

^{1.} Slobodan STEFANOVIĆ, ^{2.} Danilo MIKIC, ^{3.} Dragan GOLUBOVIC, ^{4.} Dragoslav ILIC

MODELING AND SIMULATION OF ROBOT KR 80 SERIES BASED ON MATLAB SOFTWARE PROGRAM

^{1.} GRADUATE SCHOOL OF APPLIED PROFESSIONAL STUDIES, VRANJE, SERBIA

^{2.} TECHNICAL SCHOOL, UPPER MILANOVAC, SERBIA

^{3.} TECHNICAL FACULTY, CACAK, SERBIA

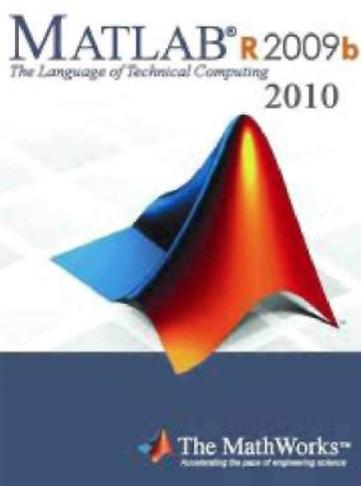
^{4.} JKP „WATER SYSTEM“, ZAJECAR, SERBIA

ABSTRACT: Robotics is a multidisciplinary branch of science that integrates knowledge from many areas of mechanics, electronics, computer science and automation. Due to the large application in practice often goes in some parts of medicine, art, economics. Development of computers and appropriate software to create and simulate the robots have led to substantial savings in the cost of such a project. This includes the development of the whole project idea through finished product, and the use of robots in manufacturing section. Further in the presentation we will experimentally demonstrate the application of MATLAB 2010 to create industrial robot KR 80 and movement simulations.

KEYWORDS: MatLab software, industrial manipulators, Robots

INTRODUCTION - ANALYSIS AND SIMULATION USING MATLAB SOFTWARE PROGRAM

Program MatLab is using for solving various mathematical problems, a series of calculations and simulations related to signal processing, control, regulation and system identification. The first version of MatLab, a simple laboratory matrix (Matrix Laboratory), was written in the end of 1970 at the University of New Mexico and Stanford University with the aim of implementation in matrix theory, linear algebra and numerical analysis. Fortran was used and parts libraries UNPACK and EISPACK. At the beginning of the 80's switched to the C programming language to add new features, primarily in the areas of signal processing and automatic control. Since 1984 MatLab is available as a commercial product from MathWorks. Today MatLab properties are far beyond the original "matrix laboratory". It is an interactive system and programming language for general scientific and technical calculations. In addition to the basic system, there are many software packages that extend him and cover almost all areas of engineering activities: signal and image processing, 2D and 3D graphics, automatic control, system identification, statistical analysis, analysis in time and frequency domain, symbolic math and other. One of the most important package is SIMULINK -visual tool that enables the simulation of continuous and discrete systems using function block diagrams without request from the user detailed knowledge of the syntax of a programming language.



MatLab is also designed as a system in which users can easily build their own tools and libraries, and modify existing ones. For this purpose, the simple programming language is using. It is also possible to use C, Fortran, Ada or Java.

Simulink is part of the MatLab designed for simulation of dynamic systems. To enter a description of a system which simulate is used a simple graphical environment in which we compile/draw model combining ready-made components. Such an approach is a simulation of the system significantly easier because the user does not require the input of differential equations describing the system with the knowledge of MatLab programming language, but it is sufficient to know the block diagram of the system.

The next section of this paper work I will try to analyze the robot model using MatLab respectively its parts: SimMechanics and Robotics toolbox. SimMechanics is standard part of the MatLab, applicable to all areas of mechanics, while the Robotics toolbox specialized part of software for simulation manipulated robots.

DETERMING PARAMETERS OF INDUSTRIAL MANIPULATORS

Our robot model that we will analyze in the following section is based on the famous industrial robot KR 180 series manufactured by KUKA.

Robots series KR180 are robots the last generation company KUKA Robotics, designed to work in the industry. KR180 robot is based on a simple and reliable design that enables excellent motion control with high productivity. KR180 robot (Figure 1) has four degrees of freedom that is four rotational cranks that run by electric servo-motors. The robot is designed for high productivity packaging and manipulating with finished products. There is also a version with two rotational cranks, but we will this time keep up on the complex model.



