

Design, Simulation and Control in Virtual Reality of a RV-2AJ robot

Mircea COMAN¹, Sergiu-Dan STAN¹, Milos MANIC², Radu BĂLAN¹

¹Dept. of Mechatronics, Technical University of Cluj-Napoca, Romania

²University of Idaho, Idaho Falls (UIIF), USA

e-mail: coman_mircea1@yahoo.com, sergiustan@iee.org, misko@uidaho.edu, radubalan@yahoo.com

Abstract—The paper presents the realization of a geometrical model of the RV-2AJ robotic arm built by Mitsubishi using the CAD environment SolidWorks. The geometrical modelling was done according to the robots dimensions and without putting in virtual model the parts that don't have a major influence in his kinematics; like screws, nuts or wires. The robots arm kinematics where then resolved and its schematics where presented. The direct kinematic problem, that shows how to calculate the position of the end-effector when the angles in the structures joins are known, is then presented with his equations. Also the inverse kinematics problem is resolved and the calculations that were made to obtain the equations are presented. The model was then exported from the geometrical modelling environment to Matlab, were its behaviour was simulated and was visualized in a virtual reality environment. To simulate the robots arm behaviour various types of inputs where used to control the virtual model movement. These control signals will be changed from implemented predefined curves, and signals that are generated by external devices. The external device that was used to obtain the control signal was a normal computers joystick.

I. INTRODUCTION

Industrial robots are essential components of today's factory and even more for the factories of the future. The demand for the use of robots stems from the potential for flexible, intelligent machines that can perform tasks in a repetitive manner at acceptable cost and quality levels.

Integration of man in the process is essential for a rapid setup, programming and robot system maintenance. Three elements of the controller are important for effective human integration: human input / output devices, giving the operator the information available in graphical form, and modes of operation that include human interaction. Some robots have teleoperated type entries devices, which allow manual input commands for coordinated movement. They are extremely useful in the learning of positions stage. Graphic interfaces available on some industrial robots are very effective for transmitting information to the operator quickly and efficiently.

A graphic interface is important for applications that require frequent programming and configuration changes. There are several software programs from third-party suppliers, which are very useful for off-line programming. These systems use computer models of commercially

available robots to simulate the motion trajectory and provide fast programming functions. In this article is presented the modelling and model control of the RV-2AJ robot. The geometrical modelling is done in SolidWorks and the simulation and the control of the model is done in Matlab/Simulink.

II. ROBOT KINEMATICS

In the study of any mechanical structures stand kinematics analysis. This analysis helps to understand the mechanical system, allowing the development of further studies or applications. For the kinematics analysis the schematization of the structure and the calculation of direct and inverse problem, based on the structure and its dimensions, is needed.

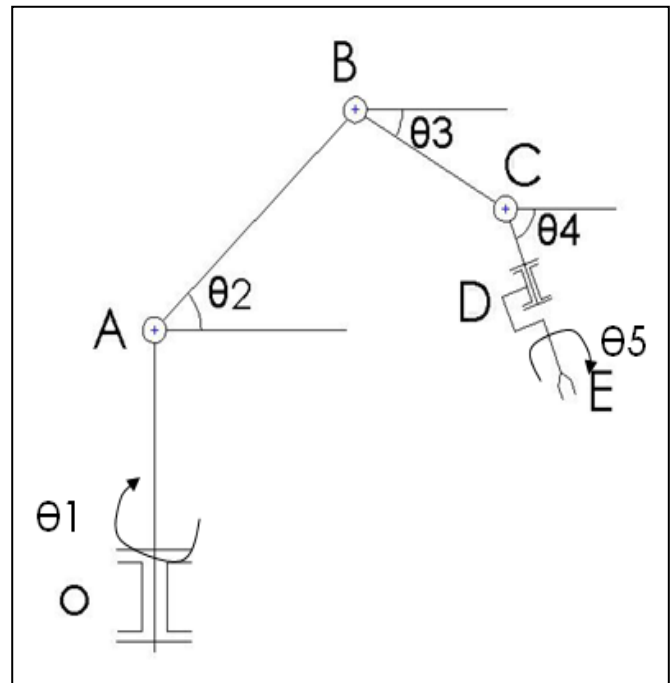


Fig.1 Kinematics scheme

A. Direct kinematics problem (DKP)

If the origin of the coordinates system is identical with the point A on the robotic arm, then the following equations are used to determine the positions of the end-effector, meaning the point E.

