Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 1, 2016, vol. LXII
article No. 2005

Vladimir BULEJ[*], Juraj URICEK[**], Manfred EBERTH[***]

THE MODELLING OF MECHANISM WITH PARALLEL KINEMATIC STRUCTURE IN SOFTWARE MATLAB/SIMULINK

MODELOVANIE MECHANIZMU S PARALELNOU KINEMATICKOU ŠTRUKTÚROU V SOFTVÉRI MATLAB/SIMULINK

Abstract

The article deals with the preparation of simulation model of mechanism with parallel kinematic structure called hexapod as an electro-mechanical system in software *MATLAB/Simulink*. The simulation model is composed from functional blocks represented each part of mechanism's kinematic structure with certain properties. The results should be used for further simulation of its behaviour as well as for generating of control algorithms for real functional prototype.

Abstrakt

Článok sa zaoberá modelovaním mechanizmu s paralelnou kinematickou štruktúrou nazývaného hexapod ako elektromechanického systému v prostredí softvéru *MATLAB/Simulink*. Simulačný model je zložený z funkčných blokov, ktoré predstavujú jednotlivé časti kinematickej štruktúry mechanizmu s určitými vlastnosťami. Výsledky môžu byť použité pre budúcu simuláciu jeho správanie, ako aj pre generovanie riadiacich algoritmov pre reálny funkčný prototyp.

Keywords

Simulation model, Stewart platform, MATLAB/Simulink, PKS, hexapod.

1 INTRODUCTION

During the last decade, there was formed a group of researchers led by professor Poppeova at the Faculty of Mechanical Engineering at University of Žilina, which deals with the parallel kinematic structures (PKS) [9]. There have been proposed various design concepts of the mechanisms with parallel kinematic structure as well as different kind of simulation software for these mechanisms within this period (figure 1 - right). One of designed prototypes is machine tool (figure 1 - left) based on mechanism Hexapod equipped with automatic tool changing and automatic part changing system. Hexapod, also known as Stewart platform, is multi-axis manipulator or moving platform capable of full six degrees of freedom (DOFs) motion [7]. This article contains the basic information about the first steps in modelling and design of simulation model of mechanism mentioned above in environment of software *MATLAB/Simulink*. Our main aim is to create the model for calculation and simulation of hexapod prototype in the software *MATLAB/Simulink*.

[*] Ing., PhD., Department of Automation and Production Systems, Faculty of Mechanical Engineering, University of Zilina, Univerzitna 8215/1, Zilina, Slovak Republic, phone. (+421) 41 513 2811, e-mail vladimir.bulej@fstroj.uniza.sk

[**] Assoc.-prof., Ing., PhD., Department of Automation and Production Systems, Faculty of Mechanical Engineering, University of Zilina, Univerzitna 8215/1, Zilina, Slovak Republic, phone. (+421) 41 513 2813, e-mail juraj.uricek@fstroj.uniza.sk

[***] Dipl.-Ing.(FH), MSc., AUDI AG, I/EE-554, D-85045 Ingolstadt, phone. (+49) 841 89 42332, e-mail manfred.eberth@audi.de

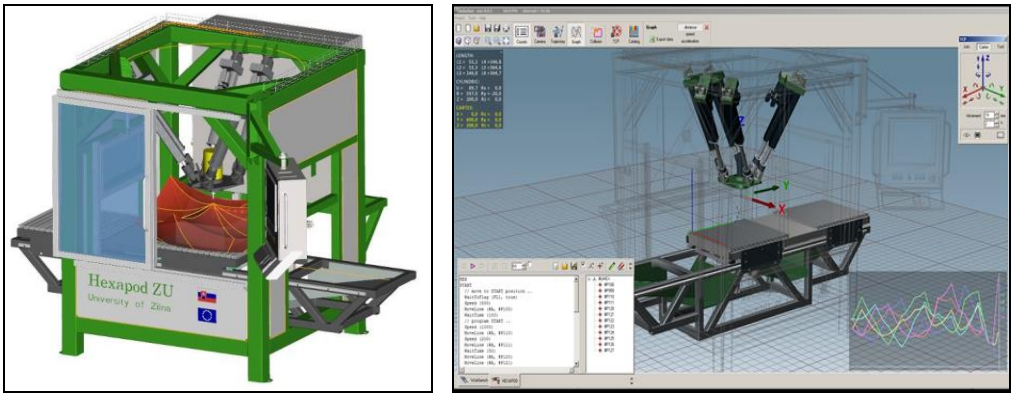

Fig. 1 Prototype of machine tools based on Hexapod mechanism and second version of simulation software designed in Delphi programming language [2]

