Modelling and Simulation of Machine Tool Prototype with 6DOF Parallel Mechanism in Matlab / Simulink

Vladimír Bulej[1,*], *Juraj* Uríček[1] , *Manfred* Eberth[2], *Ivan* Kuric[1], and *Ján* Stanček[1]

[1]University of Zilina, Faculty of Mechanical Engineering, Department of Automation and Production Systems, Univerzitna 1, 010 26 Zilina, Slovakia
[2]AUDI AG, I/EE-554, D-85045 Ingolstadt Germany

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.

Keywords: parallel mechanism, machine tool prototype, hexapod

1 Introduction

Recent progress of robotics technology and CNC machine tools brings a lot of breakthroughs in the industries as well as in the fields of amusement, welfare and medicine.

Parallel mechanisms actuated in parallel by several serial kinematic chains can offer higher accuracy and stiffness, very high motion speed, and heavier payloads than their serial opponents [1-5]. Nowadays, the conventional parallel mechanism with six degrees of freedom (DOFs) of three translations and three rotations (structure 3T3R), also known as the Stewart platform, can be found in wide variety of applications involving motion simulators [6, 7], parallel machine tools [8-9], force-torque sensors [10-12], heavy-duty free forming production machine [13], testing stands, antennas, micro-positioning units [14-15], calibration devices [16], etc.

General 6 DOFs parallel mechanism has its own drawbacks, such as high coupled motions, small workspace and singularities inside the workspace. The moving platform of general fully parallel mechanism can tilt up to approximately 40 degrees from the horizontal plane at the origin position. It is the result of joint range limits and interferences between limbs or limbs and the moving platform. All these characteristic features have negative influence on its application in certain areas. For example, the small orientation workspace limits its application for motion simulation of acrobatic aircrafts and similar

---

* Corresponding author: vladimir.bulej@fstroj.uniza.sk
Reviewers: *Maciej* Berdychowski*, Vladimír* Dekýš

kind of devices required higher tilt angle. However, for most commercial flight or car simulators the capability of the common Stewart platform is sufficient.

To reach high tilting angles, several authors proposed some concepts [12], for example Eclipse mechanism for five-face machining in a single setup. On the other hand, not all applications need to reach so high mobility. At present, such tasks can be automated by use of the Hexapod [5, 17] that provides relatively large workspace volume but limited orientation capability around three translational axes.

It is still necessary to do the research related to the topic of designing of different kind of parallel mechanisms with up to 6DOF, as well as their type synthesis, kinematic structure analysis, workspace analysis, etc. As the main motivation to do all these analyses can be considered the necessity to understand their properties and changing of their behaviour along the whole workspace more detailed. The literature repeatedly refers to the simulation and modelling of parallel mechanisms. However through the modelling and simulation in appropriate simulation software tools we can reach these objectives.

During last fifteen years, there was formed a group of researchers dealing with the parallel kinematic structures development as well as at the authors workplace [3, 4, 18]. 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. This article deals with the modelling and simulation of hexapod mechanism in software MATLAB / Simulink.