$$\begin{cases} X = \cos(\theta_1) \cdot [AB \cdot \cos(\theta_2) + BC \cdot \cos(\theta_3) + (CD + DE) \cdot \cos(\theta_4)] \\ Y = \sin(\theta_1) \cdot [AB \cdot \cos(\theta_2) + BC \cdot \cos(\theta_3) + (CD + DE) \cdot \cos(\theta_4)] \\ Z = OA + AB \cdot \sin(\theta_2) + BC \cdot \sin(\theta_3) + (CD + DE) \cdot \sin(\theta_4) \end{cases} \quad (1)$$

To make the calculations the angles $\theta_1, \theta_2, \theta_3, \theta_4$ and θ_5 is assumed to be known, and the length of OA, AB, BC, CD and DE are also know, thus the size of components robotic arm are:

OA=297.5 mm	AB=250 mm
BC=160 mm	CD=65.3mm
DE=79.3 mm	

B. Inverse kinematics problem (IKP)

The inverse problem involves the calculation of each angle of the robot arm knowing the position and orientation of the end-effector. In this case is they are represented by the point E coordinates and the DE element orientation. The entry data will be so XE, YE, ZE, θ_5 and the outputs will be $\theta_1, \theta_2, \theta_3, \theta_4$. The formulas used to calculate the angles in the joints are presented below:

