Chapter 20

Analysis of Robotic System Motion in SimMechanics and MATLAB GUI Environment

Viliam Fedák, František Ďurovský and Róbert Üveges

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/58371

1. Introduction

Robots present considerably complicated electromechanical systems with mutual interactions of robot mechanics and drives, at design of which the mechatronic approach should be taken into consideration. The computer modeling presents such basic tool for mentioned mechatronic approach. When designing control of a robot, we need to know necessary torque and angle of rotation of each motor, to visualize behavior of the robot, and to obtain mathematical model of each part. Generally, this inverse kinematic task is not solvable analytically and the numerical calculation often entails difficulties. The design of a control law for the drive system is also connected with the need of transfer function derivation and with simulation of dynamical properties of the robot mechanical system as a whole.

The physical modeling in the SimMechanics environment [1] considerably facilitates simulation efforts of complex mechanical systems regardless of their complication by elastic and damping elements and by number of degrees of freedoms. The SimMechanics program scheme having the form of interconnected blocks shows how the physical components with geometric and kinematic relationships of the robot are mutually interconnected. The SimMechanics program enables one to model mechanical systems by bodies and joints, to simulate their motion, to change easily the structure, to optimize system parameters, and to analyze results all within the Simulink environment. This approach does not require cumbersome deriving differential equations of the system and presents an easy and fast way to obtain the dynamic model of the system and saves time and effort.

Several approaches for the robot model development in the SimMechanics environment that appeared in last years are known worldwide, e.g. [3]-[7]. The robot models were developed on basis of the robot configuration. To simplify the development task we have used a special feature announced by The MathWorks in 2003 which integrates SimMechanics with The SolidWorks CAD Assemblies. [2]. Mathworks collaboration with Solidworks Corporation extended the engineering analysis capabilities of SimMechanics by allowing seamless integration of Solidworks CAD Assemblies into the SimMechanics simulation and design environment. This means that the SolidWorks models can be simulated in the Simulink environment in order to analyze forces and torques in mechanical joints, plot accelerations and displacements of each part of the system, to visualize motion of the CAD assembly, while taking into consideration masses of individual objects. This facility is enabled by installing an appropriate plug-in in SimMechanics which imports the 3D CAD model of the full system with bodies, joints, couplings, and masses from the SolidWorks program into the SimMechanics for further work with the model.

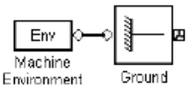
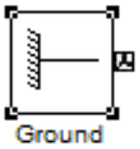
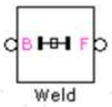
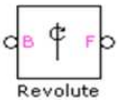
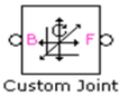
The objective of the proposed chapter is to present an application of effective way of the robot mechanics modeling and its dynamic simulation using add-on modules of the MATLAB advanced environment: SimMechanics and Graphical User Interface (GUI) MATLAB.

A complete description of the procedure is presented on example of a SEF-ROBOTER SR 25 type welding robot [1] weighing 480 kg with payload of 25 kg. Calculations were performed in the MATLAB/SimMechanics environment that enables a simple physical modeling of mechanical systems without any necessity of motion equations derivation. After this a 3D CAD model of the robot mechanics was developed using SolidWorks program. This procedure also enables verification of the model-whether it corresponds to the reality and whether it behaves according to the presumptions and requirements. By importing the 3D CAD model into SimMechanics a basic simulation scheme is obtained. After completing it by few blocks the simulation scheme is utilized for analysis in both direct and inverse kinematics tasks. Finally, a GUI model of the robot system has been developed that enables one to perform various virtual experiments and to obtain required outputs: position (angles), forces and torques. The GUI also solves analyzing of the inverse and direct kinematics tasks.

2. Employed SimMechanics blocks

SimMechanics contains a set of block libraries and special simulation interfaces (Sensor and Actuator blocks) for interconnection of the SimMechanics scheme with the Simulink environment. The SimMechanics blocks present elements enabling to model mechanical systems consisting of rigid bodies connected by joints that represent translational and rotational degrees of freedom. SimMechanics automatically sets up a single absolute inertial reference frame and coordinate system (CS) called World, [9]. For easier interpretation of the following block diagrams, the function of used SimMechanics blocks is shown in Table 1.

http://dx.doi.org/10.5772/58371

Group	Block	Name	Description
Bodies		Env	The block *Machine Environment* defines environment for calculation of the scheme. Each SimMechanics model contains one such block that is connected with the block *Ground*. Except of inputting the precision of calculation and parameters of the environment, the required analysis type can be set up: • *Forward Dynamics* – based on initial values and forces in the system the program calculates values of positions and speeds. • *Linearization* – this mode calculates the system linear model. • *Trimming* – finds the machine steady states. • *Inverse Dynamics* for open loop In this mode the SimMechanics calculates forces necessary for performing the motion forced by kinematic excitation. • *Kinematics* does the same for closed loop systems by including extra internal invisible constraints arising from those structures.
		Ground	The *Ground* block represents a fixed point having infinite mass. At least one block *Ground* connected with the *Machine Environment* must be involved.
		Body	The block *Body* in SimMechanics replaces all fixed rigid bodies among which the degrees of freedom are added. The bodies are defined by their final and non-zero masses, inertia, positions, directions, and by coordinate systems that are connected to them.
Joints		Weld	The blocks *Joints* interconnect blocks of the *Body* type and they are added degrees of freedom. The blocks determine direction and type of motion. In difference to the physical joints, in SimMechanics they present massless bodies and a physical connection of the bodies is not required. In the block there are shown ports: *B-Base* and *F-Follower* what means, the *Follower* performs a motion regarding to the *Base*. The block *Weld* – means a body without any degree of freedom.
		Revolute	The *Revolute* block from the *Joints* group represents one degree of freedom (rotation).
		Custom Joint	The *Custom Joint* is developed by the user using so called primitives (*Joint Primitives*) that create degrees of freedom. The motion performed by the *Custom Joint* is done in the order prescribed by the list of primitives.