2 Structure and design of machine tool prototype

We decided to use the machine tool prototype (Figure 1) based on mechanism hexapod developed at the authors' workplace for modelling and simulation in Matlab/Simulink software. This experimental machine tool is equipped with automatic tool changing and automatic part changing system as well. This article will later focus only on model creation of the hexapod mechanism in software Matlab/Simulink.
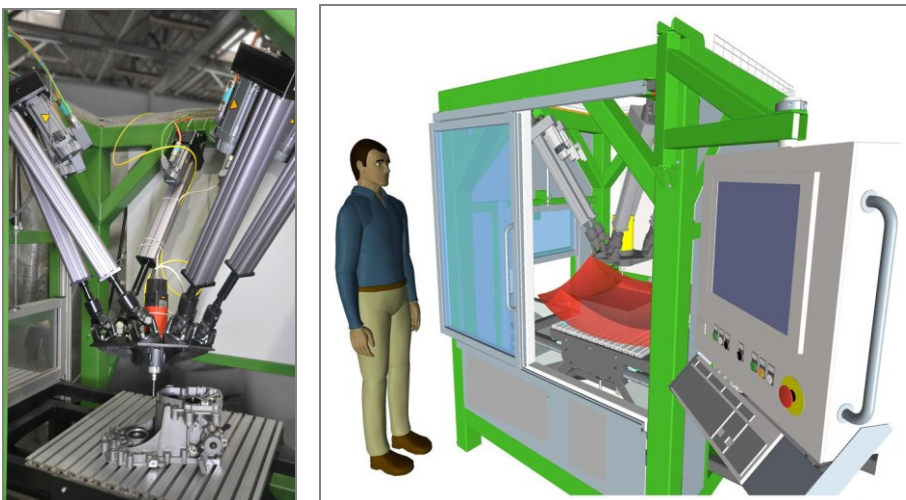
Fig. 1. Machine tool prototype with hexapod mechanism designed at the authors workplace

Designed machine tool prototype can be divided into these basic subsystems:
- main frame,
- mechanism hexapod with 6 driven electro-mechanical legs and moving platform with main milling spindle,

- system of automatic tool changing and system of automatic part changing,
- electrical switchbox with control and power units.

The machine tool prototype is based on a moving mechanism with parallel kinematic structure − hexapod. Main frame was designed as a welded structure made of steel thin-walled square tubes. The moving platform is capable of moving within its workspace and is connected to the fixed platform via six extensible legs equipped at both ends with universal joints (2 DOFs) or spherical joints (3 DOFs) respectively.

Table 1. Basic properties of actuator Exlar IX30 [6] and motor Siemens 1FT7 [5]

Linear actuator Exlar IX30-1602							
Parameter	Continuous force (N)	Stroke (mm)	Speed at Max. Rated RPM ($mm \cdot s^{-1}$)	Life (mm)	Screw lead (mm)	Max Rated Input RPM (min^{-1})	Backlash (mm)
Value	4026	406.4	338	136.5×10^6	5.08	4000	preloaded
Motor Siemens 1FT7-034							
Parameter	Nominal power (kW)	Nominal output torque (Nm)	Nominal revolutions (min^{-1})	Accuracy		Increment number pre revolution (-)	Weight (kg)
Value	1	2	6000	$\pm 40''$		2048	4

The body of each leg is created by linear actuator Exlar IX30 which can generate continuous force up to 4026 N by maximum linear velocity 338 mm per second (for more parameters see Table 1). These actuators present a new range of alternatives for linear motion solutions. Compared to the low performance spur gears provided by most ball screw actuators, the I Series' planetary gears offer an extended life, high input speed and output torque and quiet operation. They are driven by the motors Siemens 1FT7.

3 Matlab/Simulink simulation model of mechanism Hexapod

Next chapter will be focus on development of simulation model of mechanism hexapod from our machine tool prototype suitable for carrying out different further analyses. The final model will be developed in software Matlab/Simulink. The module *Multibody* or older version *SimMechanics* from MathWorks Inc. is a recently developed Matlab library based on Simulink. It is modeling software for mechanical systems that provides a single simulation environment for the construction of reliable mechanical and controller models. These models can be reused by converting them into compact, efficient C code for embedded controller implementations and hardware-in-the-loop testing applications [2, 11].

The process of simulation model making in Matlab/Simulink can be divided into several separate steps [2, 5] where will be prepared:

- model of one leg and model of entire hexapod mechanism,
- model of control system a PID controller,
- interface for trajectory generation and input of parameters.