2 SIMULATION MODEL OF MECHANISM HEXAPOD

At the beginning it was necessary to prepare an appropriate 3D model of selected mechanism for what we used software *ProEngineer* and *SolidWorks* as well. Subsequently, the 3D model was converted to a format that can be processed by the program *MATLAB/Simulink*. Finally, the simulation model of Stewart platform was developed with the help of Simulink libraries. The transformation of mechanism's 3D model into the functional schema readable by *MATLAB/Simulink* can be considered as the first step of whole process. The distribution of assembly components into the individual subassemblies is needed with respect to the organization of the whole assembly and especially with respect to minimizing the functional blocks at its transfer into the *MATLAB/Simulink*.

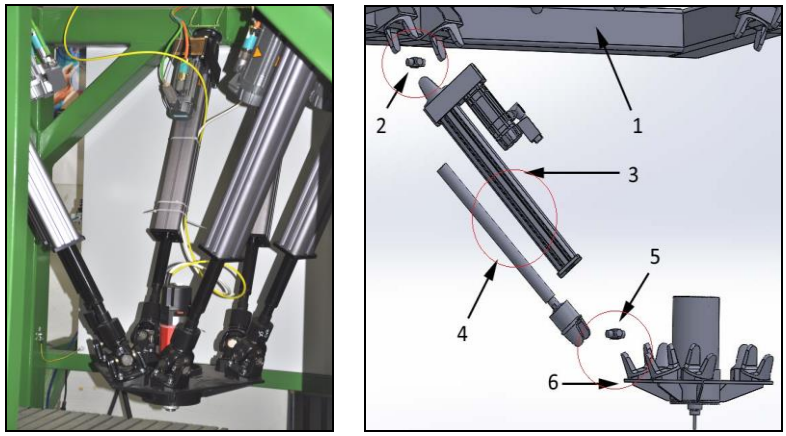

Fig. 2 Hexapod mechanism (left) and its components (right) – description in text below [1, 2]

The size of generated functional block diagram depends on the total number of free components. It means that if we want to reduce the complexity of the block diagram, it is necessary to cover the components which move together into the subassembly [8]. The result of described components' arrangement is minimized structure composed from several subassemblies without reducing the functionality and any negative impact on the simulation result. Individual parts are arranged according to the figure 2 into the several subassemblies:

1. main frame with fixed part of upper universal joints, 2. upper universal joint, 3. body of linear actuator with moving part of upper universal joint and drive unit, 4. piston of linear actuator with one half of lower universal joint, 5. lower universal joint, 6. moving platform with the spindle and second half of lower universal joints.

2.1 Export of XML File

The second step of the process is the data export to an XML file and its subsequent import into the environment of software *MATLAB/Simulink*. For this operation we need to install the free module *SimMechanics Link* for software *Solidworks*, *Autodesk Inventor* and *PTC Creo Parametric* [11]. Registration is done through the instruction **smlink_linkinv** written in MATLAB command line. After the registration in software *Solidworks* you can find in upper toolbar the *SimMechanics Link* icon. For the exporting of block diagram you can choose the option Export>>SimMechanics First Generation. After the exporting operation is automatically generated the block file **hexapod.xml** as well as the file of 3D model with suffix *.STL. Then we can import these files into the environment of software *MATLAB*. For this operation can be used command **mech_import** ('hexapod.xml') in *Simulink* window. Later will appears the 3D model of hexapod in graphic window.