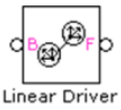
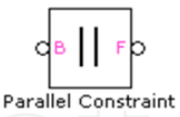
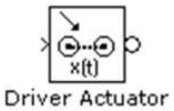
Drivers & Constraints		Linear Driver	Linear Driver – is used to define the distance between *Base* and *Follower* following x, y or z axis of the *World*.
		Parallel Constraint	The *Parallel Constraint* block ensures that vectors of axes of two bodies are parallel.
Sensors & Actuators		Driver Actuator	*Sensors and Actuators* are the blocks used as interfaces between non-SimMechanics Simulink blocks and SimMechanics blocks. By the *Actuators* it is possible to transform a Simulink signal into physical one actuating the bodies in SimMechanics diagram. The *Sensors* preform reverse functions – they transform signal from SimMechanics into Simulink environment. These blocks can be connected to *Joints*, *Drivers* and *Constraints* (only *Sensors*) into special purpose- oriented ports. In *Bodies* they are connected directly to the chosen CS. Outputs from the sensors are: positions, speeds, accelerations, reaction forces, etc.

Table 1. Description of functions of the used blocks in the SimMechanics program

3. Robot model development

3.1. 3D model of the robot

Development of dynamic model of a robot starts by identifying its parameters: based on technical specifications from the producer and completing them by own measurement of dimensions and distribution of masses depending on form of robot arms. Some missing data were estimated and approximated and the complex geometric forms were replaced by equivalent and simpler ones. Thus we gradually increased the model precision, position of centers of gravity and masses and where not possible, we used approximated data. In every case, the precise identification of model parameters is a laborious and trial-and error process, [2].

Fig. 1 shows the analyzed robot – both its picture and sketch showing possible movements of the robot joints. Based on measured and estimated data of the robot mechanical subsystem a precise technical drawing (compare the 2D sketch in Fig. 1b) create a basis for 3D modeling of the robot. By use of the SolidWorks program a 3D CAD model has been developed (Fig. 2).

Modeling of separate bodies and joints in the SolidWorks is much more advantageous because based on animation in the SolidWorks one can simultaneously verify correctness of kinematics of the mechanical model. The imported model also contains masses of the bodies, centers of inertia, tensors of the inertia and graphics that will be used at visualization. Of course, the forms of the bodies have been simplified but the time responses of such electromechanical

http://dx.doi.org/10.5772/58371

system (i.e. the mechanical system completed by the drive subsystem) lie within acceptable boundaries.

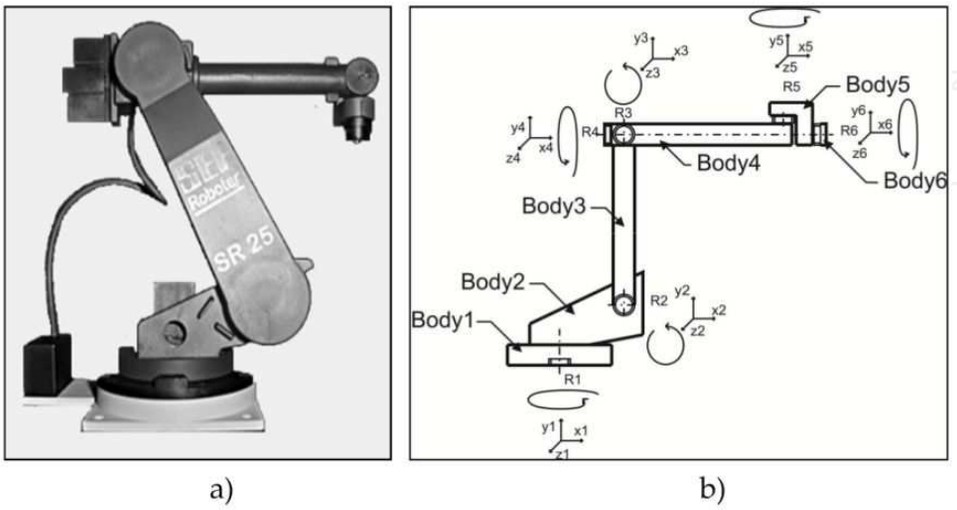


Figure 1. Picture of the analyzed robot (a); and its 2D sketch (b)

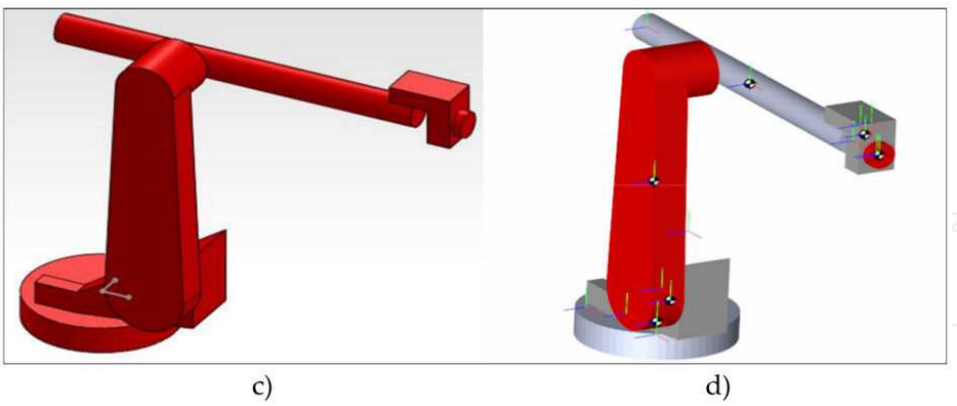


Figure 2. Visualization of the robot imported from the 3D model into the SolidWorks program

3.2. Development of the program scheme in SimMechanics

The model in SimMechanics is built in the following steps:

1. Specification of body inertial properties, degrees of freedom, and constraints, along with coordinate systems attached to bodies to measure positions and velocities.
2. Applying forces/torques and completing the model by sensors and actuators to initiate and record body motions.
3. Starting the simulation, calling the Simulink solvers to find motions of the system, while maintaining any imposed constraints.
4. Visualizing the machine while building the model and animation of the running simulation, applying the Graphical User Interface (i.e. using Handle Graphics) or a virtual reality visualization tool.