3.1 Model of one leg and model of entire hexapod mechanism

At the beginning it was necessary to prepare an appropriate 3D model of mechanism Hexapod (Figure 2) 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. The CAD-to-SimMechanics translator can help us to convert the mechanical design into a SimMechanics model faster and easier. For this operation we need to install the free module *SimMechanics Link* for software Solidworks, Autodesk Inventor or Creo Parametric [18]. After exporting operation it is automatically generated the block file "hexapod.xml" as well as the file of 3D model with suffix "STL". Both files can be imported directly to Matlab. However, the designer must make sure that the SimMechanics model corresponds exactly to the real mechanism.
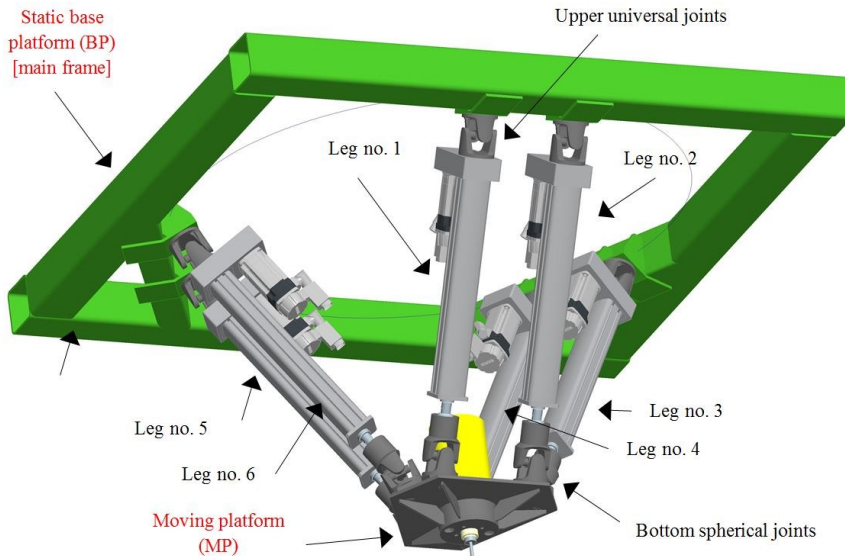
**Fig. 2.** CAD model of mechanism hexapod

Consequently the specification of actuators that are used to achieve the desired motion of the different joints of the mechanism can be set. Sensors are then used to monitor the parameters important for simulation purposes. In our case are important mainly position and velocity monitoring of each actuator as well as the end-effector (point TCP, or the tool tip). The organising of assembly components into the individual subassemblies is needed to minimize the number of functional blocks in final Matlab/Simulink model.

3.1.1 Model of one leg

As it was mentioned above, the entire hexapod mechanism is composed from 6 identical linear actuators – extensional legs. So before of modelling hexapod we can prepare the appropriate model of one leg which will be multiplied six times. The model of each leg including more detailed description of blocks Leg1 up to Leg6 is shown in figure 3. To be more illustrative, we add there also the part of static base platform (BP) as well as moving platform (MP).

The leg or guiding chain is composed from upper universal joint (UJ) with 2 DOFs, body of linear actuator itself containing drive unit and linear transmission unit with 1 DOF and bottom spherical joint (SJ) with 3 DOFs (composed from universal joint and additional revolute joint) connected to the moving platform. The same structure is represented by block schema created in Matlab/Simulink (Figure 3, right side). For connection with external objects outside the leg are used Connection blocks on both sides. The control of guiding chain motion is provided by INPUT block called "Acting intervention" where is