Figure 1. Realistic manipulators image

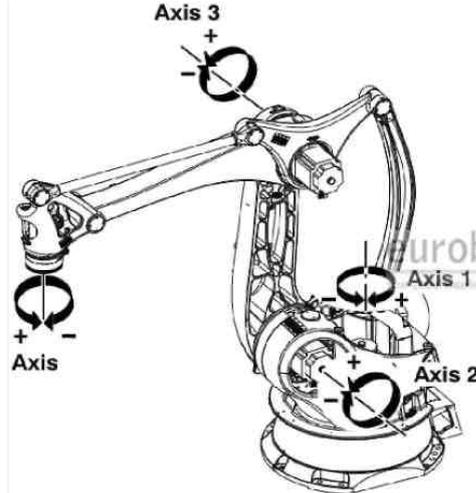


Figure 2. M-420iA robot with cranks (links)

The first step in modeling robot KR is to determining D-H parameters. DH parameters table is a notation developed by Denavit and Hartenberg, is intended for the allocation of orthogonal coordinates for a pair of adjacent links in an open kinematic system. It is used in robotics, where a robot can be modeled as a number of related solids (segments) where the D-H parameters are used to define the relationship between the two adjacent segments. The first step in determining the D-H parameters is in locating crank and determine the type of movement (rotation or translation) for each crank.

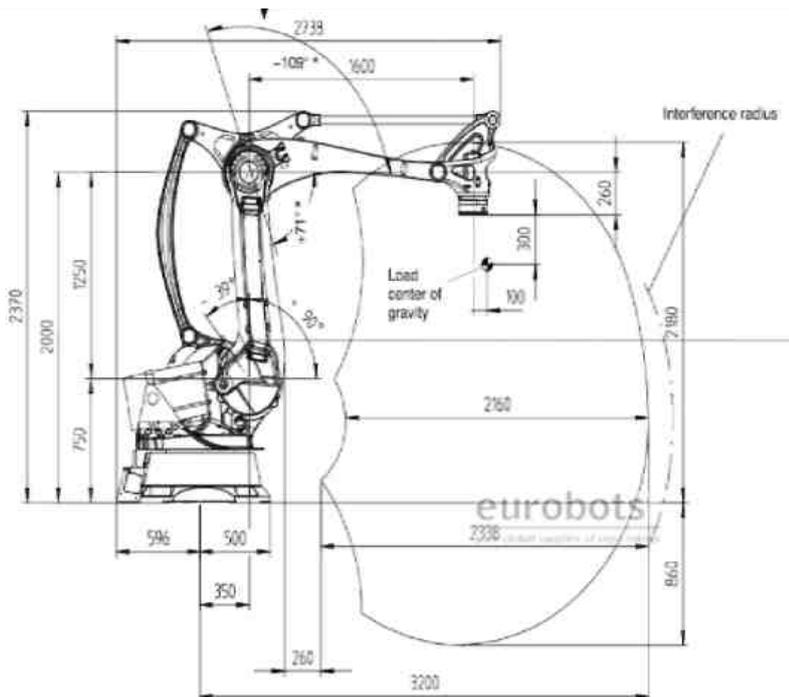


Figure 4. KR180 robot with basic dimensions

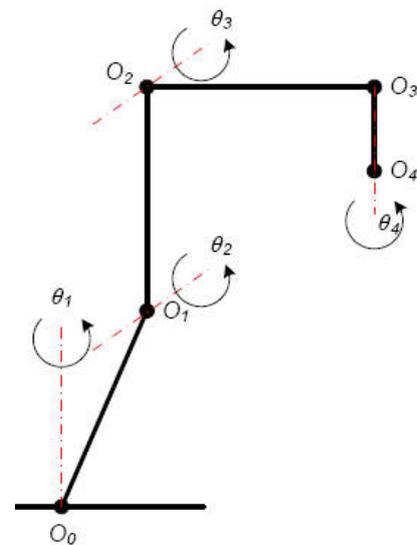


Figure 3. A simplified scheme of the robot KR180

As can be seen (Figure 2.), the robot KR180 series has four rotational cranks, where each crank has allowed angle of rotation $\theta_1, \theta_2, \theta_3$ and θ_4 . Crank, axes and rotation angles are shown in a simplified diagram in Figure 3.

Using D-H parameters defined in the previous steps in table 1., robot model was created in MatLab software using the Robotic Toolbox. Robot model in addition to previously determined D-H parameters contains physical

parameters which is using in the calculation of the dynamics movement. List of parameters used to describe the robot in MatLab, using Robotic Toolbox, is shown in table 2. Link length a_i and the segments distance D_i we will change compared to the factory model.

Table 1. D-H parameters

| Axis number | Rotation angle in relation to the previous crank a_i | Link length a_i | Distance between segments d_i | Rotaton angle θ_i |
|-------------|--|-------------------|---------------------------------|--------------------------|
| 1(0-1) | $n/2$ | 0,35 m | 0,75m | variable 61 |
| 2(1-2) | 0 | 0 mm | 1,25m | variable 62 |
| 3(2-3) | $n/2$ | 1,6 m | 0 mm | variable 63 |
| 4(3-4) | 0 | 0 mm | 2,6m | variable 64 |
| Crank | Rotation angle scope | Crank speed | | |
| J1 | $-185^\circ / +185^\circ$ | $105^\circ / S$ | | |
| J2 | $-39^\circ / 90^\circ$ | $105^\circ / S$ | | |
| J3 | $-71^\circ / 109^\circ$ | $95^\circ / S$ | | |
| J4 | $-270^\circ / +270^\circ$ | $300^\circ / S$ | | |