The 3D CAD model developed in the SolidWorks program is imported into the SimMechanics suitable format using the SimMechanics link that creates a basic mechanic chain completed by system parameters. The program scheme after importing is shown in Fig. 3. For a better relation with the robot parts also sketches of particular bodies are shown.

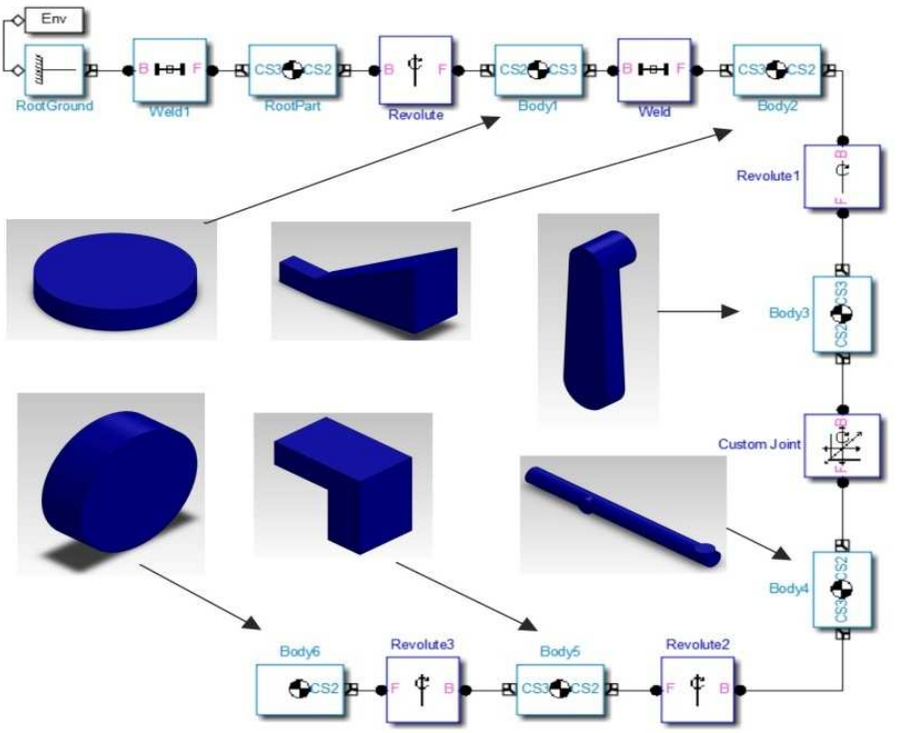


Figure 3. Program scheme of the 6-DOF robot in SimMechanics with appropriate bodies (mechanical parts - arms of the robot) and their corresponding blocks.

This basic SimMechanics scheme is completed by couplings, joints, sensors, connections and further blocks for the inverse kinematic task. The completed scheme is shown in Fig. 4.

http://dx.doi.org/10.5772/58371

4. Robot inverse kinematics

In inverse kinematic task the angles of joints (corresponding to angles of rotation of the driving motors) are calculated, [9]. In case of the robot they are necessary for obtaining a required trajectory (position, orientation and trajectory of the effector reference point). Due to nonlinearities of the mechanical system the analytic solution of the inverse kinematic task presents a quite difficult and complex problem, solution of which cannot be usually obtained in a closed mathematic formula. Although the solution with help of SimMechanics considerably facilitates the task, the problems with existence of solution and ambiguity still persist.

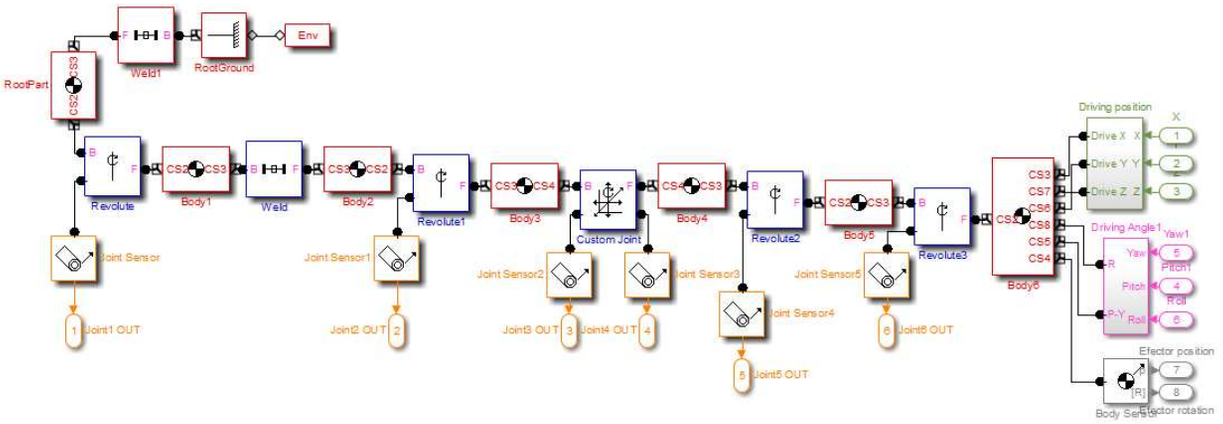


Figure 4. Program scheme – inverse kinematics of the analyzed robot mechanics

4.1. Solution of inverse kinematic task

In this case the block Joint Actuator is not connected to the joints because the system is completely and exactly defined by position and orientation of the effector. This assumption enables us to change calculation mode in the block Machine Environment to the "Kinematics". The choice also increases speed of simulation.

Position of the effector is defined in relation to the base point using the constraint *Linear Driver* (Fig. 5). It defines the distance along each axis of the {x, y, z} coordinate system (the vector basis).

Orientation of the effector is defined by a reference body and the *Parallel Constraint* of the x_e-axis to the x_b-axis of the reference body. Defined here are:

- *Pitch* (turning round z_b-axis) and
- *Yaw* (round y_b-axis).

The y_e-axis of the effector runs parallel with y_b-axis of the reference body where defined is the:

- *Roll* (turning round the x_b-axis).