coming the data from control system. Contrary, for monitoring of leg behaviour are used OUTPUT blocks "Position" and "Velocity".
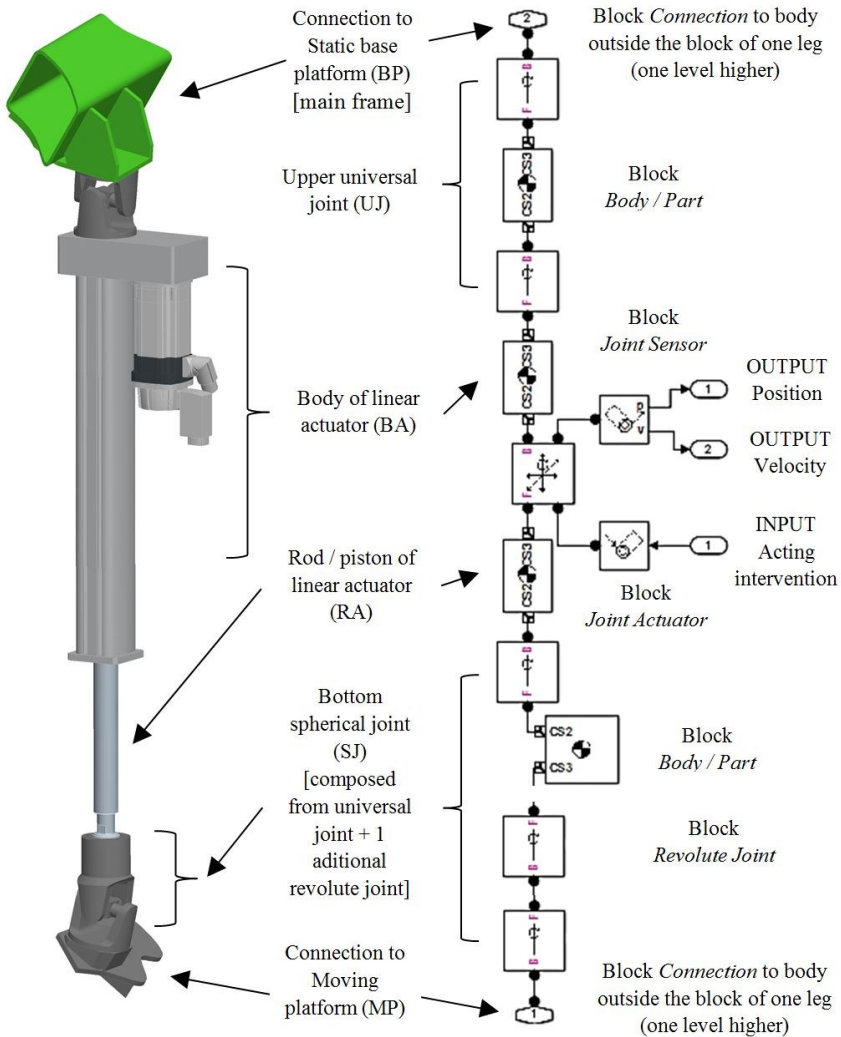

Fig. 3. Making of model for one leg in Matlab/Simulink

3.1.2 Model of hexapod mechanism

By multiplication of one leg block model six times, we can get the final model of hexapod mechanism (see Figure 4). We would like to highlight that the orientation of block schema is upside-down regarding to the real inverted configuration of the hexapod prototype (moving platform is in top). The model of one leg is "covered" under each block labelled as Leg no. 1 to 6, and it should be opened double-clicking over it. For data exchange between other levels are used the above mentioned INPUT and OUTPUT blocks. In Figure 4 is shown the hexapod block diagram. Lower red rectangle highlights the chain of blocks, which represents the fixed frame of hexapod (static upper platform).

