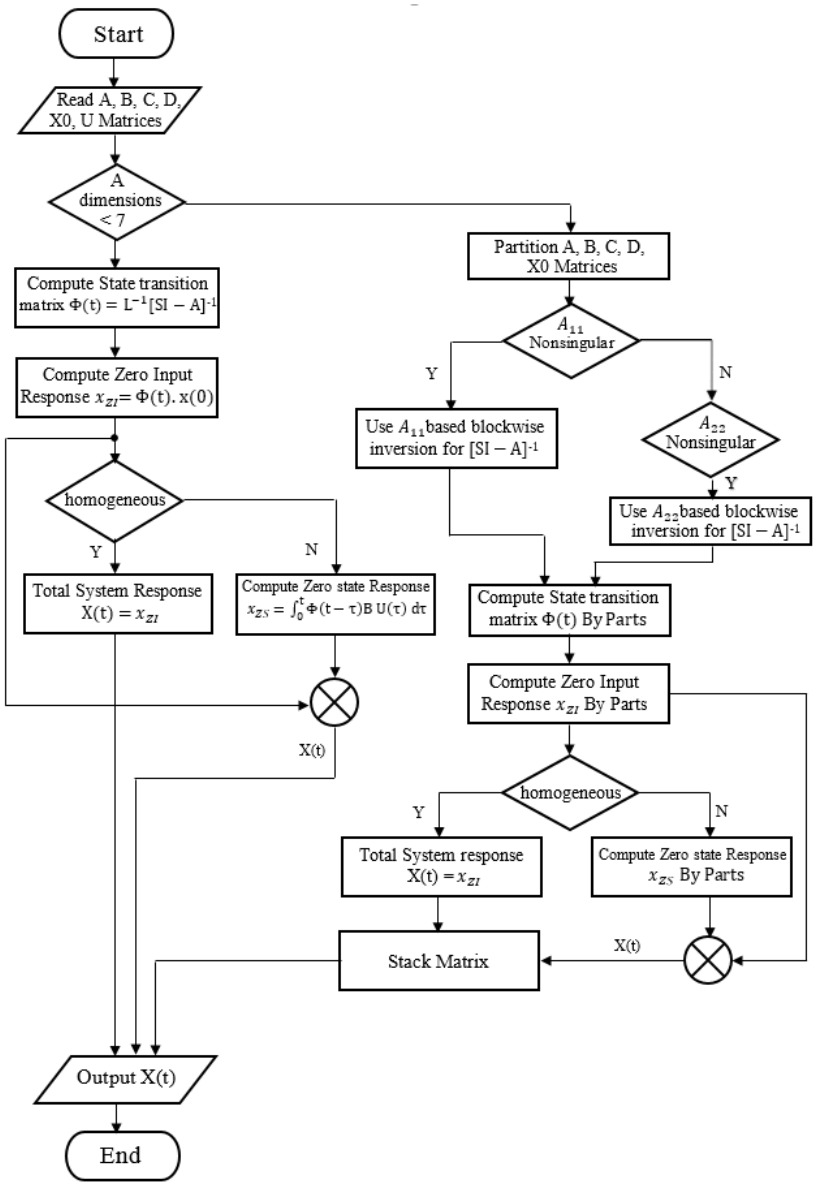


Figure 1. The proposed symbolic based technique flow chart

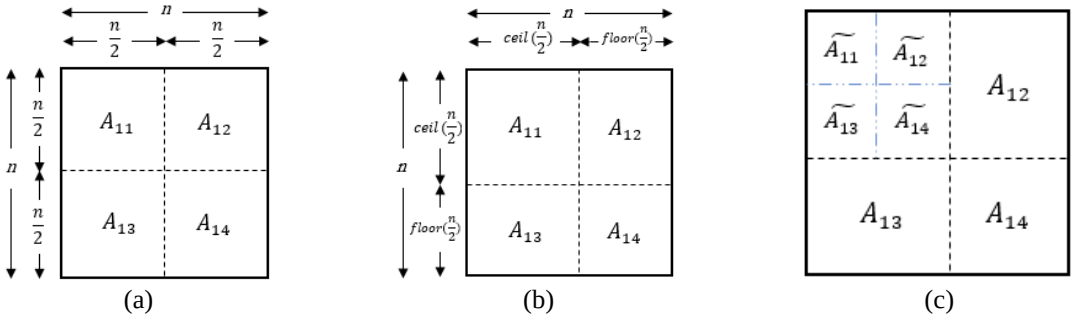


Figure 2. Partitioned matrix process in case of (a) *n* is even, (b) *n* is odd, and (c) nested partitions

