

Dynamic simulation of the CAD model in SimMechanics with multiple uses

Jiří ZÁTOPEK¹, Zdeněk ÚŘEDNÍČEK¹, José MACHADO^{2,*}, João SOUSA²

¹Department of Automatic and Control Engineering, Faculty of Applied Informatics,
Tomas Bata University in Zlín, Zlín, Czech Republic

²Department of Mechanical Engineering, School of Engineering, University of Minho, Azurém Campus,
Guimarães, Portugal

Received: 16.12.2017

Accepted/Published Online: 03.04.2018

Final Version: 30.05.2018

Abstract: When designing a mechatronic system, several steps are taken into account. One of the main steps is the design of a CAD model representing the physical part of the system, and another major point is the development of the mathematical model necessary for the respective controller design. This paper combines both design steps and shows the advantages of using this approach. First, a CAD model is created considering the kinematic and dynamic behavior of the system as well as respective material properties. This CAD model is, in parallel, used for both purposes: as the main basis for developing a mathematical model that will be used for definition of control laws and appropriate system controllers, and also to generate a physical model as result of exporting to MATLAB/Simulink (Simscape/SimMechanics library) in order to simulate the system behavior. This translation does not consider only the standard CAD model export to the SimMechanics library when forces and torques between links are clearly defined, but also the correct way to add corresponding limiting forces/torques. When comparing the behavior of the physical model and the mathematical model, it is important to obtain similar results, especially when it is necessary to perform some simplifications of a mathematical model, as happens in the context of nonlinear systems control. All these issues are discussed in this paper and the obtained simulation results for both models are similar, which confirms the proposed approach.

Key words: Model-based design, dynamics, simulation, Simscape, SimMechanics, computer-aided design, motion control

1. Introduction

The information about masses and inertias of an object of interest is a basic assumption to do precise dynamic modeling [1,2]. The CAD model is very helpful for this purpose [3–5]. Only shape and material properties are necessary for a full dynamic description, and these parameters are included in the CAD model [1]. This means that CAD modeling should always be the first step in an accurate dynamic simulation of any system. It is also advantageous in mathematical model determination [4,6–8], not only for simulation purposes.

The system that is the object of our interest is a special serial kinematic system structure known as “ball and plate”. This structure is unstable and increases nonlinearities, which is problematic for its control [7,9,10].

The main task of this work is to simulate the dynamic behavior of the system as accurately as possible using the CAD model. SolidWorks is used for the 3D model design [1,4] and MATLAB/Simulink [11–13] (specifically, the Simscape/SimMechanics library) is used for the physical model, thus providing its dynamic simulation [1,14]. The final simulation can partially replace the real (physics) prototype.

*Correspondence: jmachado@dem.uminho.pt

The estimation of process model parameters, supporting the development of complex systems, can be accomplished with a model-based design by identifying a set of experimental conditions yielding the most informative process data to be used. Several authors addressed their research related to design optimization using model-based related techniques.

The Modelica language, an extensively used physical modeling language developed in an international effort [15], is determined by noncausal models using ordinary differential and algebraic equations and it uses object-oriented constructs to facilitate the reutilization of modeling data. In [16], the software tool and the modeling language requirements for mechatronic system simulations were presented, along with the simulation of a machining center using Dymola software. The model was successfully validated using collected data from the real machining center. However, a language like Modelica only allows a one-way translation of CAD to Modelica models regarding geometrical and inertial parameters. Consequently, some researchers have tried to integrate this modeling technique with CAD software [5,11,17–19]. An alternative case of model-based modeling is Simscape [1], an integrated package within MATLAB's Simulink toolbox. In [20] the accuracy of the dynamic model of a mechanism used in Simulink was improved using a CAD system that enables the study of mass properties and the dynamics of complex mechanisms. In [21] a KUKA KR5 industrial robot's dynamic simulation was performed using SimMechanics under the Simulink toolbox along with CAD modeling software Autodesk Inventor. Another example of SimMechanics usage is the teaching of parallel mechanisms and their controllers, where students need to project a parallel manipulator of their preference using MATLAB toolboxes like SimMechanics, Simulink, and/or CAD software [3]. In [22] the possibilities of controlling the movement of a 6-DOF excavator using GPS and a computer-aided design model of the terrain surface were examined using MSC Adams software and MATLAB/Simulink [22]. In [4] the development of the CAD model of a SCARA (selective compliance articulated robot arm) manipulator using SolidWorks software was presented, along with a controller based on the proportional-integral-derivative (PID), designed in a simulation environment using the MATLAB/Simulink platform. This paper shows the advantages of the combination of MATLAB and SolidWorks, being the latest able to ease the modeling process [4]. In [23] SolidWorks and MATLAB/Simulink were used to confirm the theory and motion simulation regarding the comparison of robot positions using equal trajectory and time frame along with the establishment of a computer code for finding both kinematic and dynamic parameters.

The presented work's approach for SimMechanics model compilation [1,4,20] cannot be used, because the appropriate joints in SimMechanics do not exist for the discussed model, "ball and plate". The CAD models' mates in related works correspond with SimMechanics joints; they may not deal with this paper's problem. With a general 6-DOF joint used for the ball, corresponding reaction forces and torques have to be necessarily manually added. The forces and torques caused by friction, spring stiffness, damping coefficient, etc. are described and added to the SimMechanics model, which requires a different approach to build the physical model. The value of this paper is also in the application of the physical model, not only for simulation purposes, but also for controller adjusting, for discovering the required forces and torques for the actuators, the determination of the forces acting on joints and individual parts, or for the decision about the correctness of the mathematical model.

2. CAD model - SolidWorks

The SolidWorks model consists of 11 parts, with only 8 of them moving, as shown in Figure 1. Each part has an associated local coordinate system and designed material and they are joined to each other by structural bonds for a final fully bonded assembly.

The system has 8 DOF generally: 2 rotational DOF provided by the actuators (movement of the inclined plate) and 6 DOF for the ball. For the simulation, all 8 DOF are considered, but for the mathematical description, only 4 DOF are considered, which is sufficient for design control law purpose.