4.2. Completing the program scheme

Position of the vector is defined by the *Driving position* subsystem (Fig. 5), determining effector coordinates towards its initial (baser) point.

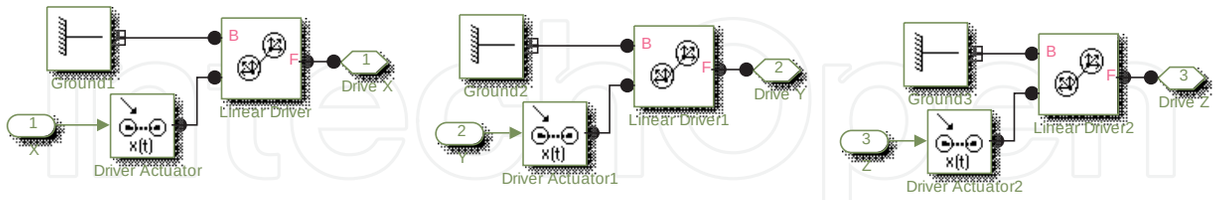


Figure 5. Subsystem Driving position

Orientation of the coordinate system of the effector terminal point is determined by the *Driving angle* subsystem. Its outputs are angles of the effector swinging in x-y-z axes in respect of initial conditions. Let's remind that the *effector axes* (Body 6 in Fig. 3) are in parallel with the *reference body axes* (Body 2, Fig. 6).

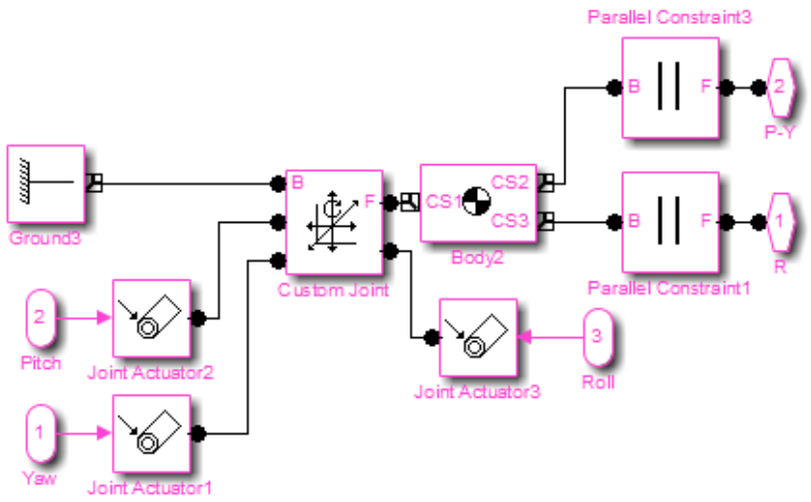


Figure 6. Driving angle" Subsystem

The inputs and outputs of the complete model of robot mechanical part in SimMechanics is shown in Fig. 7.

http://dx.doi.org/10.5772/58371

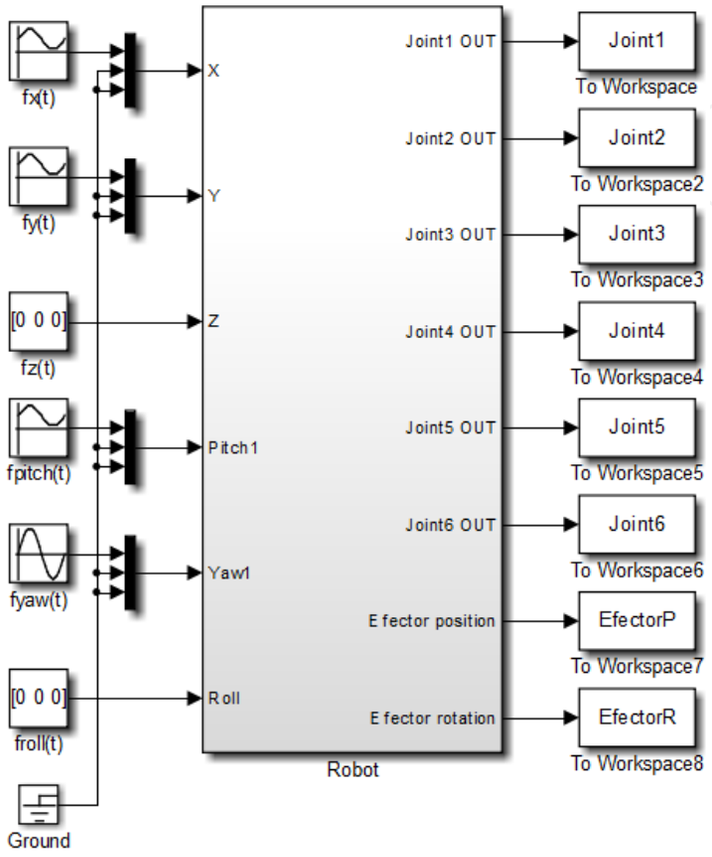


Figure 7. Robot model input and output signals

4.3. Simulation results

As an example for verification of the model properties an effector trajectory was programmed as a circle in the x-y plane with the period of 2π. In this case, the trajectory is easily described by simple mathematical functions which task is unambiguous and solvable:

$$f_x(t)=0.2\sin\left(t+\frac{\pi}{2}\right)-2,\ f_y(t)=0.2\sin(t)-1,\ f_z(t)=0$$

$$f_{pitch}(t)=10\sin(t+\pi)+85,\ f_{yaw}(t)=10sin\left(t+\frac{\pi}{2}\right),\ f_{roll}(t)=0$$

Its trajectory is shown in Figs. 8 and 9. Final time courses of turning the joints 1-6 obtained from the inverse kinematics scheme are shown in Fig. 10 (from top to bottom the courses belong to joints 6, 5, etc.). Several experiments have shown that the obtained results can be utilized in the forward kinematics.