The main part of the algorithm is responsible for the actual processes that form the zero state and zero input responses of the complete final solution. If the matrix was detected to be a low dimension model, (3) through (6) are employed directly on the full matrix of the original system model. However, if a high

dimension model is detected and partition was made to all model matrices, all processes should be applied on a sub-matrix basis. Beginning from the state transition matrix calculation according to (6), block matrix inversion should be carried out. The blockwise inversion formula for calculating the nonsingular partitioned matrix inverse states that: for any given matrix A that is partitioned into 4 blocks as follows,

$$A = \left[\begin{array}{ccc:cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ \hdashline a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (7)$$

The inverse matrix A^{-1} can be expressed as follows,

$$A^{-1} = \begin{bmatrix} \widetilde{A_{11}} & \widetilde{A_{12}} \\ \widetilde{A_{21}} & \widetilde{A_{22}} \end{bmatrix} \quad (8)$$

Now there are two sets of formulas to calculate $\widetilde{A_{11}}$, $\widetilde{A_{12}}$, $\widetilde{A_{21}}$ and $\widetilde{A_{22}}$ [22]:

– If both A_{11} and its schur complement (A_{22} - $A_{21}{A_{11}}^{-1}A_{12}$) are nonsingular; the matrix is invertible using,

$$\widetilde{A_{11}} = {A_{11}}^{-1} + {A_{11}}^{-1}A_{12}(A_{22} - A_{21}{A_{11}}^{-1}A_{12})^{-1}A_{21}{A_{11}}^{-1} \quad (9)$$

$$\widetilde{A_{12}} = -{A_{11}}^{-1}A_{12}(A_{22} - A_{21}{A_{11}}^{-1}A_{12})^{-1} \quad (10)$$

$$\widetilde{A_{21}} = -(A_{22} - A_{21}{A_{11}}^{-1}A_{12})^{-1}A_{21}{A_{11}}^{-1} \quad (11)$$

$$\widetilde{A_{22}} = (A_{22} - A_{21}{A_{11}}^{-1}A_{12})^{-1} \quad (12)$$

– If both A_{22} and its schur complement (A_{11} - $A_{12}{A_{22}}^{-1}A_{21}$) are nonsingular; the matrix is invertible using,

$$\widetilde{A_{11}} = (A_{11} - A_{12}{A_{22}}^{-1}A_{21})^{-1} \quad (13)$$

$$\widetilde{A_{12}} = -(A_{11} - A_{12}{A_{22}}^{-1}A_{21})^{-1}A_{12}{A_{22}}^{-1} \quad (14)$$

$$\widetilde{A_{21}} = -{A_{22}}^{-1}(A_{11} - A_{12}{A_{22}}^{-1}A_{21})^{-1} \quad (15)$$

$$\widetilde{A_{22}} = {A_{22}}^{-1} + {A_{22}}^{-1}A_{21}(A_{11} - A_{12}{A_{22}}^{-1}A_{21})^{-1}A_{12}{A_{22}}^{-1} \quad (16)$$

Also the pseudo inverse [23] was tested in the state transition matrix calculation, but it yields poor results in terms of time and memory compared to the blockwise inversion formula. Then, all the subsequent operations can be easily employed using the corresponding blocks as equations (3) to (6) states. As the inversion of high order matrices has always been a challenge due to the limited processing and memory capacity of traditional computers [24]. Therefore, combining the partition-matrix theory and blockwise inverse yields an interesting methodology with attractive solution of no matrix operations overhead for the problem of high dimensionality. Finally, the corresponding sub-matrices of zero input and zero state responses are added, then, assemble the blocks again altogether to form the complete state pace solution.

4. IMPLEMENTATION AND CASE STUDIES

Several experiments have been performed to prove the applicability of the proposed technique using MATLAB MuPAD Symbolic Math Toolbox [25] with different state space system model of various dimensions. In the next subsections, each case study is briefly described, symbolically solved, then the parameter varying behavior is analyzed and finally the results are compared with numerical solutions.

4.1. Case study #1 (ship motion model)

To fully represent the motion of a rigid body in space, six degrees of freedom (DOF) are required. The Standard Notation for ship motion is depicted in Figure 3 [26] and its parameters are given in Table 1 [26].