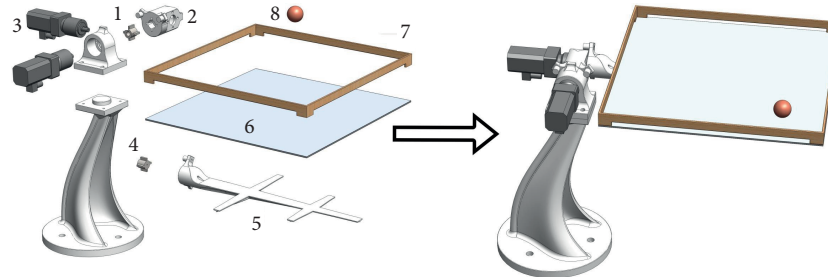


Figure 1. The CAD model of the analyzed system.

3. Physical model - SimMechanics

The SolidWorks model is exported directly into the SimMechanics Second Generation diagram along with all linkages, dimensions, and material properties. This export possibility is accessible via *Tools > SimMechanics Link > Export > SimMechanics Second Generation*. Secondly, the assembly is exported to XML format and each part to STL format. After the exporting procedure, the translated model is no longer associated with the SolidWorks CAD model. That means that if any changes in dimensions, materials, or mate associations are required, the whole model must be reexported. This process is unidirectional.

SimMechanics (in MATLAB 2014b version and lower) supports only the standard SolidWorks mates (except for tangent and lock mates); more advanced mates, required for a correct SolidWorks Motion analysis, are not supported by SimMechanics and a manual input of the mates is required.

The SimMechanics scheme in Figure 2 shows the basic configuration after importing the XML file to MATLAB - command *smimport('ExportedModel.xml')*. Only *6-DOF Ball Joint* is manually added because of the previously mentioned mate's incompatibility. In frame number 1 in Figure 2, the objects firmly connected to the ground are presented; the blocks in frame number 2 (parts 1, 2, and 3 in Figure 1) are connected to the first motor gearbox output. The blocks in frame number 3 (parts 4, 5, 6, and 7 in Figure 1) are connected to the second motor gearbox output, and the *Ball* block (part 8 in Figure 1) displays the ball, which is unconstrained. Only the position of its local coordinate system is known and can be managed.

The simulation works correctly, except for the ball, whose forces and torques must be defined.

3.1. The ball forces/torques

Different forces and torques act on the ball during the simulation. In Figure 3 possible scenarios with appropriate forces/torques are shown. In the case of the example in Figure 3a, the plate is horizontally oriented and the gravitational force F_G acts perpendicular to the plate. When the ball does not touch the plate, only gravitational force is available. After the ball touches the plate, a plate force F_P starts to act against the gravitational force. The plate force is based on the plate material properties, especially on spring stiffness and damping coefficient.

When the plate changes its orientation to that shown in Figure 3b, the ball starts rolling due to the distribution of the gravitational force (according to the tilt angles) to the plane of the plate. Rolling (more precisely, the torques T_{Ffp1} and T_{Ffp2} actuate the rolling) is based on the friction forces F_{fp1} and F_{fp2}

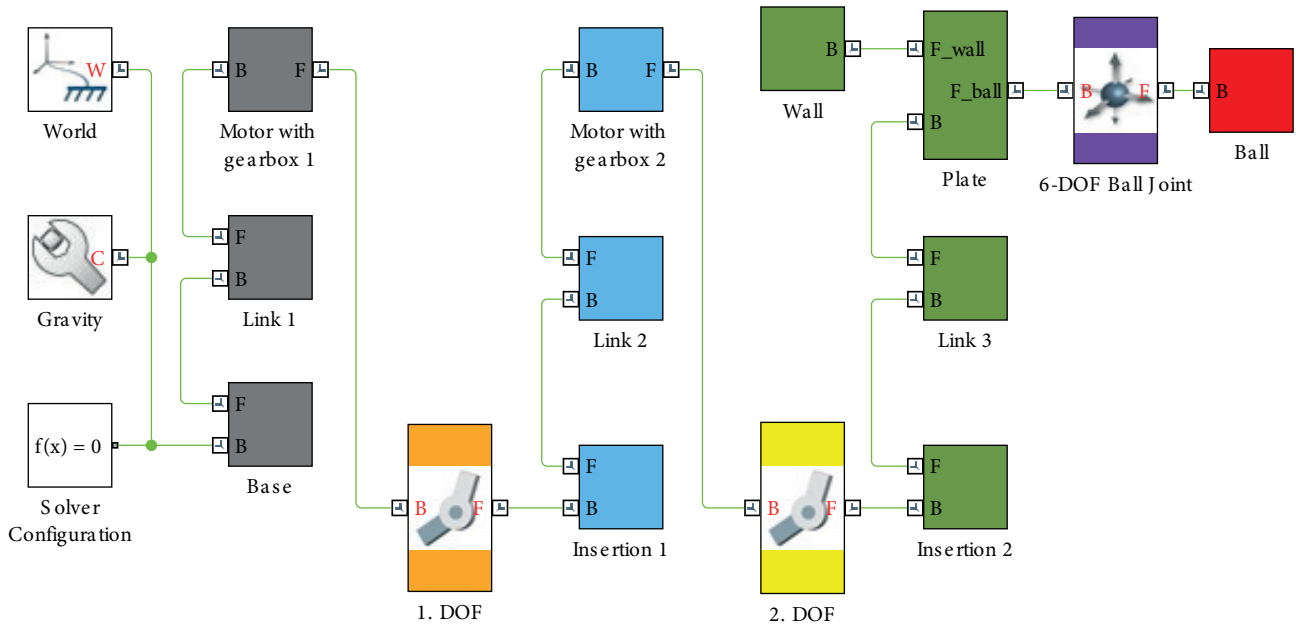


Figure 2. Native SimMechanics Second Generation scheme.

between the ball and the plate. In this dynamic simulation an ideal rolling is not considered, but sliding is taken into account.