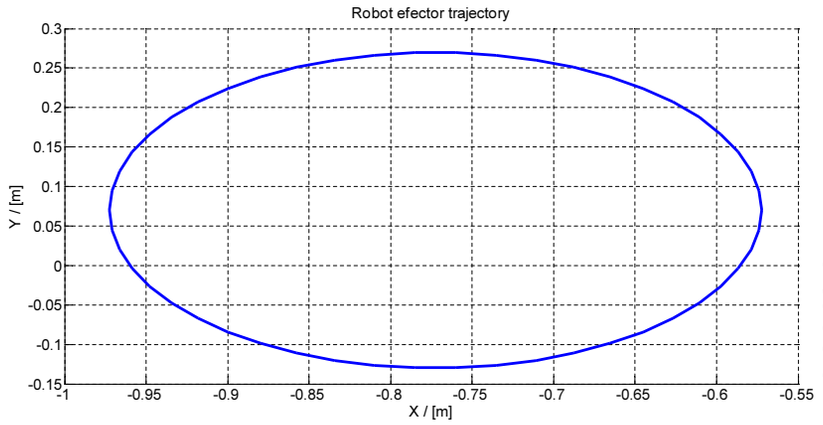


Figure 8. Desired effector trajectory in the x-y plane

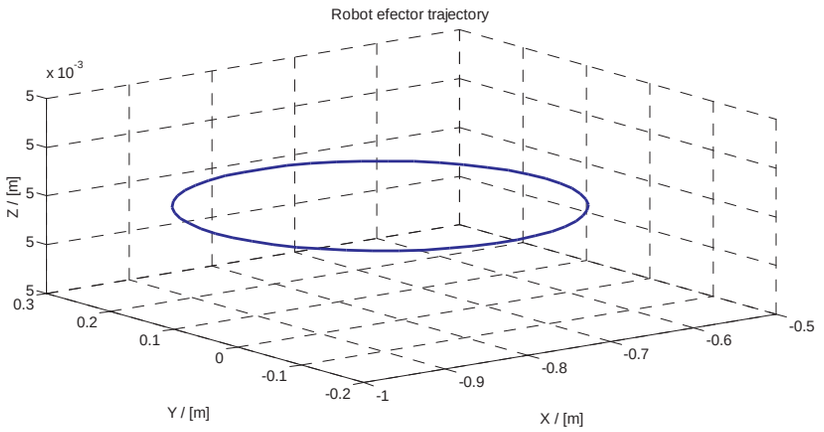


Figure 9. Desired trajectory of the effector reference point in 3D space

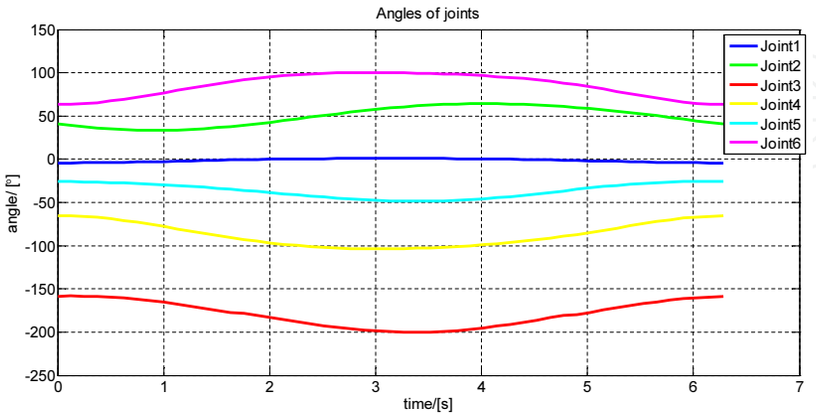


Figure 10. Time courses of angles of the joints to get the desired trajectory according to Fig. 9

http://dx.doi.org/10.5772/58371

5. Forward kinematics and inverse dynamics of the robot

The *Forward Kinematics* presents a basic problem at robot motion solving. Its result consists of geometric calculation of position of coordinates and orientation of the robot effector towards the basic coordinate system when rotating the joints by certain angles. In the *Forward Dynamics* mode, the SimMechanics uses the Simulink suite of ordinary differential equation (ODE) solvers to solve the mechanical ODEs (i.e. Newton's equations). The *Forward Kinematics Task* is simpler than the *Inverse Kinematics Task* but here, in the paper, it is described in the second place because the calculated values from the previous task can be easily used and verified.

5.1. Program scheme for forward kinematics

In this case, controlled are the angles of joints and the position and sensed is orientation of the effector. The blocks *Joint Actuator* will operate in the mode of kinematic excitation (Motion). The mode of calculation in the block *Machine Environment* is set to the *Inverse Dynamics*. The *Kinematics* mode can be used only in case of closed kinematic chain.

By connecting sensors to particular joints it is possible to determine the torques of the motors which are necessary for performing a defined trajectory of the effector. This scheme can be used also at solution of the system dynamics.

Note: During simulation an abnormal situation can occur: some parts of the block scheme behave abnormally. It is caused by the *Joint Sensor* block that senses angles of the joints within the range of $\langle -180°; 180° \rangle$. After crossing these values there appears a step change of the sensor output value from one marginal value to the other. This phenomenon can be avoided by employing the *unwrap* instruction adapting the step change signal into a smooth continuous form.

5.2. Simulation results

The trajectory in Fig. 12 calculated by the *Forward Kinematic* analysis corresponds to the trajectory of the effector from the Inverse Kinematic task presented above. Time responses of the torques in the robot joints 1 – 6 that are necessary for development of the effector trajectory are shown by graph in Fig. 13.

Graphs of the torques of the motors (Fig. 13) are as follows: the joints 5 – 6 on the top; 4 – 3 in the middle; joints 2 – 1 at the bottom (the coordinate systems are referred to Fig. 1b).

The torques in Fig. 13 depend strongly on the system parameters – masses, friction in the joints, moments of inertia of the motors. Eventual discrepancies between the simulated time responses and the real system may have their roots just here.