3 SIMULATION MODEL OF HEXAPOD

The figure 3 shows the final model of hexapod which we created on the base of the Stewart platform demo. Model creating, gradual adjustment and parameter settings require extensive study of partial problems on free internet forums, free help available, or study materials. The block model of Hexapod is divided into three main subsystems (shown in figure 3) [1]:

- subsystem of trajectory calculation,
- PID control unit,
- and subsystem of mechanism hexapod.

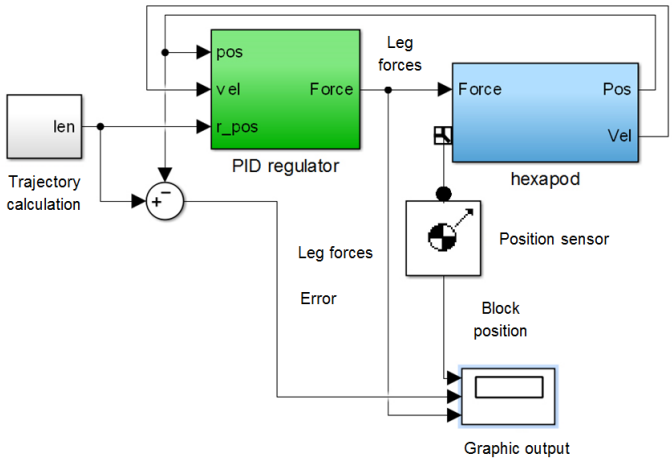

Fig. 3 Model of whole Hexapod [1]

3.1 Trajectory Calculation Subsystem

In the following text is a simple description of the individual blocks of the first subsystem – Trajectory calculation (figure 4). The block ***Constant*** generates a constant input parameter. In this case we will generate constant linear motion (later it will be replaced by another motion – e.g. circular trajectory). The recording time is set up to 0.02s. This value was chosen with respect to the speed of simulation and quality of output data as well. The blocks ***X Pos***, ***Y Pos***, ***Z Pos*** define us the linear coordinates of the TCP point (placed on moving platform with the end-effector). Their composition gives us the result motion of the platform in all three coordinates. Blocks ***X Pos*** and ***Y Pos*** can be replaced by other sources of input parameters (individual coordinates). It means that instead of block ***Constant*** should be placed the block with equation of circular motion.

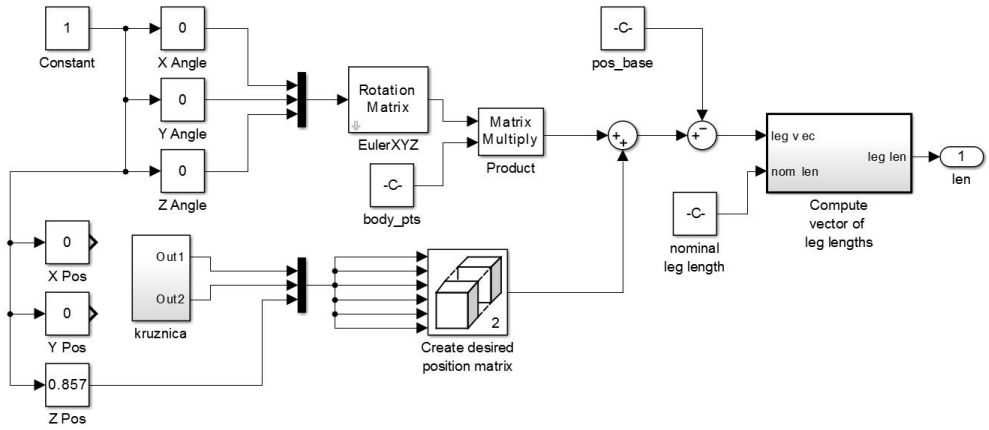

Fig. 4 Model of Trajectory calculation

The blocks ***X Angle***, ***Y Angle*** and ***Z Angle*** allow us to enter the angular settings of the moving platform in all three coordinates. Block ***EulerXYZ*** calculates the final rotation matrix (vector of size 9x1) corresponding to the orientation of the moving platform. These vectors will be transformed to the matrix with dimensions 3x3. Block ***body_pts*** represents the moving platform with 6 three-dimensional values – the joints. The block ***Product*** will produce the final transformation matrix which is composed from two input matrices ***EulerXYZ*** and ***body_pts***. Block ***Create desired position matrix*** creates the multidimensional arrays of size 6x3. Then the matrices are proceeding by two ***Sum*** blocks. Block ***pos_base*** stores coordinates of upper universal joints connected to the frame. The block ***Nominal Leg Length*** gives us the length of each leg.