The upper red rectangle describes the moving platform including the end-effector. Between the fixed frame and the moving platform are located six guiding chains (extendable struts composed from linear actuators). They must be linked properly to other objects. **Env** block must be connected to the block **Root Ground** due to machine settings. It is actually a machine environment that lets you view and change the mechanical environment. There can be set the gravity into the three-dimensional environment. When you click on a block **MP** a window of basic settings of moving platform will appear.
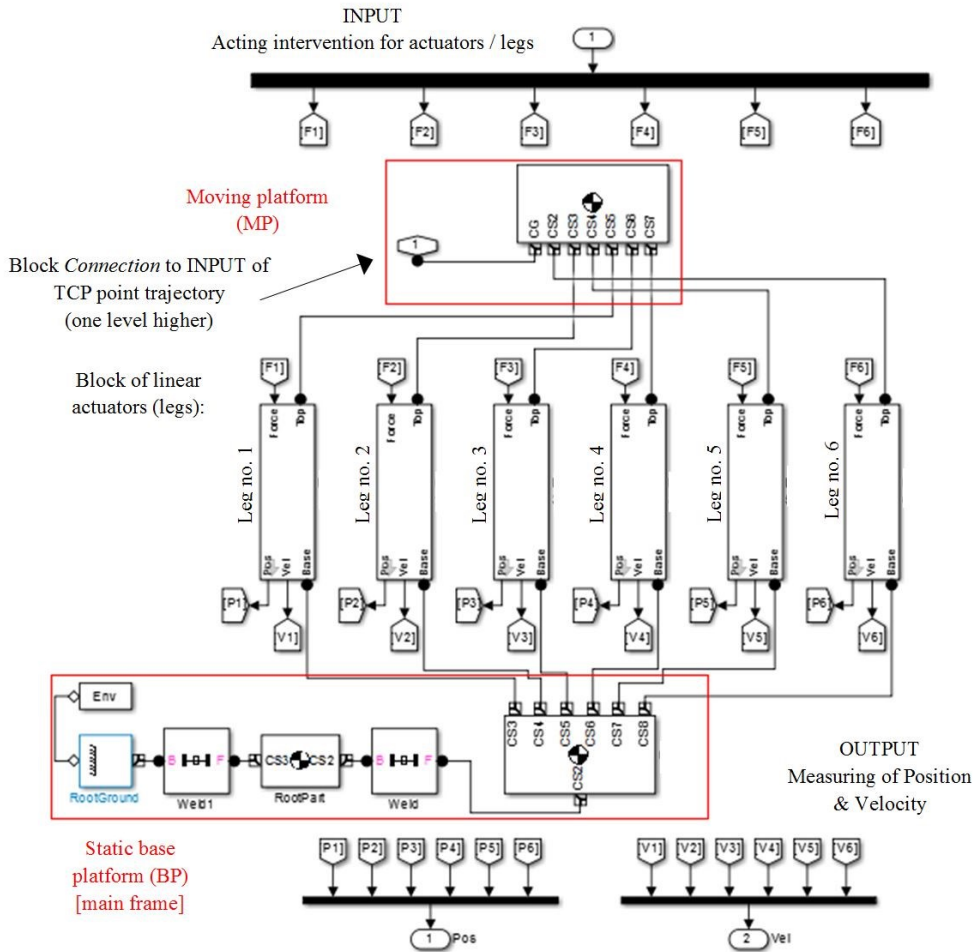

Fig. 4. Matlab/Simulink model of whole Hexapod

3.2 Model of control system, interface and PID controller

To complete the functional simulation model of mechanism hexapod we need to prepare not only the model of the mechanism itself, but also the model of control system, PID regulator and communication interface.