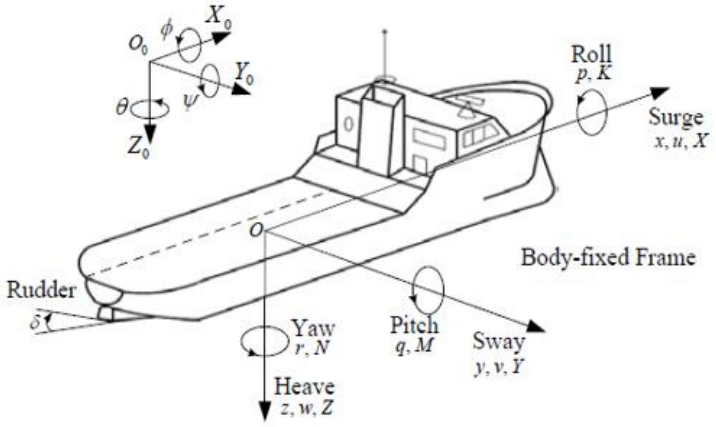


Figure 3. Standard notation for ship motion description

Table 1. Ship Motion simulation parameters definitions

Parameters	Definition	Parameters	Definition
x, y, z	Distance along body-fixed system axes [m]	Ψ	Yaw angle [rad]
u_0, *v, w*	Translational velocity in body-fixed frame [m/s]	θ	Pitch angle [rad]
Xf, Yf, Zf	Force components along body axes [kg]	ϕ	Roll angle [rad]
p, q, r	Rotational velocity in body-fixed frame [rad/s]	δ	Rudder Angle [rad]
K, M, N	Moment components along body axes [kg.m]		

However, for ship manoeuvring, a 4 DOF description that involves surge, sway, yaw and roll modes is typically regarded adequate [27]. Consequently, a fourth order state space model for the ship motion under consideration based on the Nomoto model is given in a symbolic form as follows [26],

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\Psi} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \Psi \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ b_4 \\ 0 \end{bmatrix} \delta \tag{17}$$

Applying the proposed symbolic based technique on the nonhomogeneous ship motion model, a low dimension system is indicated. Thus, the straightforward symbolic procedure that follows (3) to (6) is exploited on the original full matrices. Computing the state transition matrix, the zero input, zero state responses and finally the complete symbolic solution of the ship motion state space model is obtained. Simulation results using real ship model parameters have been undertaken for testing [26]. Numerical values for the linearized ship motion model parameters are summarized in Table 2. For illustration, Figure 4 shows the impact of varying the value of parameter a_{12} only with a step of 0.2 over the range from -2.3 to -1.5. The parameter change is chosen to be nearly within $\pm$10%.

Table 2. Numerical values for ship motion parameters

Parameter	Value	Parameter	Value
a_{11}	-0.04	a_{45}	-0.059
a_{12}	-1.933	b_1	0.1559
a_{21}	0.00011	b_3	-0.0033
a_{22}	-0.0813	b_4	0.00821
a_{44}	-0.07		

It is noted that changing a_{12} has an apparent effect on the translational velocity v, a slight effect on r and Ψ and almost no effect on both p and Φ. Also it is noted from the plot that the yaw angle response is unstable. The open-loop response does not satisfy the design criteria at all. Hence, these equations can be used to analyse the effect of any of the internal parameter changes within any range of variation.