3.2 PID regulator subsystem

In figure 5 the block ***pos*** refers to the sensor of actuator's current position, block ***r_pos*** stores the calculated value from the previous subsystem (trajectory calculation). Matrix ***r_pos*** will be subtracted from the matrix ***pos***. The resulting matrix is sent to the amplifier ***Kp*** (proportional feature). Into the ***Ki*** (integration feature) will be loaded the integrated value of a matrix. The block ***vel*** refers to the same sensor of the actuator, but in this case we measure the velocity. The block ***Gain*** inverts the sign of the result (from positive to negative or negative to positive). The block ***Kd*** makes the differentiate of this value. At the end are summed all blocks ***Kp***, ***Ki***, ***Kd*** and the result is a force intended for the actuator.

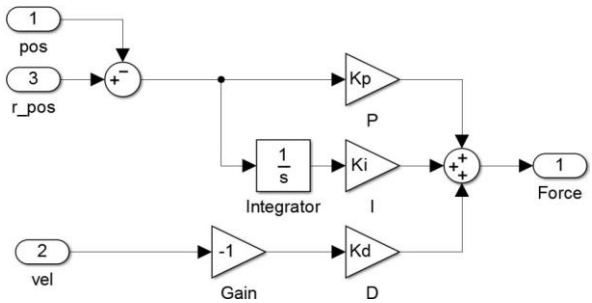

Fig. 5 Model of PID control unit

The motion error is the difference of the desired or reference length of the leg and its instantaneous or actual length [10]:

$$E_r = r_{ref} - r_{actual} = L_{traj,r}(t) - \left|\left(R \cdot \mathbf{p}_{t,r}\right) - \mathbf{p}_{b,r}\right|, \quad (1)$$

where:

E_r – motion error [m],

r_{ref} – reference length of leg [m],

r_{actual} – actual length of leg [m],

$L_{traj}(t)$ – reference trajectory length [m],

p – vector defined the distance between the centre point of base and top plate [m],

$p_{t,r}$ – vector defined the position of attachment point with respect to the centre of top plate [m],

$p_{b,r}$ – vector defined the position of attachment point with respect to the centre of base plate [m],

R – rotation matrix.

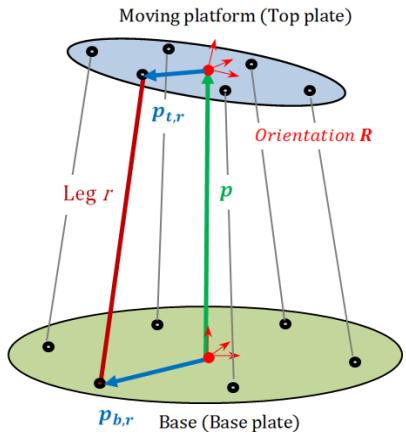

Fig. 6 Description of Hexapod kinematic structure

The reference length $L_{traj}(t)$ is given as a function of time by the output of the Leg Reference Trajectory subsystem. The vectors $\boldsymbol{p}$, $\boldsymbol{p_{t,r}}$, and $\boldsymbol{p_{b,r}}$ are defined in the figure 6. The orthogonal rotation matrix R specifies the orientation of the top plate with respect to the bottom.

The Stewart platform model use a simple PID controller and Joint Sensor blocks to measure motion. The simplest implementation of trajectory control is to apply forces to the actuator proportional to the motion error. PID feedback is a common form of linear control. A PID control law is a linear combination of a variable detected by a sensor, its time integral, and its first derivative. This Stewart platform's PID controller uses the leg position errors E_r and their integrals and velocities. The control law for each leg r has the form [10]:

$$F_{act,r} = K_p E_r + K_i \int_0^t E_r dt + K_d \left(dE_r / dt\right), \tag{2}$$

The controller applies the actuating force $F_{act,r}$ along the leg.