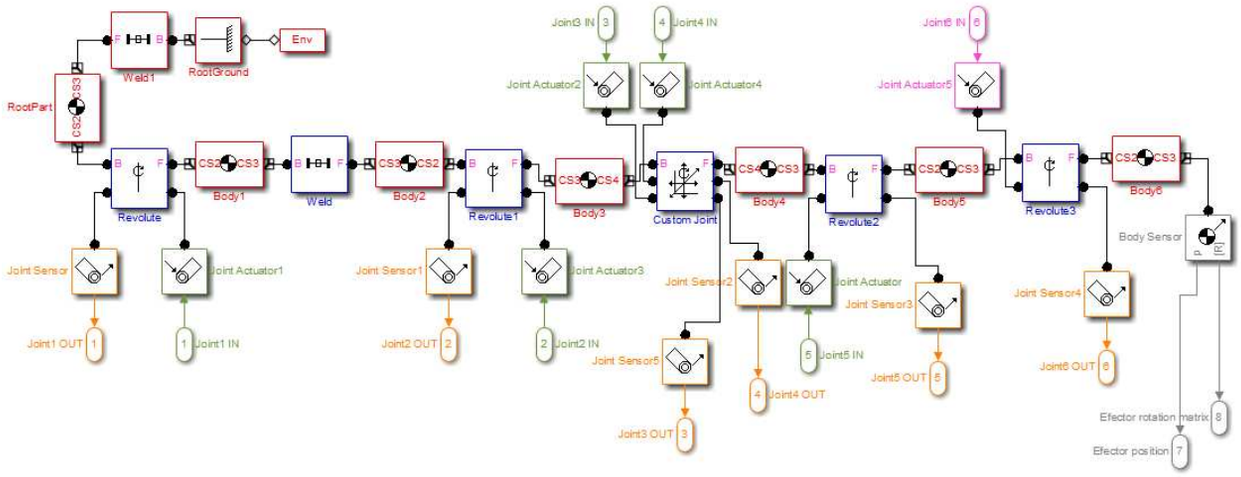


Figure 11. Program scheme – forward kinematics for the analyzed robot

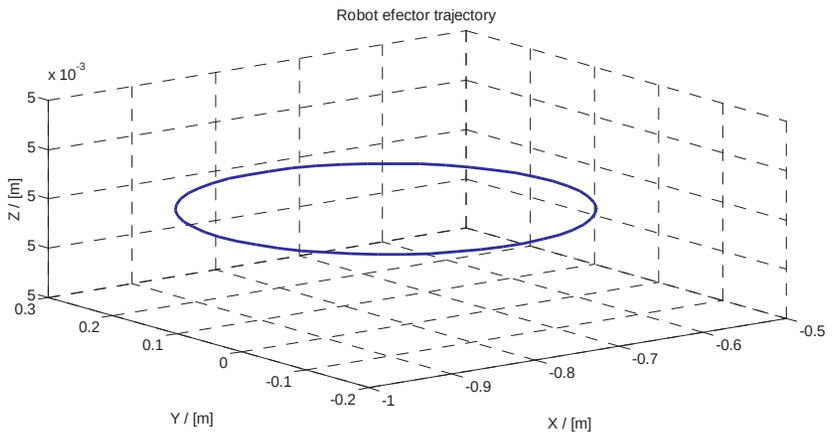


Figure 12. Robot effector trajectory in 3D space calculated by the direct kinematic task

6. Design of graphical user interface

To simplify experimentation work and to obtain a better view of the system behavior a *Graphical User Interface* (GUI) was developed in the MATLAB environment using MATLAB's *Graphical User Interface Development Environment* (GUIDE) tool. For a chosen set of parameters the GUI performs simulation showing time responses of the angles of joints, torques in joints and position of the robot effector. Various possibilities of the GUI modes and the appropriate screens are shown in Fig. 14 and Fig. 15.

Using switches in the **Mode group** the user chooses the required task type:

- Forward Kinematics,
- Inverse Kinematics,

http://dx.doi.org/10.5772/58371

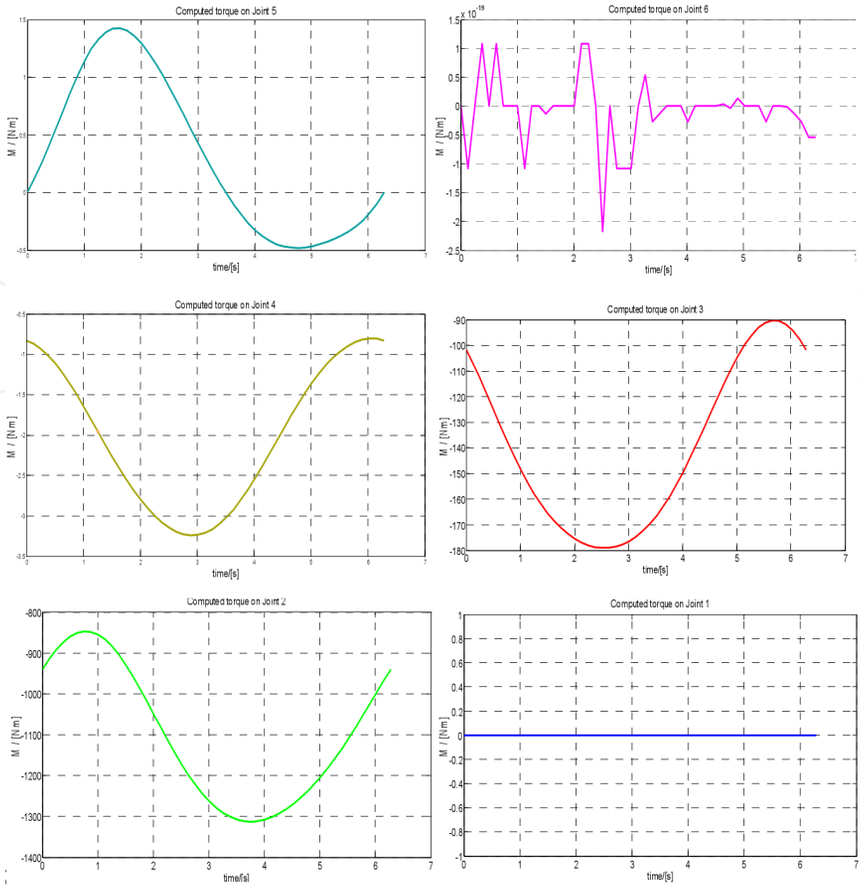


Figure 13. The torques in the robot particular joints

- Forward Dynamics,
- Inverse Dynamics.