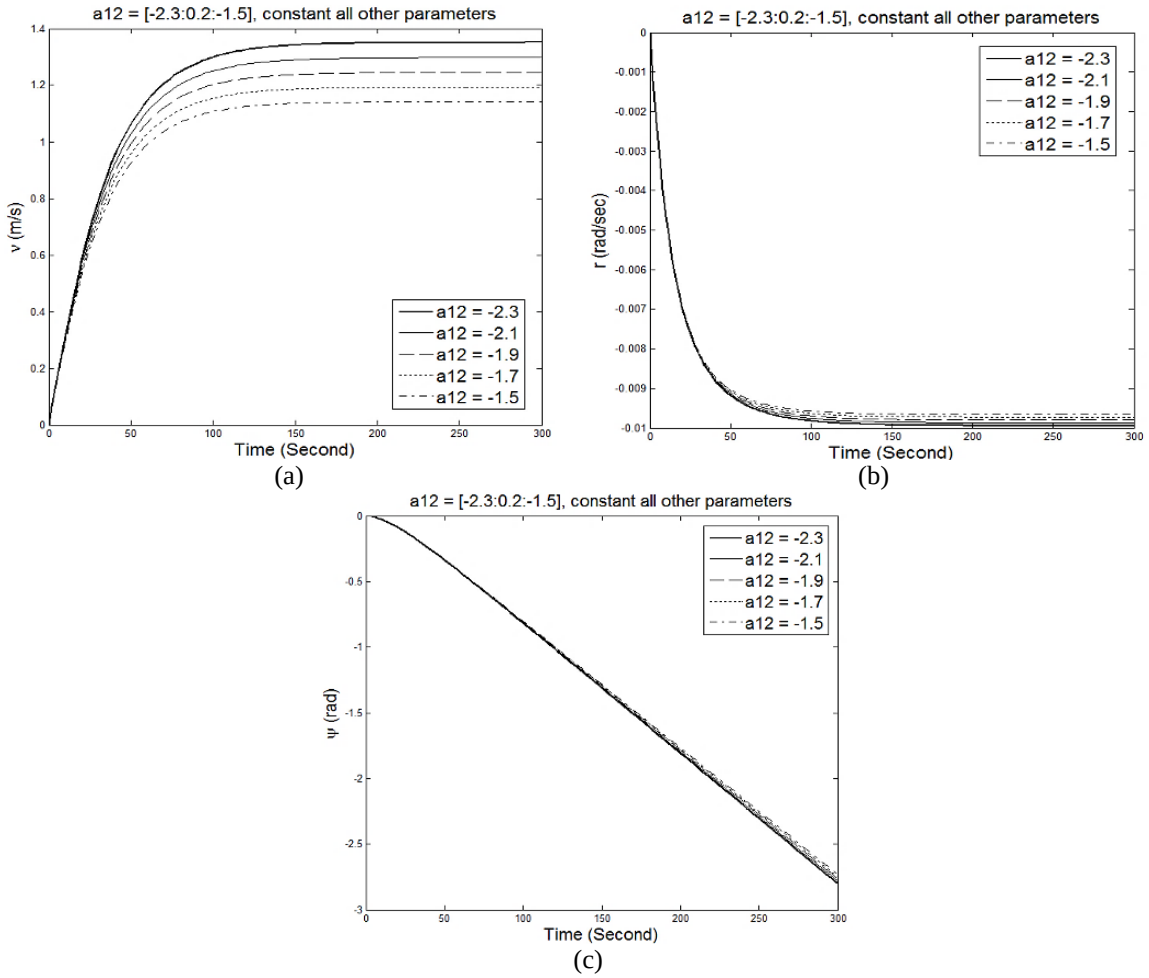


Figure 4. The effect of changing a_{12} on state variables (a) v versus time, (b) r versus time, (c) Ψ versus time

4.2. Case study #2 (liquid container motion)

This case study is concerned with a liquid container transfer system along slanted paths that reduce sloshing, avoid overflow and facilitate high-speed transmission in the steel industries [28]. The melted metal is moved to casting area from furnace after it has been poured to a ladle. A schematic diagram of a 2 DOF transferring machine on a slanted transfer path is shown in Figure 5 [29].

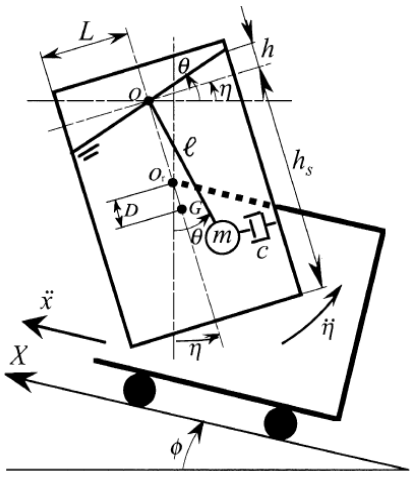


Figure 5. Liquid container transfer system model

The state space model representing the system in the form of (1, 2) is given by [29],

$$x = [\theta \quad \dot{\theta} \quad \eta \quad \dot{\eta} \quad \ddot{\eta} \quad x \quad \dot{x}]^T \tag{18a}$$

$$y = [\, h \quad x]^T \;,\; u = [\, u_1 \quad u_2]^T \tag{18b}$$

$$B = \begin{bmatrix} 0 & \frac{K_m}{l\,T_m}\cos\Phi & 0 & 0 & 0 & 0 & \frac{K_m}{T_m} \\ 0 & 0 & 0 & 0 & K_r{\omega_n}^2 & 0 & 0 \end{bmatrix}^T \tag{18c}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{-g}{l} & \frac{-c}{m} & 0 & \frac{c}{m} & 0 & 0 & \frac{-1}{l\,T_m}\cos\Phi \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -{\omega_n}^2 & -2\xi\omega_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_m} \end{bmatrix} \tag{18d}$$

where u_1 and u_2 are the linear transfer control input and rotational motion input respectively. θ is the pendulum angle from horizontal, η is the container angle, $\ddot{\eta}$ is the container rotation angular acceleration, and $\ddot{x}$ is the acceleration applied to the container. The other model parameters are summarized in Table 3.

Table 3. Liquid container transfer system model parameters

Parameter	Definition	Value	Parameter	Definition	Value
K_m	Motor gain	0.0912	Φ	Angle of slanted transfer path	5.0
T_m	Time constant	0.0227	l	Length of pendulum	0.0442
K_r	Motor gain	-0.5807	c	Coefficient of viscosity	1.88
ξ	Damping factor	0.3778	m	Mass of liquid	2.744
ω_n	Natural angular frequency	41.446	g	Gravitational acceleration	9.8
h	Liquid level	0.140			

As the model dimensions gets larger, a common criterion to facilitate the processing is to put the complex matrices in (18) in the parametric compact form as follow,