After the ball touches a wall as in Figure 3c, the wall's force F_W starts to act against the torque/force from the rolling/sliding ball (the same principle as the plate force). All the other forces/torques are transferred from the previous example. Between the ball and the wall there is also a friction force F_{fw} causing torque T_{Ffw} .

The last example in Figure 3d shows the ball in the corner of the plate. The ball touches two walls and also the plate. Compared to the previous example in Figure 3c there is an added force F_{W2} from the second wall. A friction force F_{fw2} and a torque T_{Ffw2} are the same type as in the previous case in Figure 3c: F_{fw} (now F_{fw1}) and T_{Ffw} (now T_{Ffw1}).

The wall's forces have defined its maximal effect height above the plate. After the ball overcomes this height, the wall's forces cease to act on the ball and this means the ball can overlap them. The ball can also penetrate the walls if the FW force exceeds the strength of the material. The walls can be also removed. In the simulation, rolling can be restrained, which means the simulation is valid also for a cube or other objects, not only for the ball.

3.2. Friction characteristics

Static, kinematic, and linear friction is included in the physical model. The linear friction is negligible in this case, but the static and the kinematic frictions are very important, because they are related to the rolling and/or sliding movements of the ball. The friction characteristic is approximated by the combined arctangent function with the constant value shown in Figure 4.

When slip rate $v_{slip}[\omega(t), v(t)]$ is lower than the defined maximum slip velocity v_{Fr} , the friction is in its static part with the maximum value of μ_S . When the slip rate is higher than v_{Fr} , friction is reduced to the kinematic friction constant μ_K . The resulting friction force F_{frict} is based on the friction force founded in

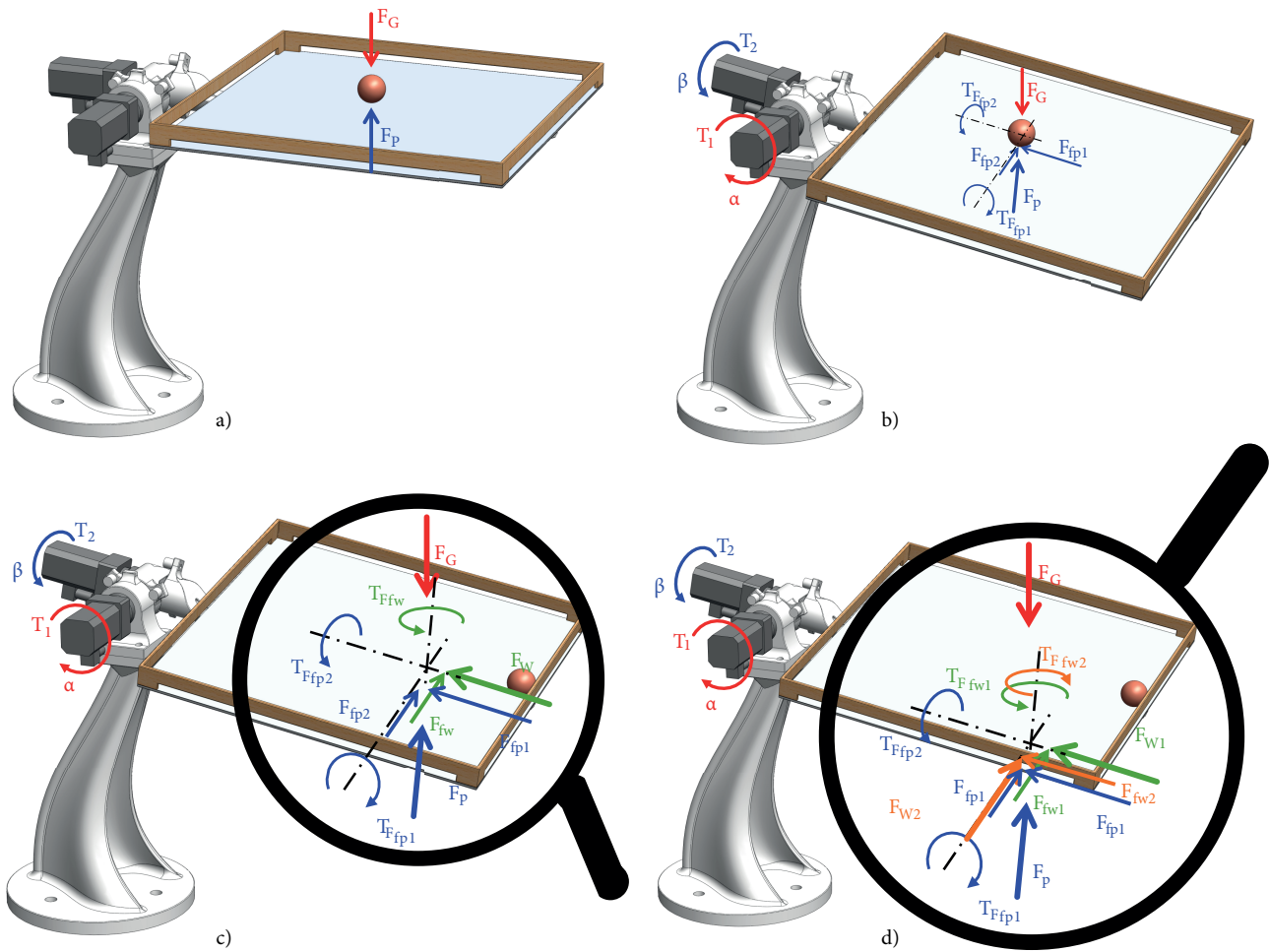


Figure 3. The ball forces/torques during a simulation.