- If E_r is positive, the leg is too short, and $F_{act,r}$ is positive (expansive).
- If E_r is negative, the leg is too long, and $F_{act,r}$ is negative (compressive).
- If E_r is zero, the leg has exactly the desired length, and $F_{act,r}$ is zero.

The real, nonnegative K_p, K_i, and K_d are, respectively, the proportional, integral, and derivative gains that modulate the feedback sensor signals in the control law:

- The first term is proportional to the instantaneous leg position error from reference.
- The second term is proportional to the integral of the leg position error.
- The third term is proportional to the derivative of the leg position error.

The result is $F_{act,r}$, the actuator force (input) applied by the controller to the legs. The proportional, integral, and derivative terms tend to make the legs' top attachment points $\boldsymbol{p_{t,r}}$ follow the reference trajectories by suppressing the motion error.

3.3 Hexapod Subsystem

In figure 7 is shown the hexapod block diagram (the orientation of blocks is upside-down regarding to the real inverted configuration of the hexapod prototype). Lower red rectangle highlights the chain of blocks, which represente the fixed frame of hexapod (static upper platform).

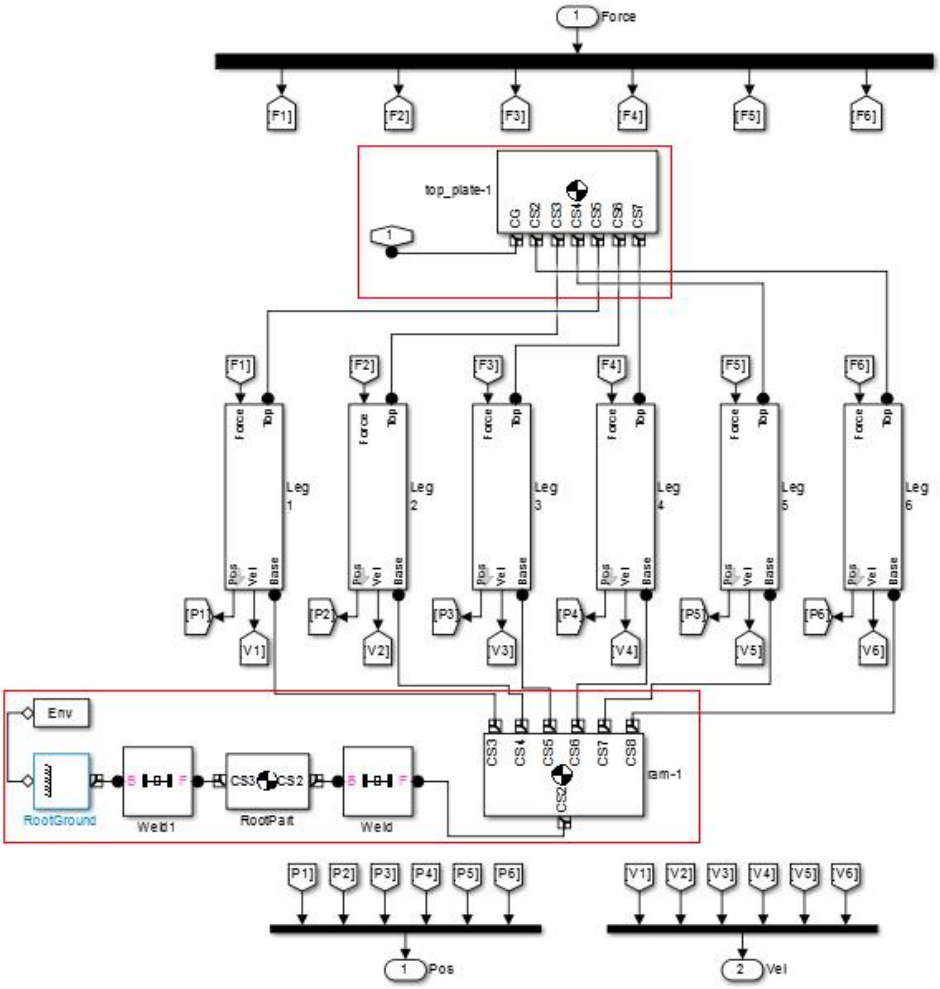

Fig. 7 Simulation model of mechanism Hexapod created in *MATLAB/Simulink*

Then the upper red rectangle describes the moving platform include the end-effector (in reality it is placed under the fixed frame). Between the fixed frame and the moving platform are located the guiding chains (extendable struts composed from linear actuators). They must be linked properly. ***Env*** block must be connected to the block ***Root Ground*** due to machine settings to the environment. It's actually a machine environment that lets you view and change the mechanical