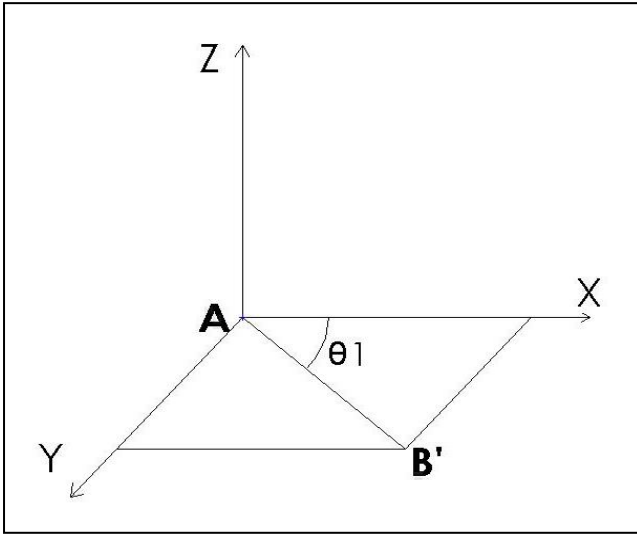


Fig. 2 Calculation of θ_1

$$\theta_1 = \arctg 2 \left(\frac{Y}{X} \right) \quad (2)$$

$$AB' = \sqrt{X^2 + Y^2} \quad (3)$$

B(X, Y, Z)-end-effector position

$$z = AB' \cdot \tan(\theta_z) \quad (4)$$

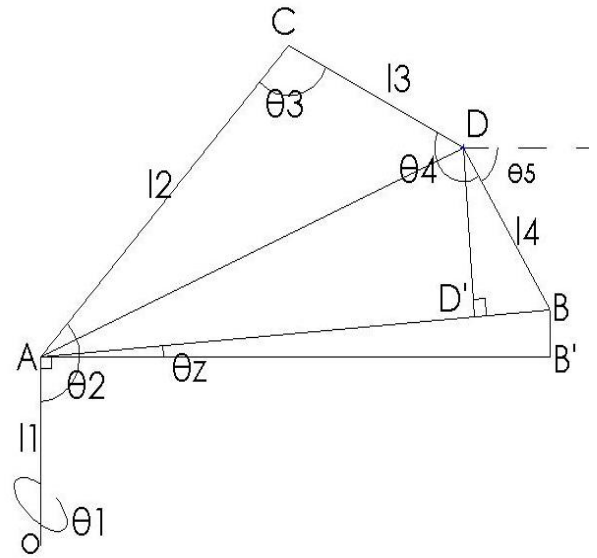


Fig. 3 Joints angles calculation

$$\theta_z = \text{atan} \left(\frac{z}{AB'} \right) \quad (5)$$

$$DD' = l_4 \cdot \sin(\theta_5 + \theta_z) \quad (6)$$

$$D'B = l_4 \cdot \cos(\theta_5 + \theta_z) \quad (7)$$

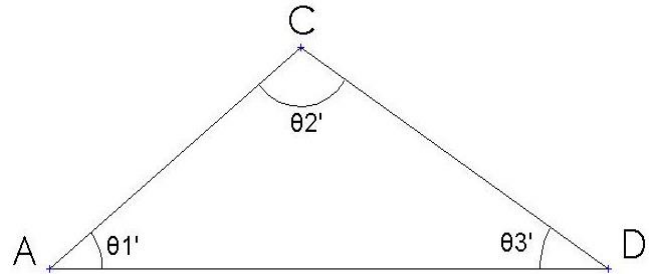
$$AD' = AB - D'B = AB - l_4 \cdot \cos(\theta_5 + \theta_z) \quad (8)$$

$$AD = \sqrt{(DD')^2 + (AD')^2} \quad (9)$$

$$AD = \sqrt{AB^2 + l_4^2 - 2 \cdot AB \cdot l_4 \cdot \cos(\theta_5 + \theta_z)} \quad (10)$$

$$AC = l_2 \quad (11)$$

$$CD = l_3 \quad (12)$$



$$CD^2 = AC^2 + AD^2 - AC \cdot AD \cdot \cos(A) \quad (13)$$

Using (12) for calculating every angle we obtain:

$$\theta_1' = \text{acos} \left(\frac{l_2^2 + l_5^2 + AB^2 - 2AB * l_5 * \cos(\theta_5 + \theta_z) - l_3^2}{2l_2(\sqrt{AB^2 + l_5^2 - 2AB * l_5 * \cos(\theta_5 + \theta_z)})} \right) \quad (14)$$

$$\theta_2' = \text{acos} \left(\frac{l_2^2 + l_3^2 - l_5^2 - AB^2 + 2AB * l_5 * \cos(\theta_5 + \theta_z)}{2l_2 l_3} \right) \quad (15)$$

$$\theta_3' = \text{acos} \left(\frac{l_3^2 + l_5^2 + AB^2 - 2AB * l_5 * \cos(\theta_5 + \theta_z) - l_2^2}{2l_3(\sqrt{AB^2 + l_5^2 - 2AB * l_5 * \cos(\theta_5 + \theta_z)})} \right) \quad (16)$$

In conclusion the resulting angles will be:

$$\theta_1 = \text{acos} \left(\frac{Y}{X} \right) \quad (17)$$

$$\theta_2 = \theta_z + \theta_1' + \text{acos} \left(\frac{DD'}{AD} \right) \quad (18)$$

$$\theta_3 = \theta_2' \quad (19)$$

$$\theta_4 = \theta_3' + \text{acos} \left(\frac{DD'}{AD} \right) + (90^\circ - \theta_5 - \theta_z) \quad (20)$$

The last angle can be found subtracting $\theta_1, \theta_2, \theta_3, \theta_4$ from 540° .

III. MODELING AND SIMULATING THE ROBOT ARM

A. Geometric Modelling with SolidWorks

First a virtual geometric model was built using the real dimensions of the robot. As modelling environmental a CAD program "SolidWorks" was used. Each of the robotic arm parts was modelled individually, ignoring the smaller parts that do not significantly affect the dynamics of the robot, to facilitate the calculations. If that had been shaped it would have been difficult to calculate the dynamics because that simulation requires too much processing power for a normal PC. Thus were shaped 7 of the robotic arm elements and they were named to facilitate future operations, as follows: base, shoulder, 1 arm, 2 arm, 3 arm, flange and end-effector. After modelling, the individual components have been assembled into a single file. After entering the parts, the rotation joints of the robot have been introduced in the model. Also when virtual assembling the robot, beside the normal constrains, the elements where put in a vertical position. This position was adopted to facilitate the following steps; the assembled position is regarded as the reference, and all the variations of the joints angles are calculated according to this position. The resulting model is presented in Fig.4.

B. Simulation of the model

For simulating the robot the program Matlab was used. The geometrical model has been exported from SolidWorks by saving it in "*.xml" format, and then imported in Matlab. By doing this the file will be converted, thus generating a dynamic model which will be analyzed and controlled. By

converting it the robot arm will be represented by blocks for each element and each joint, blocks that are connected according to the structure of the robot. The outputs of the blocks, representing parts, actually transmit the positions and orientations of the elements coordinates systems that are related to the global coordinates system. These coordinates systems have identical values for the pairs of values that enter in the same joint. In addition to these outputs for each element they can be attached others systems of coordinates. This operation of attaching coordinates system will be used later to make a virtual image of the robot.