Table 2. The parameters that describe the robot in MatLab

| Ordinal | Simbol | Description |
|---------|---|--|
| 1. | a | Rotation angle in relation to the previous crank |
| 2. | a | Link length between segments |
| 3. | θ | Rotaton angle |
| 4. | d | Distance between cranks |
| 5. | a | Crank type - 0 rotational, others translational |
| 6. | m | Crank mass |
| 7. | $r_x r_y r_z$ | Position of the center of segment mass relative to its coordinate system |
| 8. | $I_{xx} I_{yy} I_{zz} I_{xy} I_{xz} I_{yz}$ | Tensor segments inertia elements to its center mass |
| 9. | J_m | Armature inertia |
| 10. | G | Transmission ratio (crank/segment speed) |
| 11. | B | Viscous friction |
| 12. | $T_{c+} T_{c-}$ | Coulomb friction |

CREATING SCRIPTS AND MODEL IN MatLab

Creating a model of robot construction $kr180$ means the introduction of its parameters according to the modified Denavit-Hartenberg notation in $.m$ data MatLab program scripts. The file path is as follows: $C:\Program Files\MATLAB\R2010b\toolbox\robot\demos\kr180.m$

Since the file is not existing, begin creating these files in the folder with the activated pathway in the working environment (workspace) of program. For creating a $.m$ files it is used the command from the File menu.

After launching the command File - New> Script it is opening the program editor and access to the input of the programming language MatLab, which will be a model of robot construction $kr180$.

File $Kr180.m$ is stored in the program editor window script command File - Save As ... In part of this command performs the following selection and folders:

$C:\Program Files\MATLAB\R2010b\toolbox\robot\demos$ that in the previous step was added to the active path in Matlab, which are already a DEMO example of industrial robots.

$Kr180.m$ script contains the following parameter:

```
%Robot Kuka KR180
```

```
% Definition Links link ([alpha a theta d sigma])
```

```
clear L
```

```
L{1} = link([ pi/2 0.35 0 0.75 0], 'standard');
```

```
L{2} = link([ 0 0 0 1.25 0], 'standard');
```

```
L{3} = link([ pi/2 1.6 0 0 0], 'standard');
```

```
L{4} = link([ 0 0 0 2.6 0], 'standard');
```

```
% Mass of segments of the robot L{1}.m
```

```
L{2}.m =150; L{3}.m =100; L{4}.m =50;
```

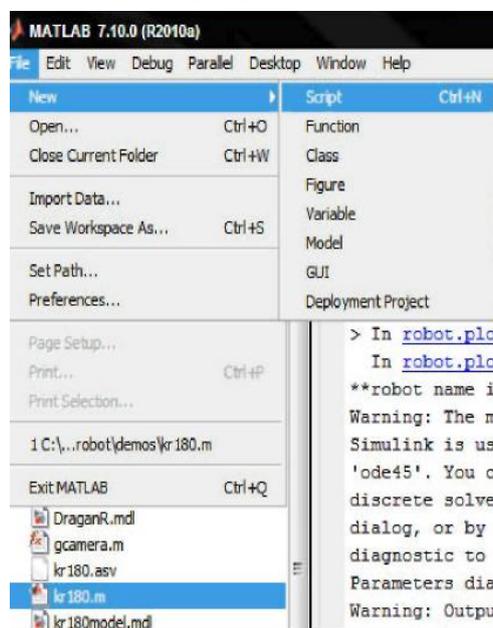
```
% The position of the center of mass of the segment relative to its coordinate system [xyz]
```

```
L{1}.r = [ 0 0 0.3];
```

```
L{2}.r = [ -0.1 -0.1 0];
```

```
L{3}.r = [ 0.1 0.05 0];
```

```
L{4}.r = [ 0 0 0.18];
```

Figure 5. Creating a new script for robot $kr180$

```

%Tensor segments inertia elements to its center mass L{1}.I = [ 0 0.35 0 0 0 0]; L{2}.I = [ .13
.524 .539 0 0 0]; L{3}.I = [ .066 .086 .0125 0 0 0]; L{4}.I = [ 1.8e-3 1.3e-3 1.8e-3 0 0 0];
% Inertia of the segment L{1}.Jm =
200e-6; L{2}.Jm = 200e-6; L{3}.Jm =
200e-6; L{4}.Jm = 33e-6;
% Transmission ratio crank/segment L{1}.G = -
62.6111; L{2}.G = 107.815; L{3}.G = -53.7063;
L{4}.G = 76.0364;
%Viscous friction L{1}.B =
1.48e-3; L{2}.B = .817e-3; L{3}.B
= 1.38e-3; L{4}.B = 71.2e-6;
% Coulomb friction L{1}.Tc = [ .395 -.435]; L{2}.Tc
= [ .126 -.071]; L{3}.Tc = [ .132 -.105]; L{4}.Tc = [
11.2e-3 -16.9e-3];
kuka = robot(L, 'KR180', 'KUKA Robotics', 'params of 8/95'); kuka.name = 'KR180';
kuka.manuf = 'KUKA Robotics';