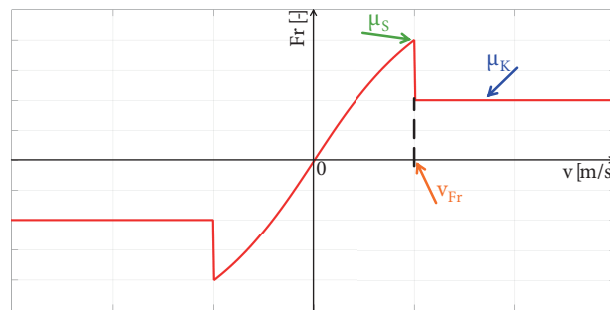


Figure 4. The static and kinematic friction included in the physical model.

the characteristic, multiplied by the normal force F_N , with which the ball acts towards the plate/walls. Also included is a linear friction μ_{Lin} , which is added to the resulting friction force and depends on the translation velocity $v(t)$ of the ball. Everything mentioned here is written in Eq. (1).

$$F_{frict} \{v_{Slip} [\omega(t), v(t)], v(t)\} = \left\{ \{ |v_{Slip} [\omega(t), v(t)]| < v_{Fr} \} \cdot \mu_S \cdot \tan^{-1} \left(\frac{v_{Slip} [\omega(t), v(t)]}{v_{Fr}} \right) \cdot \frac{4}{\pi} + \{ |v_{Slip} [\omega(t), v(t)]| \geq v_{Fr} \} \cdot \mu_K \cdot \frac{|v_{Fr}|}{v_{Fr}} - \mu_{Lin} \cdot v(t) \right\} \cdot F_N \quad (1)$$

3.3. The final physical model

Important changes in the final SimMechanics model (Figure 5) versus the native SimMechanics model (Figure 2) are in sensing the information about ball position and velocity/angular velocity and using this information for determination of the forces shown in Figure 3. The main function is in block *MATLAB Function*.

As shown in the black frame in Figure 5, all forces and torques are calculated in the function block and connected to the *6-DOF Joint* block input. The output of the *6-DOF Joint* block is firmly connected with the ball.

3.4. Physical model visualization

Each physical model contains a visualization mode, shown in Figure 6, which is enabled after the simulation has started. The visualization includes exported STL components and the graphical appearance of the whole assembly is similar to the original CAD model with only some minor changes (graphics and color quality).

The visualization can be slowed down to $1/256$ s and can also be accelerated with the same ratio. This is especially useful for rapid dynamics systems for accurate observation of fast processes. The visualization has some predefined views and can also be rotated in 3D. It is possible to display coordinate systems and centers of gravity as well and many other possibilities.

4. Mathematical model

In the case that a CAD model is available (i.e. the matrices of inertia and centers of gravity position in the local coordinate systems are known), it is advantageous to use the motion equations matrix form. Although the resulting motion equations are always more accurate, they are always more complicated, too. If the physical model is not available, it would be reasonable to make this precise mathematical model, for example, because of the controller setup.

In this case, the mathematical model is only for design control law purposes. It means that the resulting motion equations have to be as simple as possible, but at the same time, the dynamic behavior of the simplified model cannot be much different from the original system's behavior. Therefore, the system was simplified to 4 DOF (the ball rotation was neglected) and all the fixed parts were replaced by the mass point, located in their center of gravity. At the end of the paper, the results of the physical and the simplified mathematical model will be compared and it will also be discussed whether the mathematical description is sufficient for design control law purposes.

4.1. Transformation of coordinate systems

The model is separated into three parts in order to determine the homogeneous coordinates of the system. Figure 7 illustrates the placement of system coordinates.

Denavit–Hartenberg notation was used. Matrix 0T_3 in Eq. (2) is the transformation from the local coordinate system (x_3, y_3, z_3) into the global coordinate system (X_0, Y_0, Z_0) and it is the transformation for the first DOF (parts 1, 2, and 3 in Figure 1). Matrix 0T_4 in Eq. (3) is for the second DOF (parts 4, 5, 6, and