environment. Block ***Env*** has the possibility of entering the gravitational signal into a three-dimensional environment, as we set. When you click on a block ***top_plate-1*** a window of basic settings of moving platform appears. The model of each leg (more detailed description of blocks Leg1-Leg6) is shown in figure 8.

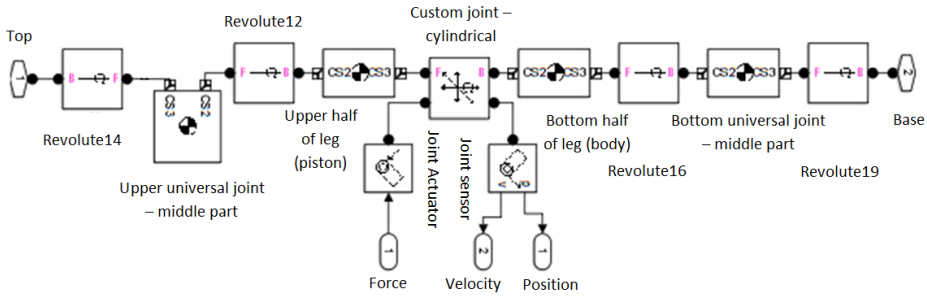

Fig. 8 Model of one leg created in *MATLAB/Simulink*

3 FIRST SIMULATIOUN OUTPUTS

Constructed model can be consequently used to carry out vide variety of simulations, different motion analysis, simulation of different regulation methods for drive units, influence of controller settings, or control stability during idle or under load. The results from simulation depend on the type of input parameters as well as on the system settings. According to our goal it is necessary to prepare suitable input information like for example mathematical description of the trajectory, loading forces, the impact of disturbance variables, etc.

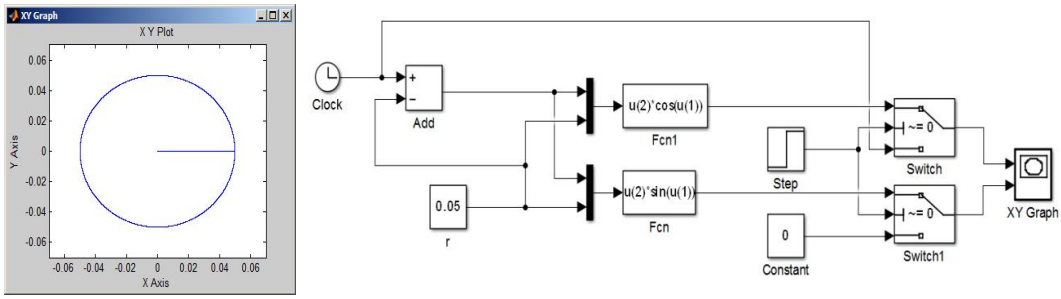

Fig. 9 Circular trajectory for first testing cycle

There we have the ability to add various input blocks generating the trajectory of various shapes, such as Sine Wave Generator Pulse, Ramp, Random Number, Step and others. We chose a circular trajectory as input information for the first functional tests of the mechanism. Then was composed a system of blocks generating a circular trajectory with a radius of 5 cm (the value of X = 0.05), along which is moving the tool end point with moving platform and the spindle. From the centre, the tool will move along a line until it touches the outer circumference and subsequently complete the circle (fig. 9 and 10). Whole system is covered by block ***XY Graph*** [1]. Later we can simulate the reaction of the mechanism on the material resistance during machining process.