```

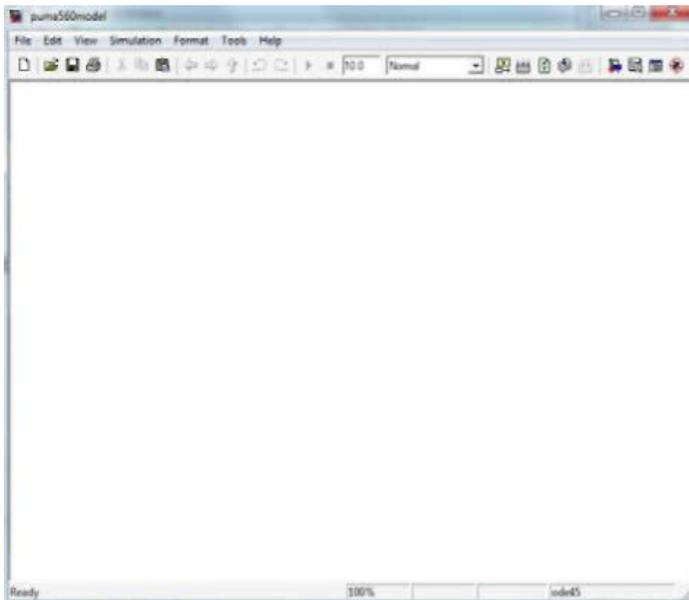


Figure 6. Display window of the new model

After creating the script `krl80.m` we move to creating a model of the robot using blocks from Robotics toolbox.

First, from the main menu Matlab we are choosing File - New - Model thus create a new Simulink model. With File - Save As we will save model under desired name (eg. `model420`). We will get a blank window model as shown:

Open the library `robblocks.mdl` which is located in the directory `demos`. Within this library is a robot model `krl80` with all blocks for its simulation, which are provided to a Robotics Toolbox.

For a start we are interested in block `fkine` which serves for calculation direct kinematics. Put it in `krl80model` and by double-clicking on it we modify its name in `kuka` which we connect his functional parameters with parameters model `krl80` defined Robotics Toolbox.

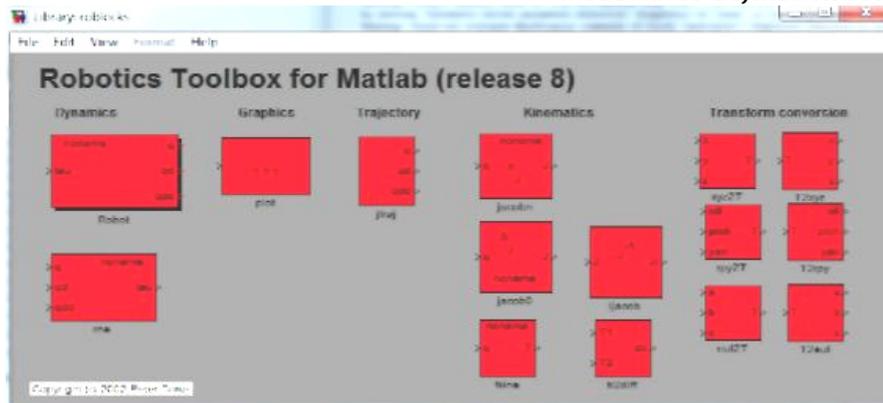


Figure 7. The appearance of a window Roboblocks

In order to provide a graphical display of the robot paste block plot from Robotics Toolbox in the created model. Identical to the previous we change name into `kuka`, which we also made connection functional parameters with parameters `robot` model `krl80` defined Robotics Toolbox.

Obtained from the Simulink block function library Robotics Toolbox can first select and drag the mouse into Simulink model block `jtraj`. Block `jtraj` calculated trajectory cranks (in their respective coordinate systems) between the two set positions: initial q_0 and final q_f . Specifies the block back in `jtraj` and `tmax`, which is set to time how long the simulation will last (applied in the case 10 - means a second). q_0 and q_f values and q_f can be typed directly into the `jtraj` block.

Setting parameters of Simulink blocks is carried out in the dialog window that gets by double-clicking on a given block. Dialog box for setting `jtraj` block is shown in Figure 8.