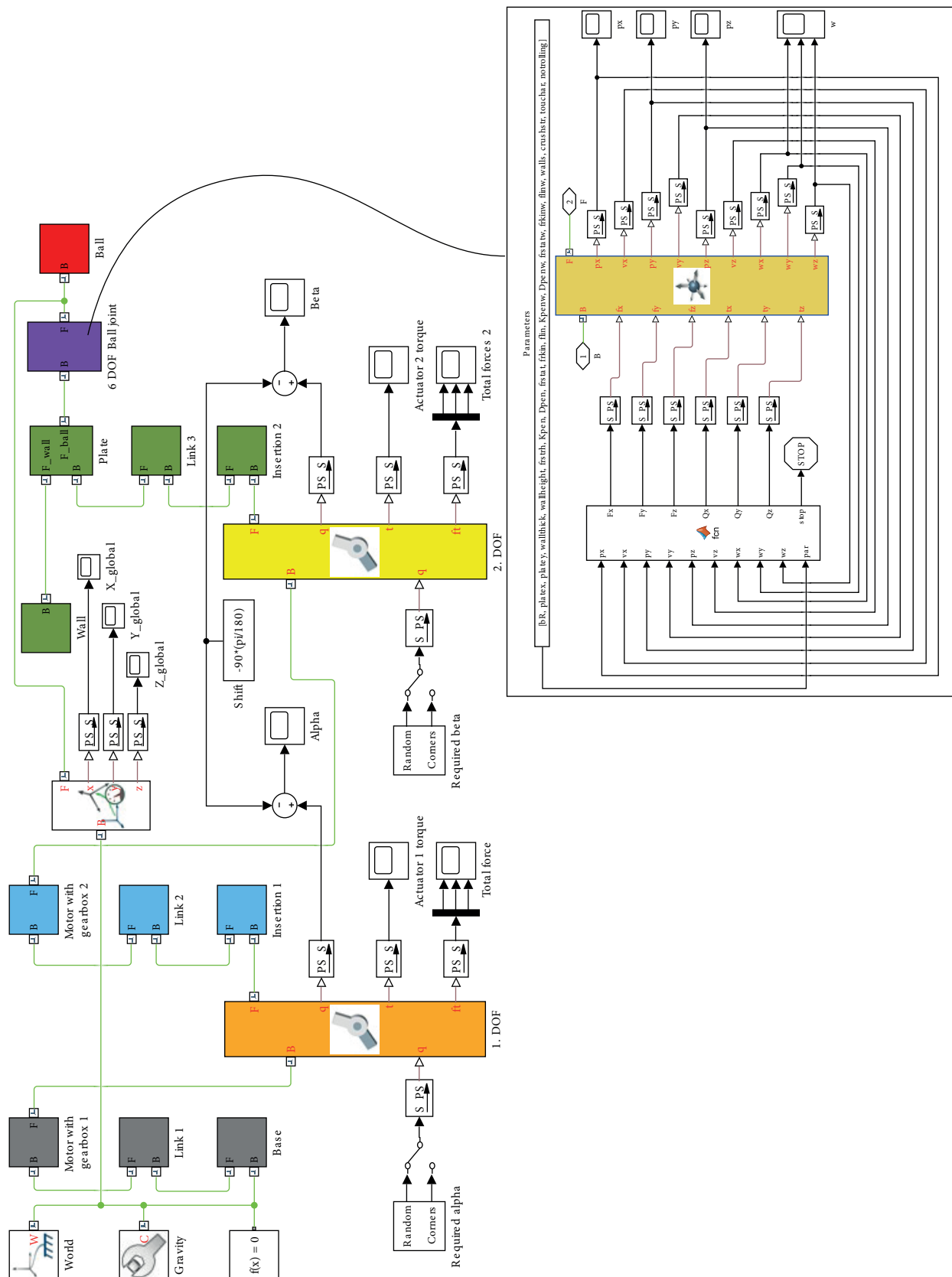


Figure 5. The final SimMechanics physical model.

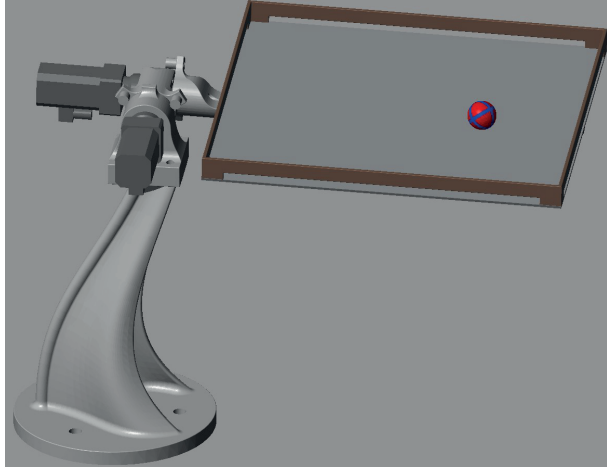


Figure 6. The SimMechanics physical model visualization.

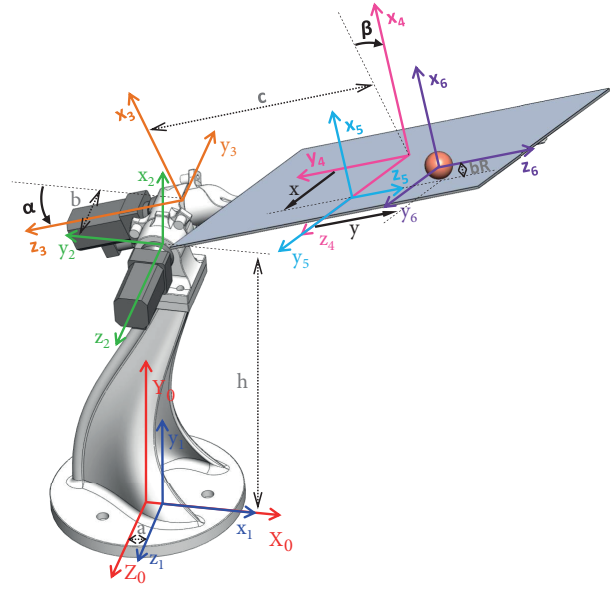


Figure 7. Coordinate system placement.

7 in Figure 1) and matrix 0T_6 in Eq. (4) is for the third and fourth DOF (part 8 in Figure 1). It follows that the state variables are α , β , x , y .

$${}^0T_3 = \begin{bmatrix} -\sin(\alpha) & 0 & -\cos(\alpha) & a \\ \cos(\alpha) & 0 & -\sin(\alpha) & h \\ 0 & -1 & 0 & -b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^0T_4 = \begin{bmatrix} -\cos(\beta) \cdot \sin(\alpha) & -\cos(\alpha) & -\sin(\alpha) \cdot \sin(\beta) & a + c \cdot \cos(\alpha) \\ \cos(\beta) \cdot \cos(\alpha) & -\sin(\alpha) & \cos(\alpha) \cdot \sin(\beta) & h + c \cdot \sin(\alpha) \\ -\sin(\beta) & 0 & \cos(\beta) & -b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^0T_6 = \begin{bmatrix} -\cos(\beta) \cdot \sin(\alpha) & -\sin(\alpha) \cdot \sin(\beta) & \cos(\alpha) & a + c \cdot \cos(\alpha) + y \cdot \cos(\alpha) - x \cdot \sin(\beta) \cdot \sin(\alpha) - bR \cdot \cos(\beta) \cdot \sin(\alpha) \\ \cos(\beta) \cdot \cos(\alpha) & \cos(\alpha) \cdot \sin(\beta) & \sin(\alpha) & h + c \cdot \sin(\alpha) + y \cdot \sin(\alpha) + x \cdot \sin(\beta) \cdot \cos(\alpha) + bR \cdot \cos(\beta) \cdot \cos(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 & -b + x \cdot \cos(\beta) - bR \cdot \sin(\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The center of mass position vectors for the above mentioned local coordinate systems are:

$${}^3r_3 = [r_{11} \ r_{12} \ r_{13} \ 1]^T; {}^4r_4 = [r_{21} \ r_{22} \ r_{23} \ 1]^T; {}^6r_6 = [r_{31} \ r_{32} \ r_{33} \ 1]^T \quad (5)$$

4.2. Motion equations

The system has 4 DOF and therefore it has 4 motion equations. The Lagrange equations of the 2nd kind were used to find the motion equations. To be able to use Lagrange equations, it is necessary to define the kinetic

and potential energy of the whole system. The transformation matrices and the center of mass position vectors are used to determine the absolute position and velocity of the mass point.