### 3.2.1 Model of hexapod with control system and I/O features

The block model of whole hexapod control system was designed as the schema composed from three particular blocks – block of hexapod mechanism, block for trajectory input and calculation and block of PID regulator. Besides these components there are used some peripheral blocks, as for example Graphic output for output parameter monitoring and graph plotting, position sensor and connections. Block model (Figure 5) represents also the interface for control of whole Simulink model of hexapod. According to this block model we can see, that the behaviour of entire hexapod mechanism is controlled by PID regulator along the trajectory generated by module Trajectory calculation. As the main control parameter is used Acting intervention signal (ActingIntv) continuously send from PID regulator to hexapod block.
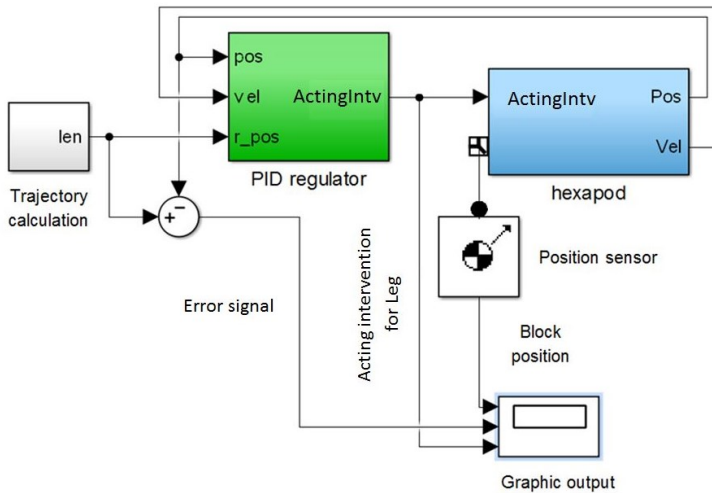
Fig. 5. Matlab/Simulink model of interface and control system

3.2.2 Model of PID controller

A proportional–integral–derivative controller (PID controller) is a control loop feedback mechanism A PID controller continuously calculates an error value as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively). In our case its function is to control the motion along the desired trajectory with desired velocity to be stable with minimum position and velocity error or deviations.
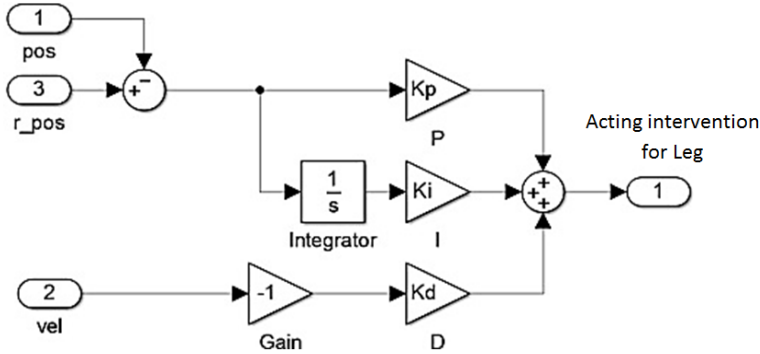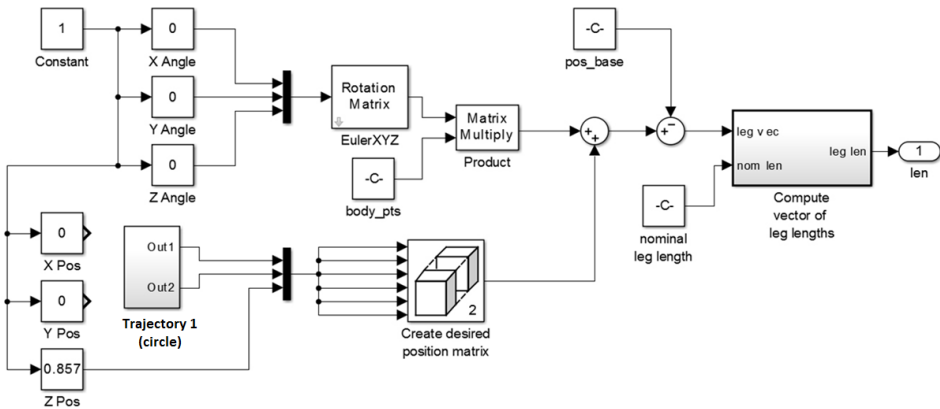
Fig. 6. Matlab/Simulink model of designed PID controller for mechanism stable control

In Figure 6 the block **pos** refers to the actuator's current position while block **r_pos** stores the calculated values from the block Trajectory calculation. Matrix **r_pos** is calculated from the matrix **pos**. The calculation result is amplified by proportional **Kp**, integral **Ki** and derivative influence **Kd**. The block **vel** refers to the same sensor on the actuator, but in this case we measure the velocity. The block **Gain** inverts the sign of the result (from positive to negative and vice versa). At the end all blocks **Kp**, **Ki**, **Kd** are summed and the result is final acting intervention intended for the actuator.