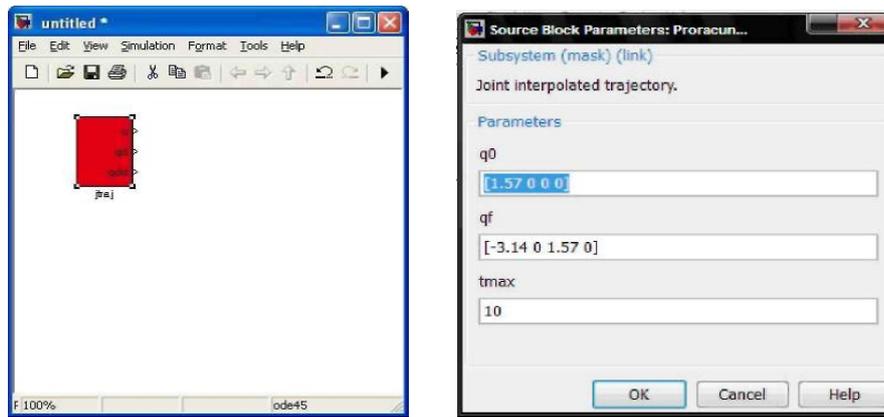


Figure 8. A dialog window

Block *fkine* solve direct kinematics task. For each crank position of the trajectory from the block *jttraj* gets the position (transform matrix *T*) manipulators. View in Cartesian coordinates based on homogeneous transformation matrix *T* is achieved by using the block *T2xyz*. For each position in the trajectory can be visualized set the appearance segments and cranks of the robot using block *plot*.

To observed the tool path in the projection plane imports into the robot model draw graphics *XY Graph* from *Simulink - Sinks* library. Since this block is drawn only manipulators path in the *XY* plane, and we want to show in all three projections, we import block for conversion and transformation of the output signal *T2xyz* from *Robotics Toolbox* robblocks library that we associate with access *fkine* robot. It divides the signal to the *X*, *Y* and *Z* component, so if we want to get the graph of paths in all three projections we import another 2 *XY Graph* block to rename the *XZ Graph* and *YZ Graph* and connect it to the analog output *T2xyz* block.

If we are interested in the numerical values manipulator position of the absolute coordinate system, we add the *Display* block from *Simulink - Sinks* library.

Benefit may be in graphic format of change angles during the cycle. He does this through a *Scope* block that is also imported from *Simulink - Sinks* library. Speed is obtained by differentiating the current position and the acceleration differentiating speed.

After adding these blocks in the *Simulink* model and their connecting is obtained model as in Figure 9.

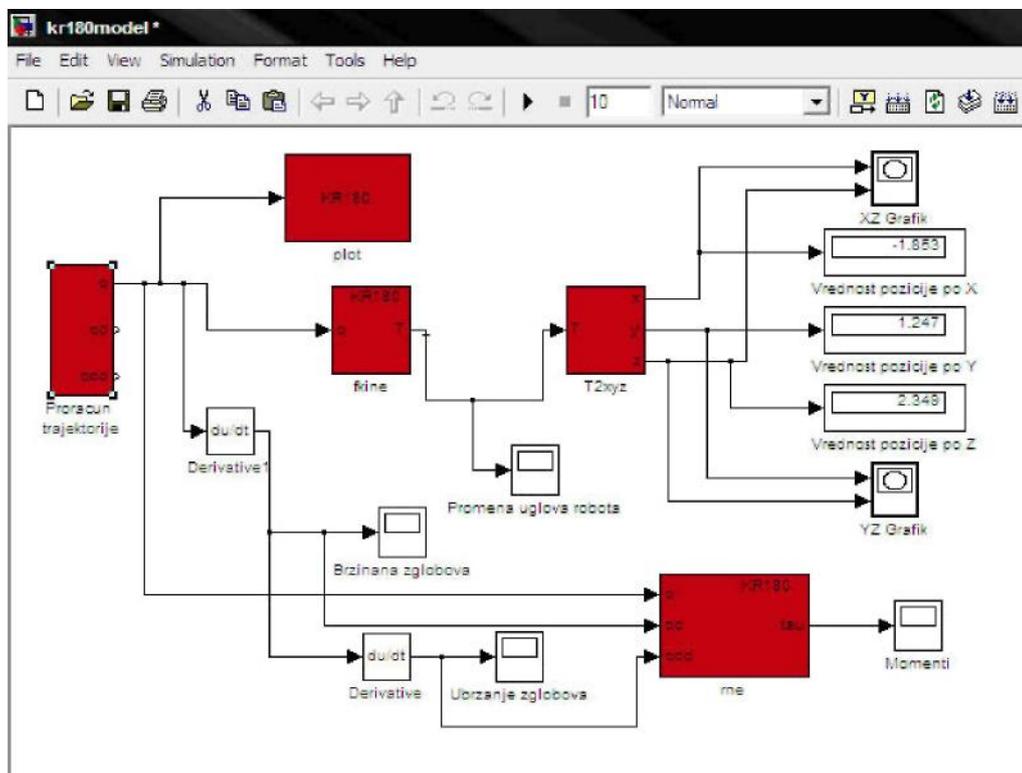


Figure 9. The Simulink model

Rne block is used to calculate the inverse dynamics and for his work requires an initial signal from *jttraj* block, speed and acceleration of each segment. Speed is obtained by differentiating the initial signal from the acceleration *jttraj* block differencing speed. Exit the *rne* block connects *Scope* block displaying moments with the help of input parameters.