The first part's position vector in the global coordinate system:

$$p_1 = {}^0\mathbf{T}_3 \cdot {}^3r_3 = [p_{11} \ p_{12} \ p_{13} \ 1]^T \quad (6)$$

The first part's velocity vector in the global coordinate system:

$$v_1 = \frac{dp_1}{dt} = [v_{11} \ v_{12} \ v_{13} \ 1]^T \quad (7)$$

The kinetic energy of the first part:

$$E_{K1} = \frac{1}{2} \cdot m_1 \cdot (v_{11}^2 + v_{12}^2 + v_{13}^2) \quad (8)$$

The potential energy of the first part (gravitation is in the global axis Y direction, therefore using the p_{12} position):

$$E_{P1} = m_1 \cdot g \cdot p_{12} \quad (9)$$

The kinetic and potential energy of the second part and the ball is found in the same way as for the first part. The total kinetic energy of the system:

$$E_K = E_{K1} + E_{K2} + E_{K3} \quad (10)$$

The total potential energy of the system:

$$E_P = E_{P1} + E_{P2} + E_{P3} \quad (11)$$

The Lagrangian:

$$L = E_K - E_P \quad (12)$$

The final motion equations:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} = T_1; \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\beta}} \right) - \frac{\partial L}{\partial \beta} = T_2; \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = 0; \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial L}{\partial y} = 0 \quad (13)$$

In which:

T_1, T_2, \dots the first and the second joint torque

Eq. (13) can be rewritten in the motion equations matrix form:

$$\mathbf{D}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{T} \Rightarrow \mathbf{D}(\alpha, \beta, x, y) \cdot \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} + \mathbf{H}(\alpha, \dot{\alpha}, \beta, \dot{\beta}, x, \dot{x}, y, \dot{y}) + \mathbf{G}(\alpha, \beta, x, y) = \begin{bmatrix} T_1 \\ T_2 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

The motion equations are derived generally and after determination of Eqs. (4) and (5), the problem is algorithmizable. Generally expressed motion equations are very extensive and because of that, for these

parameters:

$$h = 0,46m; a = 0,05m; b = 0,09m; c = 0,345m; bR = 0,02m; g = 9,81m \cdot s^{-2}$$

$$m_1 = 1,940kg; m_2 = 1,821kg; m_3 = 0,298kg$$

$${}^3r_3 = [-0,0009 \quad -0,00641 \quad 0,06504 \quad 1]^T; {}^4r_4 = [-0,00273 \quad 0,04798 \quad -0,00021 \quad 1]^T; {}^6r_6 = [0 \quad 0 \quad 0 \quad 1]^T \tag{15}$$

the specific solution for the system shown in Eq. (14) is:

$$D = \begin{bmatrix} D_{11} & D_{12} & \sin(\beta) \cdot (0,298y + 0,1027) & -5,96 \cdot 10^{-3} \cos(\beta) - 0,298x \cdot \sin(\beta) \\ D_{21} & 0,298x^2 + 1,329 \cdot 10^{-4} & -5,96 \cdot 10^{-3} & 0 \\ 1,49 \cdot 10^{-4} \sin(\beta) \cdot (2000y + 689) & -5,96 \cdot 10^{-3} & 0,298 & 0 \\ -5,96 \cdot 10^{-3} \cos(\beta) - 0,298x \cdot \sin(\beta) & 0 & 0 & 0,298 \end{bmatrix}$$

$$D_{11} = 0,2053y + 6,635 \cdot 10^{-5} \cos(2\beta) + 1,044 \cdot 10^{-6} \sin(2\beta) + 5,96 \cdot 10^{-3} x \cdot \sin(2\beta) - 0,149x^2 \cdot \cos(\beta) + 0,149x^2 + 0,298y^2 + 0,2038$$

$$D_{12} = 0,1027x \cdot \cos(\beta) - 5,791 \cdot 10^{-4} \sin(\beta) - 1,134 \cos(\beta) - 5,96 \cdot 10^{-3} y \cdot \sin(\beta) + 0,298x \cdot y \cdot \cos(\beta)$$

$$D_{21} = D_{12} \tag{16}$$

$$H = \begin{bmatrix} 0,2053\dot{\alpha} \cdot \dot{y} + 0,1027\dot{\beta} \cdot \dot{x} \cdot \cos(\beta) + 5,96 \cdot 10^{-3} \cdot \sin(\beta) \cdot (\dot{\alpha} \cdot \dot{x} - \dot{\beta} \cdot \dot{y}) \\ +0,596\dot{\alpha} \cdot [\dot{x} \cdot x + \dot{y} \cdot y + \dot{x} \cdot x \cdot \cos^2(\beta)] + 0,298\dot{\beta} \cdot \dot{x} \cdot y \cdot \cos(\beta) \cdot (\dot{y} \cdot x + \dot{x} \cdot y) 6,635 \\ \cdot 10^{-5}\dot{\alpha}^2 \cdot \sin(2\beta) - 1,044 \cdot 10^{-6}\dot{\alpha}^2 \cdot \cos(2\beta) - 0,01192\dot{\alpha} \cdot \dot{y} \cdot \sin(\beta) - 0,149\dot{\alpha}^2 \cdot x^2 \cdot \sin(2\beta) \\ +0,596x \cdot [\dot{\beta} \cdot \dot{x} + \dot{\alpha} \cdot \dot{y} \cdot \cos(\beta)] + +5,791 \cdot 10^{-4}\dot{\alpha} \cdot \dot{\beta} \cdot \cos(\beta) + \dot{\alpha} \cdot \dot{\beta} \cdot \sin(\beta) \\ \cdot (0,298x \cdot y - 1,134 \cdot 10^{-4}) + 5,96 \cdot 10^{-3}\dot{\alpha} \cdot [\dot{\beta} \cdot y \cdot \cos(\beta) - \dot{\alpha} \cdot x \cdot \cos(2\beta)] \\ +0,1027\dot{\alpha} \cdot \dot{\beta} \cdot x \cdot \sin(\beta) 0,596\dot{\alpha} \cdot \dot{y} \cdot \sin(\beta) - 0,298x [\dot{\beta}^2 + \dot{\alpha}^2 - \dot{\alpha}^2 \cos^2(\beta)] \\ -5,96 \cdot 10^{-3}\dot{\alpha}^2 \cdot \cos(\beta) \cdot \sin(\beta) - \dot{\alpha} \cdot \dot{\beta} \cdot \cos(\beta) \cdot (0,1027 - 0,298y) \\ 5,96 \cdot 10^{-3}\dot{\alpha} \cdot \sin(\beta) \cdot (\dot{\beta} - 1000\dot{x}) - 0,1027\dot{\alpha}^2 + 0,298\dot{\alpha} \cdot [\dot{\beta} \cdot x \cdot \cos(\beta) - \dot{\alpha} \cdot y] \end{bmatrix} \tag{17}$$