7. Program in the inverse kinematics mode

In this mode the user defines trajectory of the effector motion. Choices include a possibility of entering position and orientation or eventually their derivations, both by entering mathematical functions into the text boxes *Effector rotation* and *Effector position*, including variable of the time, denoted as "t". This way of entering the required trajectory is impractical for the use and here it serves for demonstration purposes only.

For practical solution the program was modified by adding a possibility of importing the set of coordinates and speeds of the effector from a table developed in an Excel file. Based on the points of the trajectory the program calculates the trajectory of the effector movement. The speed among the points is considered to be constant. The results of the described simulations are shown in Fig. 14 (the right window). In the **Effector position window** displayed is the chosen trajectory consisting of sections of straight lines and in the right **Angles of joints**

window there are joint angles that are necessary to generate the prescribed trajectory of the effector in the 3D space.

7.1. Program in the forward kinematics mode

In this mode the user defines joint variables as mathematical functions written into the text boxes that are shown in the frame **Angles of Joints** (Fig. 15). The angles of the joints can be defined by the angle, angular velocity and angular acceleration (a choice there).

Based on given functions the program calculates the trajectory of the effector in the 3D space (shown in the window on the left side of the screen) and time responses of required angles of joints 1 – 6 in the graphs on the right side there.

7.2. Program in the inverse and direct dynamics mode

SimMechanics can solve the reverse of the forward dynamics problems: instead of starting with given forces/torques and finding the resulting motions, the Inverse Dynamics mode determines the forces/torques needed to produce a given set of motions that you apply to the machine. This mode only works with open topology systems (model diagrams without closed loops).

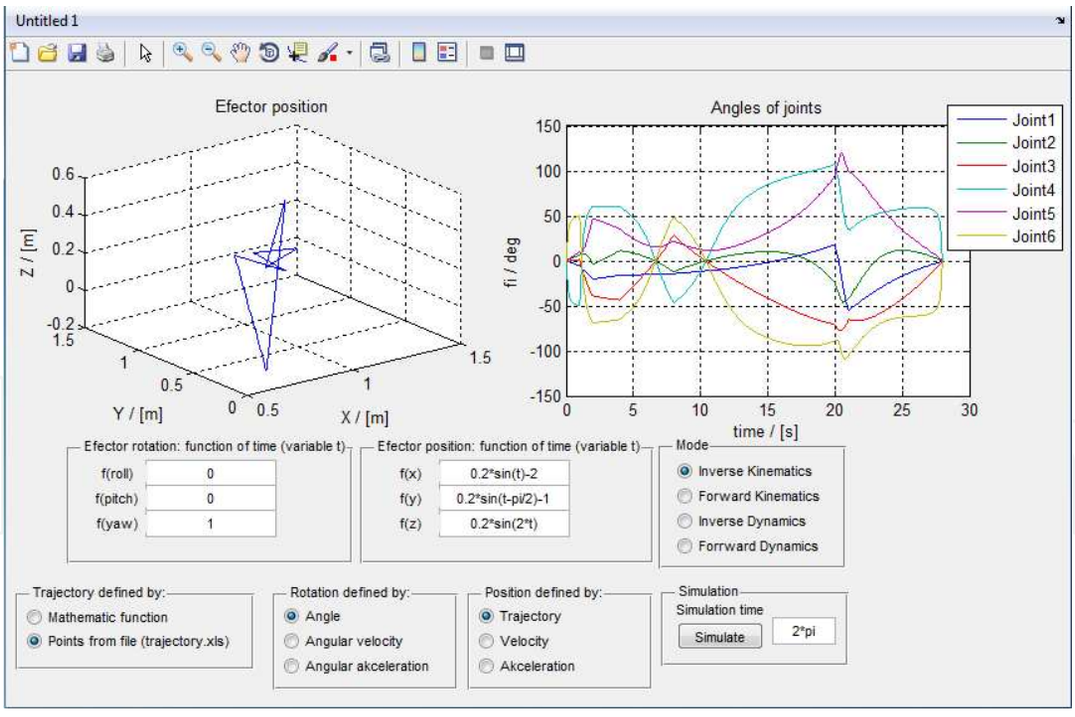


Figure 14. GUI for the inverse kinematics of the robot

When switching the program into the **Inverse Dynamics mode** the interface environment remains unchanged. The only change is that after simulation the time course of each torque is displayed on display in the right window.