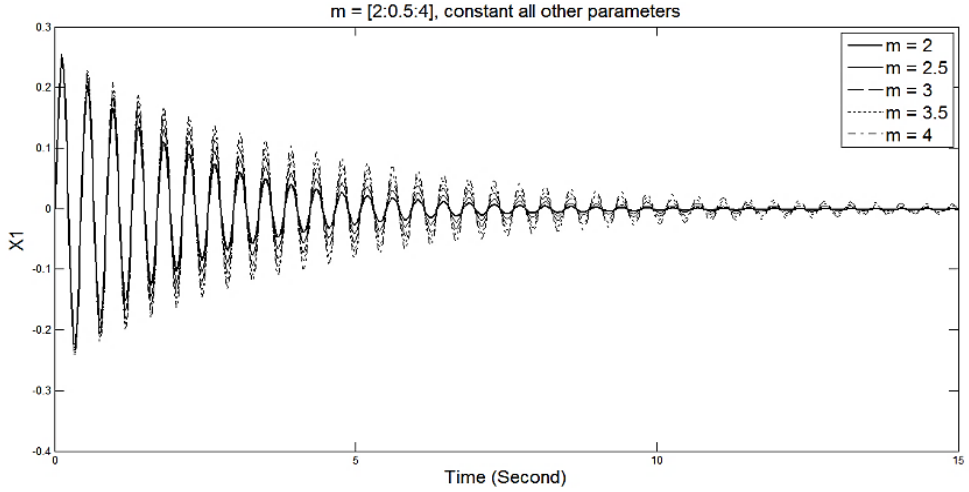
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & a_{24} & 0 & 0 & a_{27} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{77} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ b_{21} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_{52} \\ 0 & 0 \\ b_{71} & 0 \end{bmatrix} \tag{19}$$

Applying the proposed technique on the nonhomogeneous liquid container transfer state model, the matrix level decision indicates a high dimension system n=7 i.e. odd number. Thus, the system model is partitioned into lower dimensional subsystems. Then all the succeeding processes in turn are made on the submatrix level. First computing the term$[\mathrm{SI} - \mathrm{A}]^{-1}$, here both A_{11} and its schur complement are nonsingular; so, it is invertible using equations (9) to (12). Symbolic calculations are proceeded consistently, the state transition matrix is calculated as four separate blocks denoted ϕ_{ij}. Then the zero-input response is generated in two separate partitions of size (4×1) and (3×1) respectively. The zero-state response is also computed on a block-basis according to (5) that yields two separate blocks of size (4×1) and (3×1). Finally, the zero state and zero input response blocks are added and simplified to form the complete symbolic solution.

Simulation results using real model parameters have been undertaken for testing [29]. Numerical values for the liquid container transfer model parameters can be found in Table 3. Figure 6 shows the impact of varying the value of the liquid mass m over the range from 2 to 4 kilograms which represents an approximate change around $\pm$20%, whereas all other parameters values are kept unvarying. The change will affect the value of both parameters a_{22} and a_{24}.

It is noted that increasing liquid mass *m* increase the oscillations in the pendulum angle and velocity, while in the meantime has no effect on the other state parameters at all. Also, it is noted from the plot that some responses are unstable and in need for controlling criteria. Hence, these equations can be used to analyze the effect of any of the internal parameter changes within any range of variation.

(a)



(b)

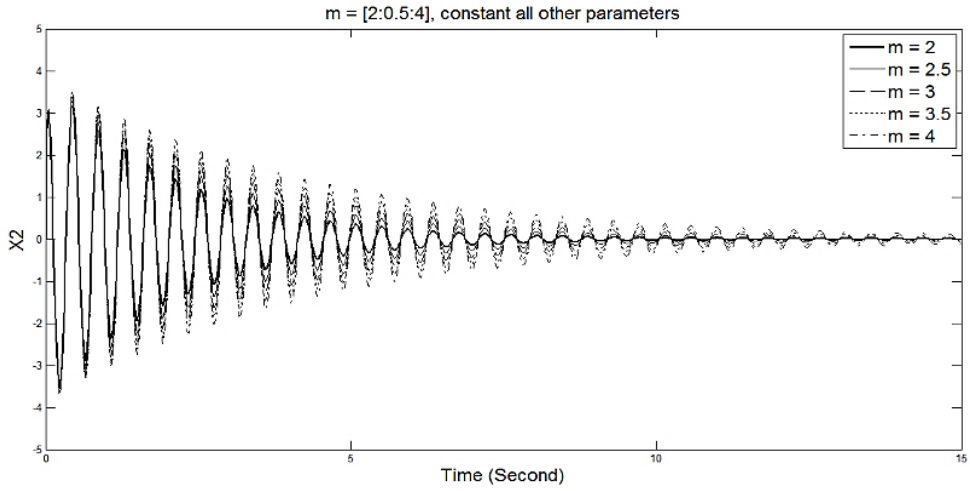


Figure 6. The effect of changing m on state variables (a) θ versus time, (b) $\dot{\theta}$ versus time