$$G = \begin{bmatrix} 5,066 \cos(\alpha) + 0,01713 \sin(\alpha) + 3,751 \cdot 10^{-3} \cdot \sin(\alpha) \cdot \sin(\beta) + 2,923y \cdot \cos(\alpha) \\ -9,699 \cdot 10^{-3} \cdot \cos(\beta) \cdot \sin(\alpha) - 2,923x \cdot \sin(\alpha) \cdot \sin(\beta) \\ 2,923x \cdot \cos(\alpha) \cdot \cos(\beta) - 9,699 \cdot 10^{-3} \cos(\alpha) \cdot \sin(\beta) - 3,751 \cdot 10^{-3} \cos(\alpha) \cdot \cos(\beta) \\ 2,923 \cos(\alpha) \cdot \sin(\beta) \\ 2,923 \sin(\alpha) \end{bmatrix} \tag{18}$$

4.3. Comparison of results

This mathematical model is compared with the physical model in Figure 8. As seen in the graph, the torque outputs from the mathematical model are very similar to the torque outputs from the physical model, even in the case of using mass point substitution. For the purposes of design control law, the mathematical model is acceptable because it sufficiently reflects the dynamic behavior of the system.

The physical model can be used for simulation, analysis, and testing of regulators and many others. It can be preferably used as a replacement for the real model, due to the high precision based on the direct use of the CAD model, the inclusion of friction, material elasticity, force barriers, and others. Simultaneously, the

physical model's connection with MATLAB provides an uncomplicated way for regulator verification in one software.

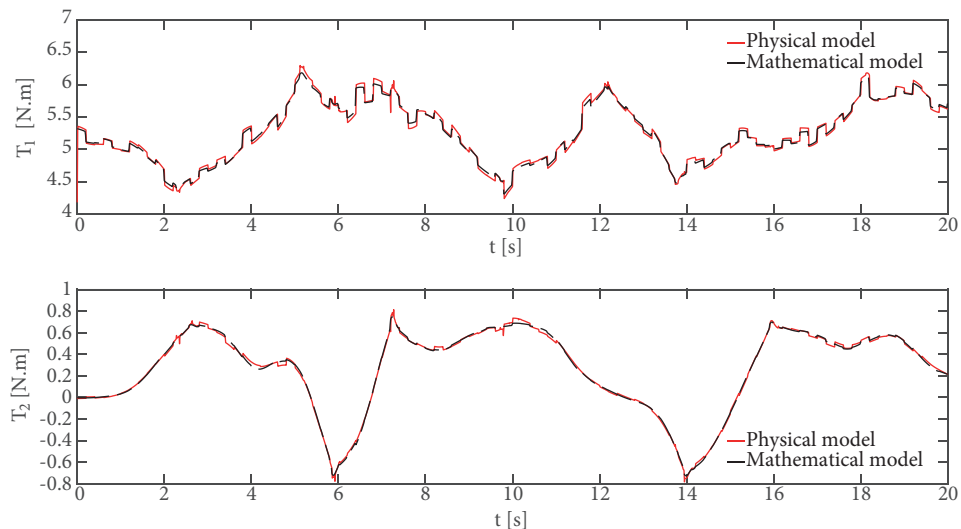


Figure 8. The physical and the mathematical model comparison.

5. Discussion

The differences between the presented paper and related works are in the physical (SimMechanics) model determination. Related papers [4,20] used SolidWorks SimMechanics export (or manual joints redrawing) with SimMechanics library mates, i.e. forces and torques between links are clearly defined, and no other force/torque needs to be added. In the case of the dynamic simulation in this paper, the system has 8 DOF, and only 2 DOF are controllable. There is no joint that includes the remaining 6 DOF concurrently with the restriction to 5 DOF when the ball touches the plate, 4 DOF/3 DOF when the ball touches the plate and one/two wall/walls, etc., not to mention friction, spring stiffness, and damping coefficient. Corresponding limiting forces/torques have to be added and the presented paper shows how to do it. The other possible uses of the physical model are also shown, not only a dynamic simulation.

Friction modeling is generally a complicated matter, and almost all of these types of mechanical structures do not take it into account. For accurate simulation, however, it is necessary to determine this friction, first by tabular properties of the material and then by measuring of the real model; the same applies to elasticity. Both parameters are crucial for accurate simulation of a rolling ball; nevertheless, they have been determined only by the tabular material properties. However, after completion of the real model, they will be experimentally measured and included in the physical model presented in this paper.

6. Conclusion

Effort in this work was devoted to dynamic simulation. The CAD model designed with SolidWorks is an essential part of accurate dynamic modeling. The CAD model is further used to manufacture the whole system; it means it had to be designed in any case.