SIMULATION OF INDUSTRIAL MANIPULATORS IN MatLab

By running the file from the MatLab left panel `krl80model.mdl` loading robot model `krl80` its visual display segments and crank at the home position that we have defined in the block `jtraj`.

Before starting the simulation time can be adjusted during which simulates the operation of the robot, in the dialog box shown on the slide, which is obtained by clicking on the menu command `Simulation - Configuration Parameters ...` from the Simulink model window. This time must coincide with the time entered for the parameter `tmax` block `jtraj`.

In the Simulink model simulation is started by clicking on the tool. On the picture it shows some positions in the robot movement calculated by block `jtraj`, by position defined coordinates of the crank, to the position `[3.14 0 1.57 0]` from which we enter directly into the block `jtraj`.

Positions were obtained by pausing the simulation. Pausing is done by using tool, available from the Simulink model window with the simulation in progress.

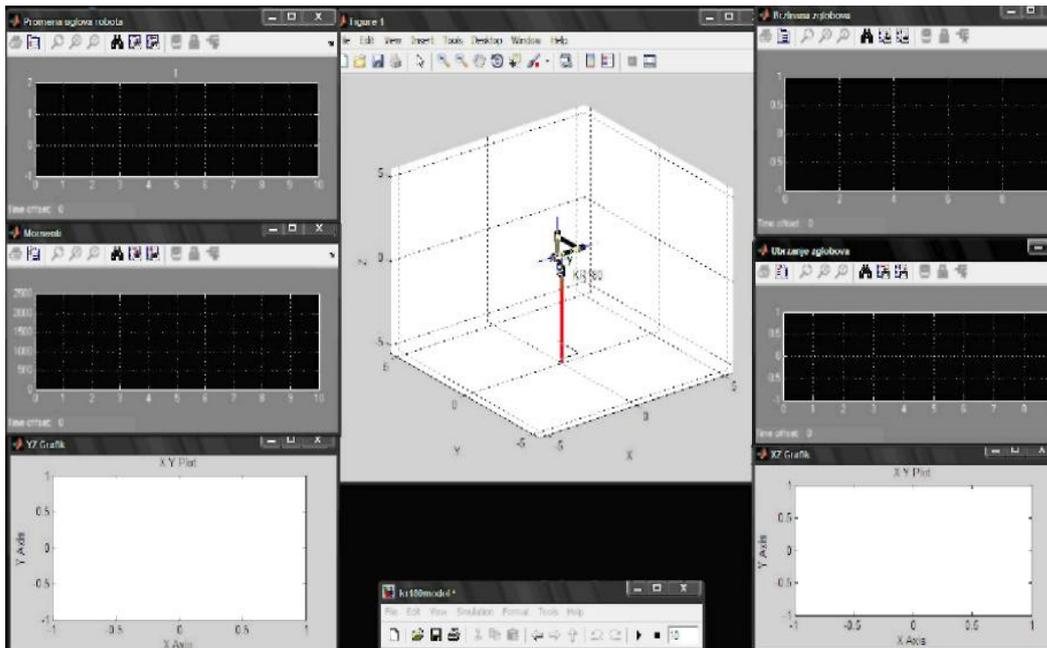


Figure 10.