4.3. Case study #3 (Wind turbine model)

In this case study, the wind turbine state space model to be investigated is a three-blade variable speed horizontal axis wind turbine [30]. Overall state space representation in the form of (1, 2) is given by,

$$A = \begin{bmatrix} -\frac{B_{dt}+B_r}{J_r} & \frac{B_{dt}}{N_g J_r} & -\frac{K_{dt}}{J_r} & 0 & z_1 & 0 & z_2 & 0 & z_3 & 0 \\ \frac{\eta_{dt} B_{dt}}{N_g J_g} & -\frac{\eta_{dt} B_{dt}}{{N_g}^2 J_g} - \frac{B_g}{J_g} & \frac{\eta_{dt} K_{dt}}{N_g J_g} & \frac{-1}{J_g} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{-1}{N_g} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{t_g} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -{\omega_n}^2 & -2\xi\omega_n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -{\omega_n}^2 & -2\xi\omega_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -{\omega_n}^2 & -2\xi\omega_n \end{bmatrix} \quad (20a)$$

$$B=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\\frac{1}{t_g}&0&0&0\\0&0&0&0\\0&{\omega_n}^2&0&0\\0&0&0&0\\0&0&{\omega_n}^2&0\\0&0&0&0\\0&0&0&{\omega_n}^2\end{bmatrix} \quad (20b)$$

where the state and input vectors are,

$$x(t) = [\omega_r \quad \omega_g \quad \theta_\Delta \quad \tau_g \quad \beta_1 \quad \dot{\beta}_1 \quad \beta_2 \quad \dot{\beta}_2 \quad \beta_3 \quad \dot{\beta}_3]^T$$
$$u(t) = [\tau_{g,ref} \quad \beta_{1,ref} \quad \beta_2, ref \quad \beta_{3,ref}]^T$$

where ω_r is rotor speed, ω_g is generator speed, β_i is pitch position for blade i, θ_Δ is the drive train torsion angle, and τ_g is the generator torque controlled by the reference $\tau_{g,ref}$. Other parameters are given in Table 4.

Table 4. Wind turbine model parameters

Parameter	Definition	Value	Parameter	Definition	Value
B_{dt}	Drive train torsion damping coefficient	775.49	N_g	The gear ratio	390
J_r	Moment of inertia for low-speed shaft	55×10^6	t_g	Time constant	0.02
J_g	Moment of inertia for high-speed shaft	95	ξ	The damping factor	0.6
K_{dt}	The torsion stiffness of the drive train	27×10^8	ω_n	The natural frequency	11.11
η_{dt}	The efficiency of the drive train	0.97	g	Gravitational acceleration	9.8
B_g	Viscous friction of high-speed shaft	45.6			

To facilitate the processing, the complex matrices (20) are represented in the parametric compact. By applying the proposed symbolic based technique on the nonhomogeneous wind turbine state space model, the matrix level decision indicates a high dimension system n=10 i.e. even number. Thus, the system model is partitioned into lower dimensional subsystems. Then all the succeeding processes in turn are made on the submatrix basis. First computing the term $[\mathrm{SI}-\mathrm{A}]^{-1}$, where both A_{11} and its schur complement are nonsingular; so it is invertible using equations from (9) to (12). Symbolic calculations are proceeded consistently, the state transition matrix is calculated as four separate blocks denoted ϕ_{ij} of size (5×5) for each block. Then the zero-input response is generated in two separate partitions of size (5×1). The zero-state response is also computed on a block-basis according to (4) that yields two separate zero state blocks of size (5×1). Finally, the zero state and zero input response blocks are added and simplified to form the complete symbolic solution of the wind turbine state space model.

Simulation results using real model parameters have been undertaken for testing [30]. Numerical values for wind turbine model parameters are summarized in Table 4. For illustration, Figure 7 shows the impact of varying the value of parameter z_1 only over the range from 0.001 to 0.02, whereas all other parameters values are kept unvarying. It is noted that changing z_1 has a noticeable impact on rotor and generator speed and torsion angle of drive train ($\omega_r, \omega_g, \theta_\Delta$), while in the meantime has no effect on the other state parameters at all. Hence, these equations can be used to analyze the effect of any of the internal parameter changes within any range of variation. Moreover, it is noted from the plot that some responses are unstable and in need for a controlling criterion.

For all the previous case studies, the performance of the proposed symbolic-based technique has been compared to the corresponding Simulink model. Up to now, Simulink is totally numerical. To verify the correctness of the resulted symbolic solution, a Simulink model for each case study has been developed and the numerical values for the parameters have been fed into the state space block. The resulted outputs from the scope blocks show the same behaviour identically to that generated from the symbolic equations in all case studies. At last it can be concluded from all case studies implementations that the symbolic solution gives identical results as the numerical corresponding solution. So, the symbolic based model can be a better alternative to the numerical methods which has the advantage of generality and flexibility. Moreover, the partitioned matrix technique improves the memory use and time consumption than the direct method where the time is decreased by 20% and memory use by about 5% as calculated from the previous case studies experimentations.