After exporting the CAD model to the SimMechanics format, it was necessary to add a 6-DOF joint and determine all forces and torques that the plate actuates to the ball (except the gravity force). In this

dynamic simulation ideal rolling is not considered. The ball can also slide; the related friction characteristic is approximated by the combined arctangent function with constant value. The final physical model will be used as a partial substitution of a real (physics) model for setting the controller and appropriate actuators selection. The further application was in static analysis for improving a shape to increase the factor of safety. There are many other uses, but the main benefit of this particular solution is a direct connection to MATLAB/Simulink as a tool for control law design.

Motion equations are the first step in the design of the controller. The transformation matrices and the center of mass position vectors were determined from the CAD model. The motion equations were derived generally; the problem is algorithmizable with previously mentioned parameters. As shown in the graph, the mathematical model is acceptable even when using mass point substitution. The mathematical model suitability and the decision about mathematical model accuracy are the last mentioned application possibilities of the dynamic model.

The main added value of this paper is a different approach during physical (SimMechanics) model determination because of high system joints complexity and the impossibility of using standard methods. Equally valuable is the illustration of the various purposes of physical model usage.

References

- [1] *MathWorks. Simscape: Model and Simulate Multidomain Physical Systems. Natick, MA, USA: MathWorks, 2017.*
- [2] Şafak KK. *Dynamics, stability, and actuation methods for powered compass gait walkers. Turk J Elec Eng & Comp Sci 2014; 22: 1611-1624.*
- [3] Tlale NS, Zhang P. *Teaching the design of parallel manipulators and their controllers implementing MATLAB, Simulink, SimMechanics and CAD. Int J Eng Educ 2005; 21: 838-845.*
- [4] Ibrahim BSKK, Zargoun AMA. *Modelling and control of SCARA manipulator. Procedia Computer Science 2014; 42: 106-113.*
- [5] Güleç M, Yolaçan E, Demir Y, Ocak O, Aydın M. *Modeling based on 3D finite element analysis and experimental study of a 24-slot 8-pole axial-flux permanent-magnet synchronous motor for no cogging torque and sinusoidal back-EMF. Turk J Elec Eng & Comp Sci 2016 24: 262-275.*
- [6] Kudryavtsev AV, Laurent GJ, Cleve C, Tamadazte B, Lutz P. *Characterization of model-based visual tracking techniques for MOEMS using a new block set for MATLAB/Simulink. In: IEEE International Symposium on Optomechatronic Technologies; 5-7 November 2014; Seattle, WA, USA. New York, NY, USA: IEEE. pp. 163-168.*
- [7] Urrea C, Cortés J, Pascal J. *Design, construction and control of a SCARA manipulator with 6 degrees of freedom. J Appl Res Technol 2016; 14: 396-404.*
- [8] Ganesan E, Dash SS, Samanta C. *Modeling, control, and power management for a grid-integrated photo voltaic, fuel cell, and wind hybrid system. Turk J Elec Eng & Comp Sci 2016; 24: 4804-4823.*
- [9] Adam SAA, Ji-Pin Z, Yi-hua Z. *Modeling and simulation of 5DOF robot manipulator and trajectory using MATLAB and CATIA. In: 2017 3rd International Conference on Control, Automation and Robotics; 24-26 April 2017; Nagoya, Japan. New York, NY, USA: IEEE. pp. 36-40.*
- [10] Jagatheesa Perumal SK, Natarajan SK. *Investigation of adaptive control of robot manipulators with uncertain features for trajectory tracking employing HIL simulation technique. Turk J Elec Eng & Comp Sci 2017; 25: 2513-2521.*
- [11] Amadori K, Tarkian M, Ölvander J, Krus P. *Flexible and robust CAD models for design automation. Adv Eng Inform. 2012; 26: 180-195.*

- [12] Kocaarslan İ, Akçay MT, Ulusoy SE, Bal E, Tiriyaki H. Creation of a dynamic model of the electrification and traction power system of a 25 kV AC feed railway line together with analysis of different operation scenarios using MATLAB/Simulink. *Turk J Elec Eng & Comp Sci* 2017; 25: 4254-4267.
- [13] Solmaz S, Coşkun T. An automotive vehicle dynamics prototyping platform based on a remote control model car. *Turk J Elec Eng & Comp Sci* 2013; 21: 439-451.
- [14] Jonaitis A, Miliune R, Deveikis T. Dynamic model of wind power balancing in hybrid power system. *Turk J Elec Eng & Comp Sci* 2017; 25: 222-234.
- [15] Mattsson SE, Elmquist H. An overview of the modeling language Modelica. In: *Eurosim'98 Simulation Congress; 14-15 April 1998; Helsinki, Finland*. pp. 1-5.
- [16] Ferretti G, Magnani G, Rocco P. Virtual prototyping of mechatronic systems. *Annu Rev Control* 2004; 28: 193-206.
- [17] Tian F, Voskuyl M. Automated generation of multiphysics simulation models to support multidisciplinary design optimization. *Adv Eng Inform* 2015; 29: 1110-1125.
- [18] Rocca GL, Tooren MJLV. Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. *Journal of Design Research* 2007; 5: 333.
- [19] Barth M, Fay A. Automated generation of simulation models for control code tests. *Control Eng Pract* 2013; 21: 218-230.
- [20] Mostyn V, Skarupa J. Improving mechanical model accuracy for simulation purposes. *Mechatronics* 2004; 14: 777-787.
- [21] Udai AD, Rajeevlochana CG, Saha SK. Dynamic simulation of a KUKA KR5 industrial robot using MATLAB SimMechanics. In: *15th National Conference on Machines and Mechanisms; 30 November-2 December 2011; Chennai, India*. pp. 1-8.
- [22] Makkonen T, Nevala K, Heikkilä R. A 3D model based control of an excavator. *Automat Constr* 2006; 15: 571-577.
- [23] Gouasmi M, Ouali M, Fernini B, Meghatria M. Kinematic modelling and simulation of a 2-R robot using SolidWorks and verification by MATLAB/Simulink. *Int J Adv Robot Syst* 2012; 9: 245.