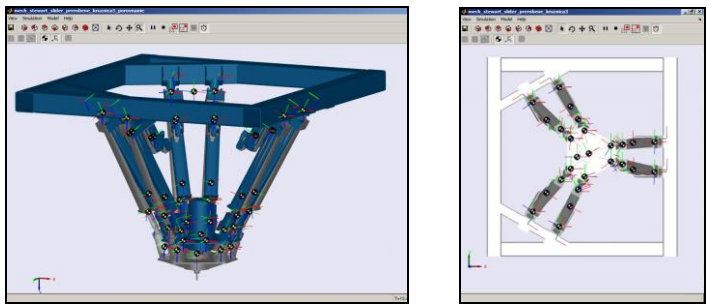

Fig. 10 Simulation results of hexapod motion during testing of circular interpolation

4 CONCLUSIONS

Hexapod can be considered as a general parallel mechanism with complex DOFs because it produces 6 DOF of a general rigid body motion. Machine tools based on parallel or hybrid mechanisms can be successfully applied for high speed cutting, etc. [3, 5, 6]. At the authors' workplace was designed the prototype of machine tool with hexapod kinematic structure. In the article is described the modelling process of hexapod mechanism in *MATLAB/Simulink*. The created model is composed from individual functional blocks mutually interconnected by links. The model describes the kinematics as well as the characteristics of its individual parts. The simulation model can be consequently used to carry out vide variety of simulations, different motion analysis, simulation of different methods for drive units regulation, controller settings with respect to minimizing the output positioning error, or simulation of control system stability during idle or under load. The model will be in next period improved to better describe the characteristics of the real mechanism.

ACKNOWLEDGEMENT

This article was created by the solution of project - code ITMS 26220220046: "The Development of Parallel Kinematic Structure Prototypes for Application in the Area of Machine Tools and Robots" supported by operational program Development and research, financed from European foundation for regional progress.

REFERENCES

[1] DRESTO, D.: *Use of Simulink for drive control of machine with parallel kinematic structure*. Diploma thesis. Žilina, 2015.

[2] POPPEOVÁ, V. et al.: Parallel Mechanism and its Application in Design of Machine Tool with Numerical Control. In *Applied mechanics and materials*, 2013, Vol. 282, p. 74-79, ISSN 1660-9336

[3] D. ZHANG: Parallel Robotic Machine Tools, Faculty of Engineering and Applied Science, University of Ontario Institute of Technology,Canada,LLC 2010, ISBN 978-1-4419-1116-2

[4] BŘEZINA, T., FLORIAN, Z., HOUŠKA, P., et al.: 2006, Device for Experimental Modeling of Biomechanical Systems Properties. Simulation Modelling o Mechatronic Systems II, 2004, ISBN 80-214-3341-8, p. 225-252, pp. 276

[5] ZELENÝ, J.: Numerically Controlled Machine Tools and Accesories. ČVUT Praha, 1999.

[6] MERLET, J.-P. 2006. Parallel Robots. Dordrecht: Springer Academic Publisher, 2006, ISBN-10 1-4020-4133-0

[7] LUNG–WEN TSAI: Robot Analysis: The Mechanics of Serial and Parallel Manipulators. Interscience 1999, ISBN 04-7132-593-7, s. 520

[8] KUMIČÁKOVÁ, D., RENGEVIČ, A.: New possibilities of robot arm motion simulation. In: *Communications : scientific letters of the University of Žilina*. - ISSN 1335-4205. - Vol. 18, no. 1A (2016), s. 81-86.

[9] KURIC, I. et al.: Experimental device for practicing routines of machine tool precision measurement. In: *Academic journal of manufacturing engineering.* - ISSN 1583-7904. - Vol. 13, no. 1 (2015), s. 39-44.

[10] SimMechanics in Academia. 2015. *MathWorks* [online]. [cit. 2015-05-13]. Available: http://www.mathworks.com/products/simmechanics/features.html#modeling-multibody-system

[11] GREPL, R. Modelování mechatronických systémů v Matlab/SimMechanics. BEN - technická literatura, 2007, p. 152, ISBN: 8073002268