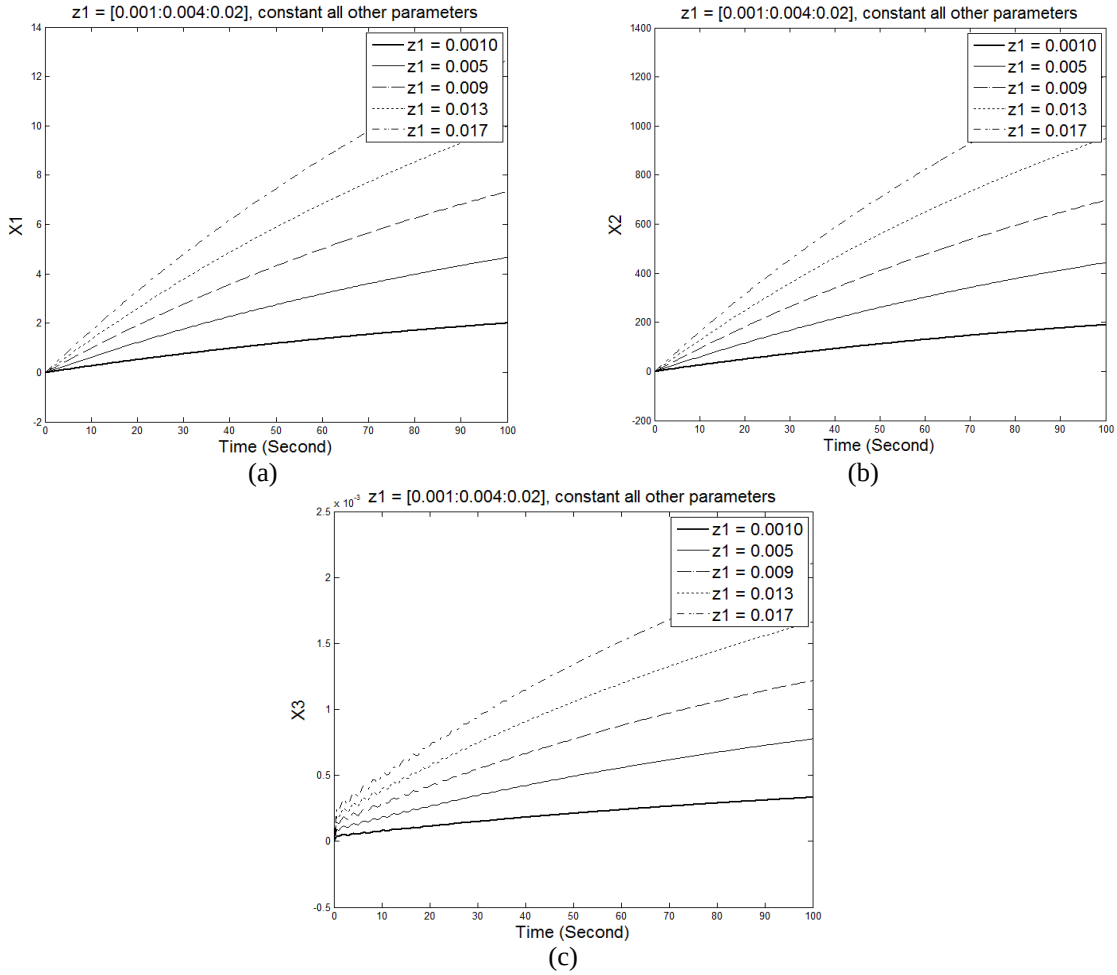


Figure 7. The effect of changing z_1 on state variables (a) x_1 vs. time, (b) x_2 vs. time, and (c) x_3 vs. time

5. CONCLUSION

In this paper, a fully generic symbolic-based technique is proposed for solving complex state space models. The proposed technique presents a methodology that facilitates the applicability of fully symbolic processes for large-scale systems using the partitioned matrices theory and blockwise inversion method. The aim is to attain final neat equations that are easy to be implemented and, in the meantime, not computationally intensive in terms of time and memory requirements. Simulation results reveal a no-error prone methodology that achieves identical results when compared to the corresponding numerical Simulink model. So, the symbolic based model can be a good alternative to the numerical methods with the advantage of flexibility and generality that leads to better understanding of the system being investigated. In the future work, the fully symbolic method will cover different types of controllers, multi-input multi-output (MIMO) systems manipulation and parameter identification.

REFERENCES

[1] S. Thota and S. D. Kumar, "Symbolic algorithm for a system of differential-algebraic equations," *Kyungpook Math Journal*, vol. 56, no. 4, pp. 1141-1160, 2016.
[2] A. M. Abd-Alrahem, et al., "Toward Symbolic Representation and Analysis of Parameter Varying Control Systems," *Proc. 20th International Middle East Power Systems Conference MEPCON 2018*, pp. 894-899, 2018.
[3] J. R. Leigh, "Control theory : A guided tour. Institution of Engineering and Technology," *3rd ed.*, 2012.
[4] N. Munro, "The Symbolic Methods in Control System Analysis and Design," *Stevenage, UK: Institution of Electrical Engineers*, 1999.
[5] B. Palancz, et al., "Product review-Control system professional suite," *IEEE Control Systems Magazine*, vol. 25, no. 2, pp. 67-75, 2005.
[6] M. T. Söylemez and İ. Üstoğlu, "Block Diagram Reduction Toolbox," 2006.