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 acting intervention to the actuator proportional to the motion error. PID feedback is a common form of linear control.

3.3 Model of interface for trajectory input and calculation

The last element of designed model of control system for hexapod mechanism is the interface for trajectory input and calculation (Figure 7) labelled also as Trajectory calculation. Thanks to this model we can easily generate the shape and parameters of desired trajectory which we want to navigate hexapod along.

Fig. 7. Matlab/Simulink model of designed interface for trajectory input and calculation

The block **Constant** generates a constant input parameter representing constant linear motion. The recording time is set up to 0.02 s with respect to the speed of simulation and quality of output data as well. The blocks **X Pos**, **Y Pos**, **Z Pos** define three positioning

coordinates of the TCP point (placed on moving platform with the end-effector) while the blocks *X Angle*, *Y Angle* and *Z Angle* allow us to control the angular motion of the moving platform in all three coordinates. As an example are in Figure 7 the blocks *X Pos* and *Y Pos* are replaced by other sources of input parameters, concretely by the block with equation of circular motion. Block *EulerXYZ* calculates the final rotation matrix corresponding to the orientation of the moving platform. 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*. The block *Nominal Leg Length* calculates the length of each leg.

4 Simulation results and verification of designed model

The functionality of designed Matlab/Simulink model of mechanism hexapod was verified on initial simulations. We have chosen the circular trajectory with a radius of 5 cm along which was moving the tool end point with moving platform and the spindle for the first functional tests. Simulation results are shown in Figures 8 and 9.
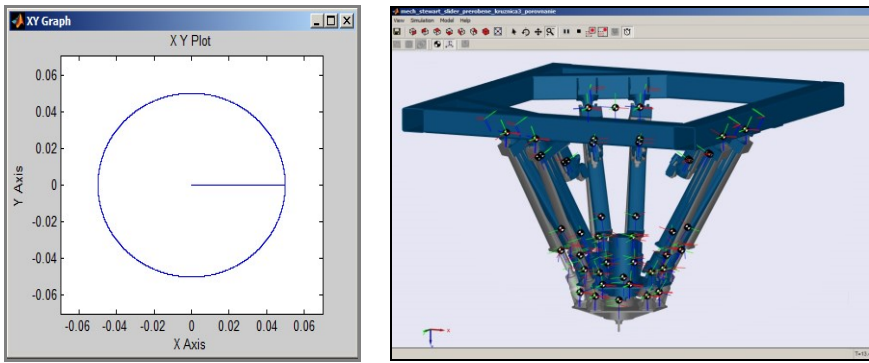

Fig. 8. Preliminary tests for verification of model functionality (motion along circular trajectory)
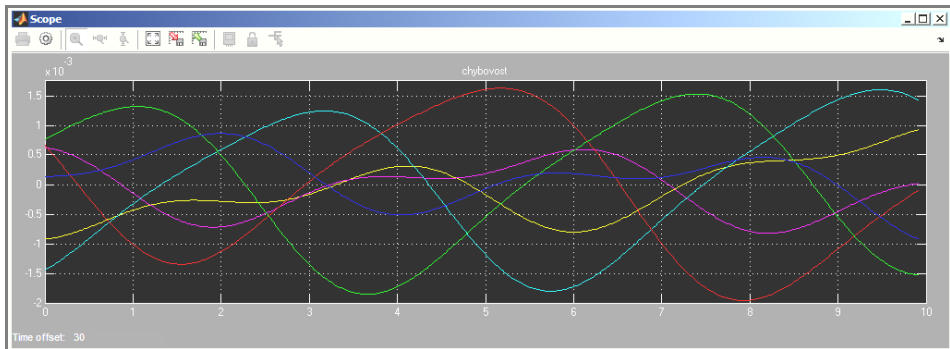

Fig. 9. Results of preliminary tests – positioning inaccuracy under the external error signal

Consequently the model can be 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. After these preliminary tests we plan to verify the shape and dimensions of generated workspace with its 3D model designed by other simulation software as well as to simulate the distribution of hexapod mechanism properties along such workspace.