Some moments robot simulation `krl80.m`

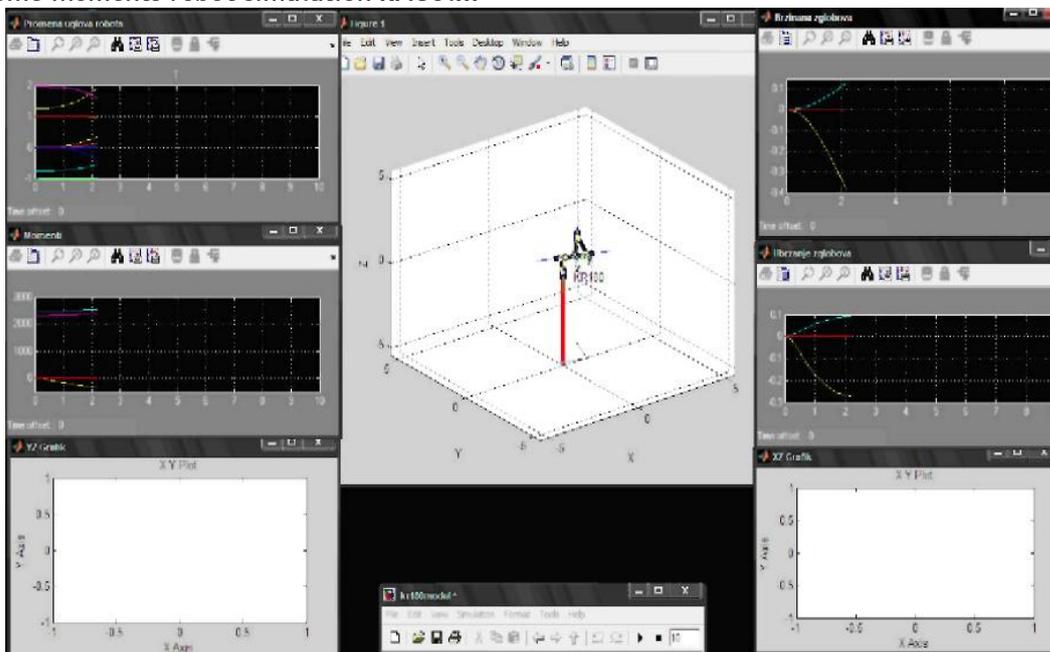


Figure 11.a. Certain moments robot simulation

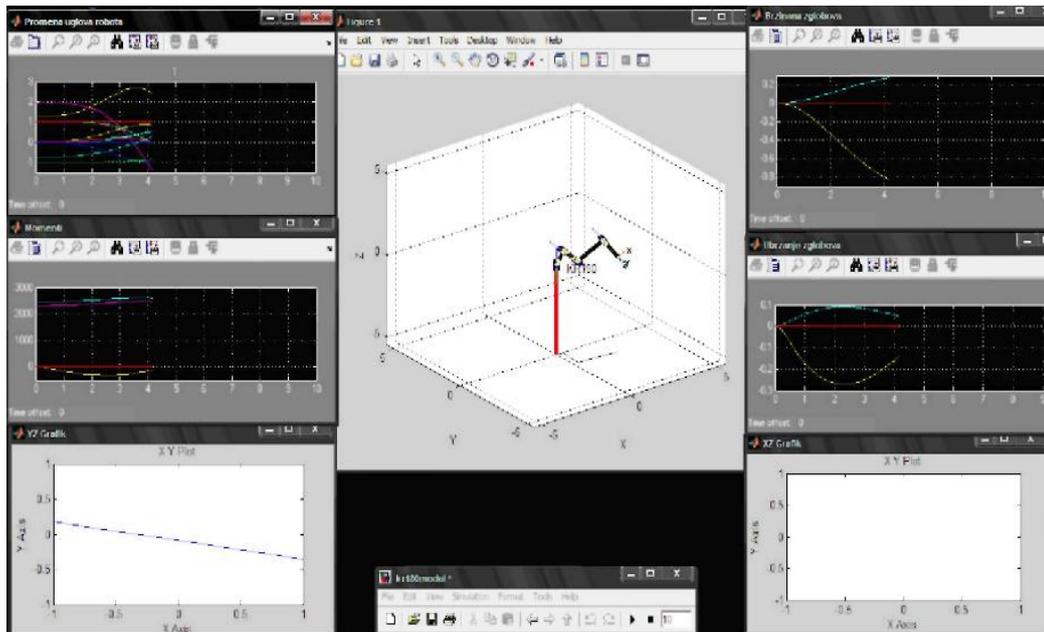


Figure 11.b. Certain moments robot simulation

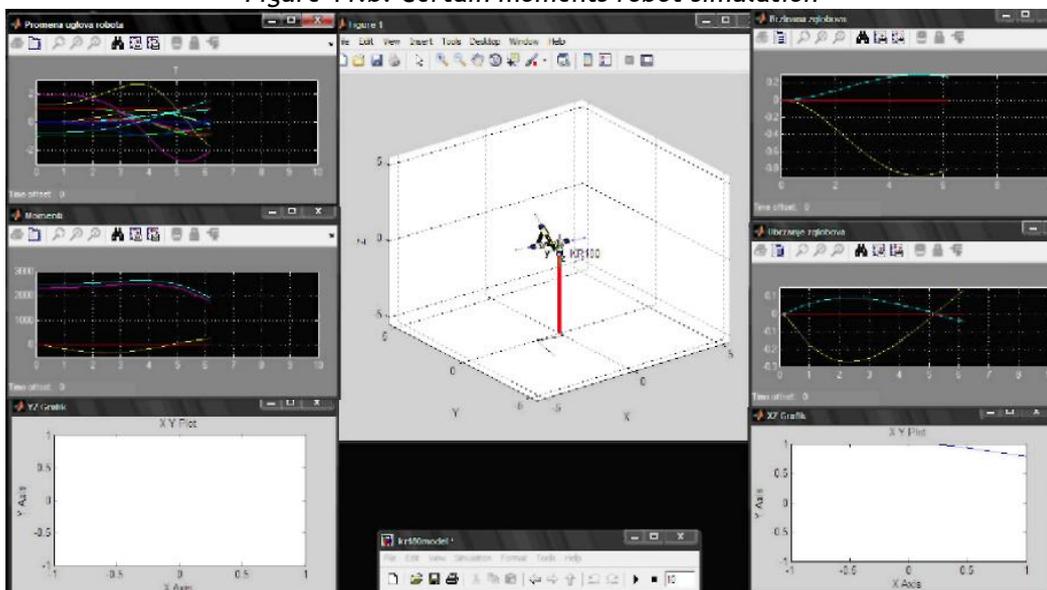


Figure 11.c. Certain moments robot simulation

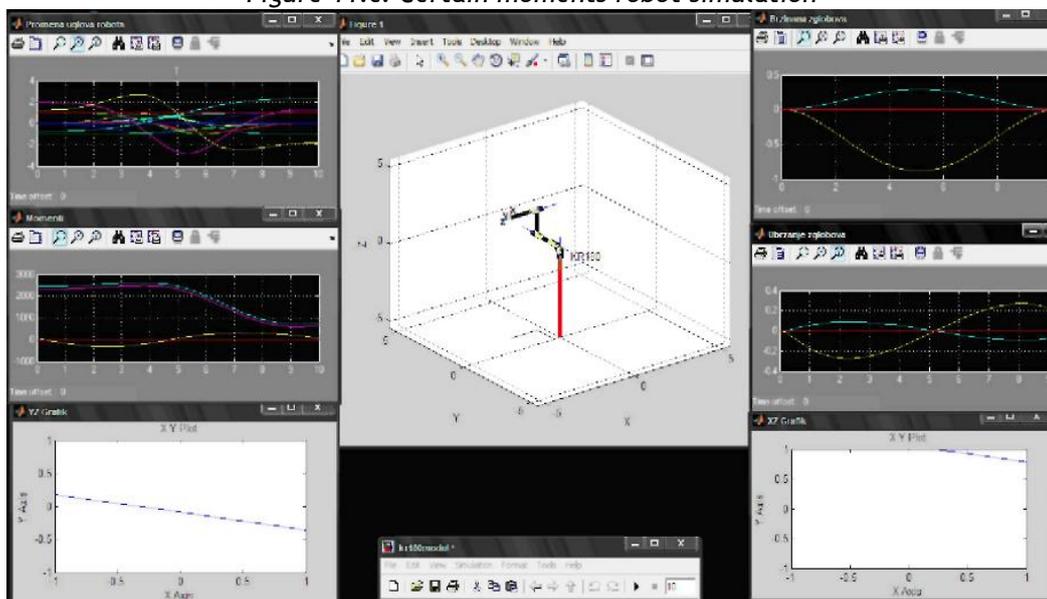


Figure 11.d. Certain moments robot simulation

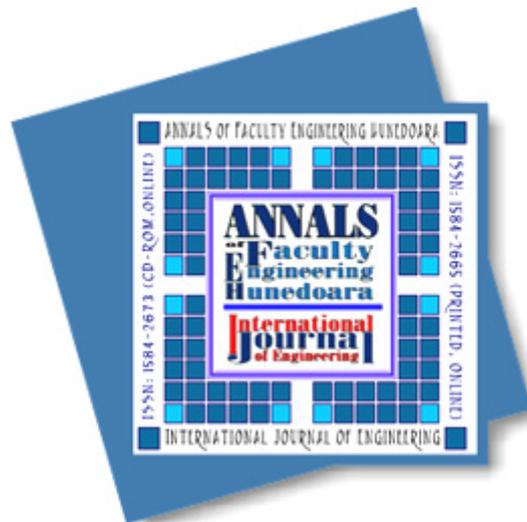
CONCLUSIONS

Through this example we can see that the modeling and simulation of an industrial robot can do without much effort and calculating as it was needed until recently. This type of modeling and simulation is suitable for mastering basic courses in designing and simulation of a robot movement. The most important thing is that all this has been made available to all those who want to try to create such a project.

Matlab proved that doing robotics is not more time-consuming, difficult and expensive process. Its possibilities MatLab opens a new interest among designers and projectors in robotics areas.

REFERENCES