http://dx.doi.org/10.5772/58371

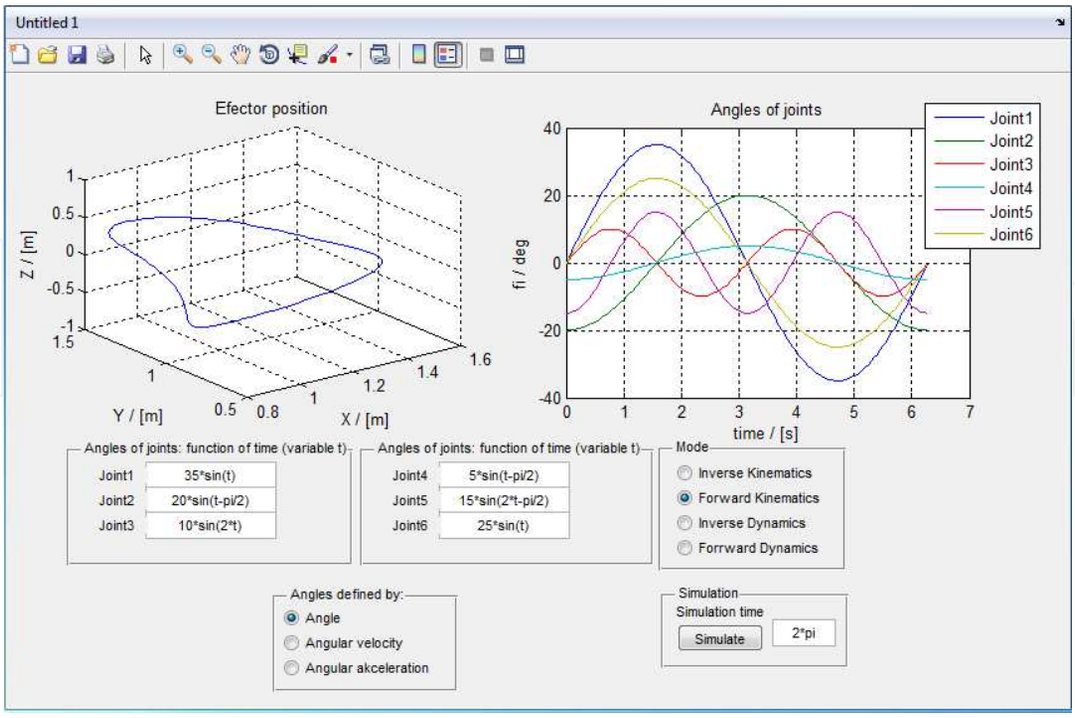


Figure 15. GUI for the direct kinematics of the robot

In the **Direct Dynamics mode** there are entered time courses of the torques acting on the particular joints. The required values of the torques are displayed in the left window and the calculated time responses of the angles appear in the right window.

8. Conclusion

SimMechanics presents a powerful tool for modeling mechanics of rigid bodies. It is suitable for modeling of dynamics and kinematics of considerably complicated systems with many joints without using any mathematical description. For these advantageous properties it is often used in the first phase of designing robotic systems, esp. due to simplicity of changing parameters and dimensions of particular bodies without necessity to repeat design of new model.

To get a dynamical model for the systems with more complex bodies of various shapes and connections connected through joints it is advantageous to model them in a compactible 3D CAD software and then to import the model from the program into the SimMechanics program. This solution enables us to avoid analytical calculations, esp. in case of the of moment of inertia tensor of an irregular body. The CAD software automatically calculates this tensor and moreover it adds visualization to the developed model.

The described development of the 3D CAD space model of the robot in the SolidWorks program and its import into the SimMechanics facilitates the work at developing the model of

robot mechanics. This procedure also reduces a possibility of error occurrence at modeling the system.

A drawback in using the SimMechanics program for simulation consists in solution of collision of bodies and in limitation of angles of joints (what makes difficulties esp. at inverse kinematics task). In the mode of dynamics it is possible to avoid this disadvantage by introducing a feedback with a spring having a high stiffness and damping that is activated after exceeding set boundaries. In the mode of forward kinematics the required value of joint variables can be easily limited. In case of inverse kinematics it is possible to complete the scheme by suitably chosen parallel kinematic chains not allowing any motion out of preset range. Disadvantage of all described solution consists in slower simulation due to the added bodies and blocks.

At present time the developed model serves for further research – forward kinematics, analysis of dynamics and finally - for design of controllers for joint drives. The future research will concern completing the developed graphical user interface by advanced algorithms taking into consideration advanced algorithms of the trajectory design, e.g. curved line with considering obstacles, maximal reachable speed, verification of reachability according to limitation of the particular joints and dimensions of the arms. Within framework of the planned Hardware in the Loop system we plan to interconnect the master computer the SimMechanics program with a real robot of the SF25 type where on the first computer will run the SimMechanics program serving together with Simulink as a generator of controlling signals in the mode of inverse dynamics together with RT-LAB (a real time digital simulation software from Opal Technologies). Through a CAN bus the second computer will control frequency converters (of the SINAMICS CU 320 type from SIEMENS) supplying electrical drives of the robot.

Acknowledgements

The work was supported by Slovak Cultural and Educational Agency of the Ministry of Education of Slovak Republic under the contract KEGA 042TUKE-4/2012 “Teaching Innovation in Control of Mechatronic Systems”.

Author details

Viliam Fedák*, František Ďurovský and Róbert Üveges

*Address all correspondence to: Viliam.Fedak@tuke.sk

Department of Electrical Engineering and Mechatronics, FEEaI, Technical University of Kosice, Slovakia

References

[1] SimMechanics-Model and Simulate Multibody Mechanical Systems. http://www.mathworks.com/products/simmechanics/ (accessed 15 February 2014).

[2] The MathWorks Integrates SimMechanics with SolidWorks CAD Assemblies. http://www.embeddedstar.com/press/content/2003/5/embed ded8871.html (Accessed 15 February 2014).

[3] Shanoiqiang Y., Zhong L., Xingshan L., Modeling and Simulation of Robot Based on Matlab/SimMechanics. In: The 27th Chinese Control Conference, 16-18 July 2008, Kunming, Yunnan, China, pp. 161-165.

[4] Vavrinčíková V., Hroncová D., Modeling of Robot Dynamics in the SimMechanics Environment. ATP Journal PLUS, "Intelligent Motion Systems", 2009, 60-64, ISSN 1336-5010. (in Slovak)

[5] Hanchen L., Xinhua Z., Haoliang X., Modeling and Simulation of 3-RRRT Parallel Manipulator Based on MATLAB with SimMechanics. In: Proceedings of 2009 IEEE Int. Forum for Information Technology and Applications IFITA '09, 15-17 May 2009, Chengdu, China, pp. 290-293.

[6] Dung Le Tien, Kang Hee-Jung, Ro Young-Shick, Robot Manipulator Modeling in Matlab-SimMechanics with PD Control and Online Gravity Compensation. In: Proceedings of the 5th IEEE International Forum on Strategic Technology, IFOST 2010, 13-15 Oct. 2010, 4p.

[7] Boros T., Lamár K., Six-axis Educational Robot Workcell with Integrated Vision System. In: Proc. of 4th IEEE International Symposium on Logistics and Industrial Informatics "LINDI 2012", Smolenice, Slovakia, 2012, pp.239-244. ISBN 978-1-4673-4518-7.

[8] SEF Roboter SR 25. http://www.youtube.com/watch?v=disu_RAoBAY. (Accessed 15 February 2014)

[9] Grepl R., Modeling of Mechatronic Systems. Praha, BEN, 2007, ISBN 978-80-7300-226-8. (in Czech).

IntechOpen