5 Conclusions

At the authors' workplace there was designed the prototype of machine tool with hexapod kinematic structure during the last couple of years. Hexapod can be considered as a general parallel mechanism with complex mobility because it produces motion in all 6 DOFs.

The main aim of this article was the design of simulation model of mechanism hexapod in software Matlab/Simulink. The designed model is composed of individual function blocks arranged to the block schema and mutually connected by suitable links. Besides the kinematic structure of entire mechanism the model contains as well as the control system, PID controller and interface for trajectory data input. The functionality of designed model was verified on preliminary tests with motion along circular trajectory.

The simulation model can be consequently used to carry out vide variety of simulations, different motion analysis, simulation of different control methods for drive units, controller settings with respect to minimizing the output positioning error, or simulation of control system stability during idle or under load. For the future work we plan to continuously improve the designed model and extended it by more detailed model of electromechanical linear actuator to get more precise results closer to the real mechanism.

References

1. J.P.Merlet, *Parallel Robots* (Dordrecht: Springer, 2006)

2. R. Grepl, *Modelování mechatronických systémů v Matlab/SimMechanics* (BEN, 2007)

3. SimMechanics in Academia. 2015. MathWorks [online].

4. V. Poppeova, et al. Appl. Mech. a. Mat, **282**, p. 74-79 (2013)

5. V. Bulej, et al, App. Mech. a. Mat. **436**, p 366-373 (2013), DOI: 10.4028/www.scientific.net/AMM.436.366

6. A.Sapietova, M. Saga, I. Kuric, S. Vaclav, Int. J. of Adv. Rob. Sys. **15**, 1 (2018), doi: 10.1177/1729881417754152

7. I. Kuric, V. Bulej, M. Saga, et al., Int. J. of Adv. Rob. Sys. **14**, 6 (2017), DOI: 10.1177/1729881417743029

8. M. Cisar, N. Cubonova, I. Kuric. MATEC-W. of Conf., **94**, (2017), DOI: 10.1051/matecconf/20179401003

9. N. Cubonova, M. Cisar, Adv. in Sc. and Technology-Res. J., **9**, 26, p. 34-40 (2015)

10. P. Bozek, P. Pokorny, et al, Int. J. of Adv. Rob. Sys. **13**, (2016), DOI: 10.1177/1729881416663665

11. M. Kelemen, T. Kelemenova, I. Virgala, Modelling of Mech. and Mechat. S. **96**, p. 215-220 (2014), DOI: 10.1016/j.proeng.2014.12.146

12. V. Bulej, J. Stanček, et al. MATEC-W. of Conf., **94**, (2017), DOI: 10.1051/matecconf/201815702006

13. J. Stancek, V. Bulej, Acad. J. of Man. Eng., **13**, 4, p 38-43 (2015)

14. L. Jakubovicova, et al, Proced. Eng. **177**, p 520-525 (2017), DOI: 10.1016/j.proeng.2017.02.255

15. M. Handrik, M. Vasko, et. al, Proced. Eng. **136**, p 359-364 (2016), DOI: 10.1016/j.proeng.2016.01.223

16. A. Rudawska, et al. J. of Adh. Sci. and Techn., **10**, 30 p. 182-192 (2016) DOI: 10.12913/22998624/62513

17. J. Augste, M. Holub, R. Knoflicek, et al. Mechatronics 2013: Rec. T. and Sc. Adv. p 1-7 (2014), DOI: 10.1007/978-3-319-02294-9_1

18. D. Dresto, *Use of Simulink for drive control of machine with parallel kinematic structure* (Diploma thesis, Zilina, 2015)