- [1] D. Golubovic I. Milicevic, "Applying transformation matrix in solving the kinematics and dynamics of robots manipulacionih", Technical Faculty, Cacak, 2009
- [2] D. Golubovic, "Engineering Mechanics - advanced course," Faculty, Cacak, 1987
- [3] D. Ristic, SIR_1 robot simulation, seminars in subjects of modeling and simulation of mechanical computer, Technical Faculty Cacak, 2011.
- [4] V. Aleksic, simulation robot Puma 560, seminar subjects mechanical computer modeling and simulation, Technical Faculty, Cacak, 2010.
- [5] S. Đurasović, Robot simulation M420iA, seminar subjects mechanical computer modeling and simulation, Technical Faculty, Cacak, 2011.
- [6] S. Stefanovic, Z. Janjic, Ž. Adamović, D. Mikic, Construction foundations of modeling robot - manipulators with five degrees of freedom, II International Conference Industrial Engineering And Environmental Protection 2012 (IIZS 2012), Oktober 31st, 2012, Zrenjanin, Serbia



ANNALS of Faculty Engineering Hunedoara



- International Journal of Engineering

copyright © UNIVERSITY POLITEHNICA TIMISOARA,
 FACULTY OF ENGINEERING HUNEDOARA,
 5, REVOLUTIEI, 331128, HUNEDOARA, ROMANIA
<http://annals.fih.upt.ro>