[7] Y. Zeng et al., "Modeling electromechanical aspects of cyber-physical systems," *Journal of Software Engineering for Robotics*, vol. 7, no. 1, pp. 100-119, 2016.
[8] G. Pola, et al., "Symbolic control design of nonlinear systems with outputs," *Automatica*, vol. 109, 2019.
[9] M. Fakhroleslam, et al., "Time-optimal symbolic control of a changeover process based on an approximately bisimilar symbolic model," *Journal of Process Control*, vol. 81, pp. 126-135, 2019.
[10] A. Girard, et al., "Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1537-1549, 2016.
[11] N. M. Radaydeh and M. R. D. Al-Mothafar, "Small-signal modeling of current-mode controlled modular DC-DC converters using the state-space algebraic approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 139-150, 2020.
[12] M. Mizoguchi and T. Ushio, "Deadlock-free output feedback controller design based on approximately abstracted observers," *Nonlinear Analysis: Hybrid Systems*, vol. 30, pp. 58-71, 2018.
[13] A. Borri, et al., "Design of symbolic controllers for networked control systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1034-1046, 2019.
[14] L. Shamgah, et al., "A Symbolic approach for multi-target dynamic reach-avoid problem a symbolic approach for multi-target dynamic reach-avoid problem," *IEEE 14th Int. Conf. on Cont. and Automation*, pp. 1022-1027, 2018.
[15] H. T. Dorrah, et al., "Derivation of symbolic-based embedded feedback control stabilization expressions with experimentation," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 427-441, 2018.
[16] V. Mladenović, et al., "Symbolic analysis as universal tool for deriving properties of symbolic analysis as universal tool for deriving properties of non-linear algorithms-case study of EM algorithm," *Acta Polytechnica Hungarica*, vol. 11, no. 2, pp. 117-136, 2014.
[17] E. Setiawan, et al., "Accurate symbolic steady state modeling of buck converter," *International Journal of Electrical & Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2374-2381, 2017.
[18] K. Ogata, "Modern Control Engineering," *5th ed., Prentice Hall*, 2010.
[19] R. Fateman, "Manipulation of matrices symbolically," University of California, Berkeley, 2003, [Online]. Available: http://www.cs.berkeley.edu/~fateman/papers/symmat2.pdf.
[20] M. Shokouhifar and A. Jalali, "Evolutionary based simplified symbolic PSRR analysis of analog integrated circuits," *Analog Integrated Circuits and Signal Processing*, vol. 86, no. 2, pp. 189-205, 2016.
[21] H. T. Dorrah, et al., "Generic symbolic parameters varying systems frameworks versus other techniques: Returning back to the roots," *Alexandria Engineering Journal (AEJ, Elsevier)*, vol. 57, no. 4, pp. 3577-3594, 2018.
[22] T. -T. Lu and S. -H. Shiou, "Inverses of 2×2 block matrices," *Computers & Mathematics with Applications*, vol. 43, no. 1, pp. 119-129, 2002.
[23] M. K. Viswanath and M. Ranjith Kumar, "Design and implementation of a secure communication protocol," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1814-1821, 2018.
[24] I. C. S. Cosme, et al., "Memory-usage advantageous block recursive matrix inverse," *Applied Mathematics and Computation*, vol. 328, pp. 125-136, 2018.
[25] Matlab, "Symbolic Math Toolbox, User's Guide, Ver. 2016b," *The MathWorks Inc., Natick, MA, USA*, 2016.
[26] R. Fraga and L. Sheng, "An effective state-space feedback autopilot for ship motion control," *Journal of Control Engineering and Technology*, vol. 2, no. 2, pp. 62-69, 2012.
[27] E. Uyar, et al., "Dynamic modelling, investigation of manoeuvring capability and navigation control of a cargo ship by using Matlab simulation," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 104-110, 2016.
[28] H. Sayyaadi and A. Soltani, "Modeling and control for cooperative transport of a slung fluid container using quadrotors," *Chinese Journal of Aeronautics*, vol. 31, no. 2, pp. 262-272, 2018.
[29] K. Yano, et al., "Motion control of liquid container considering an inclined transfer path," *Control Engineering Practice*, vol. 10, no. 4, pp. 465-472, 2002.
[30] R. Yang, "Control of wind turbines using takagi-sugeno approach," *M.S. thesis, Politècnica de Catalunya University, Barcelona, Spain*, 2017.