Blocks representing the rotation joints have, at the beginning, two entries for each element which that they connect. Also in addition to these, the operator can add additional parts, which can be chosen as inputs or outputs for sensors or actuators. Although this robotic arm has only rotation joints similarly may be introduced, if necessary, any type of connection (translation, plane, spherical, universal), thus allowing the building of any type of configuration and to modify it.

The model generated by Matlab is presented in Fig. 5. To build a virtual reality in Matlab we used the "Virtual Reality Toolbox" module. This was made by exporting the model form SolidWorks, by saving it with the "*.wrl" extension.

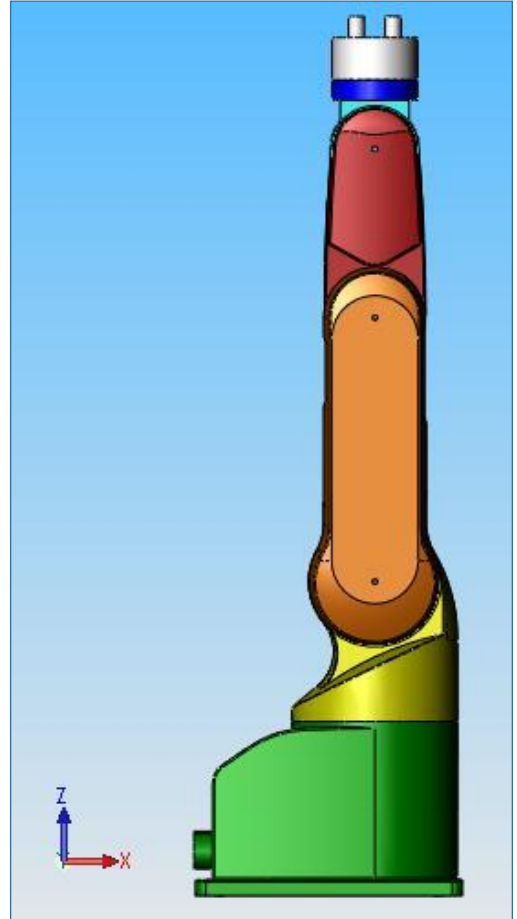


Fig. 4 The geometrical model of the robot

Then the file was edited using the VRealm Builder. After that a new block was added to Matlab model which is used to visualize the model. Also in here the simulation can be recorded in a video. This block has as inputs the variation of elements positions and orientation. To obtain these variations the blocks presented in the Fig. 6 are used. These body sensor are attached to each parts of the robot arm and their outputs are transmitted to the virtual reality block.

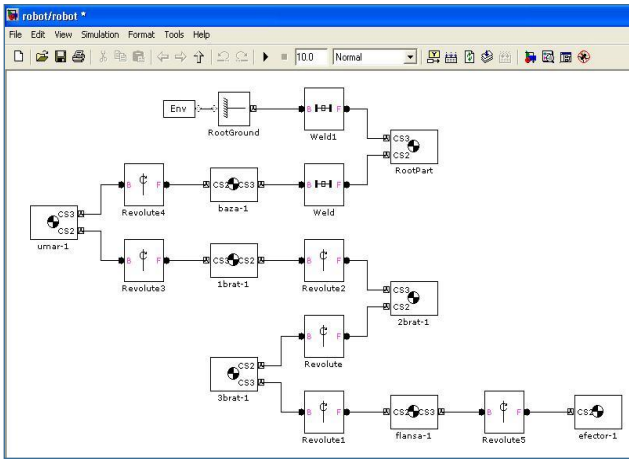


Fig. 5 The robot model generated by Matlab

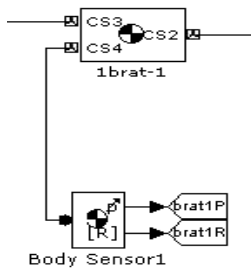


Fig. 6 Body sensor block

The next step involves controlling the robots motions, meaning controlling the structure joints. Sensors will also be added for each joint. By controlling the robot joints they will become actuators of which parameters can be controlled. This is achieved by adding an additional port to the joints blocks. At this port will connect a block called "Joint Actuator", which controls the movement or the force/moment of the actuator. After these operations these operations the model will look like in the Fig. 7.

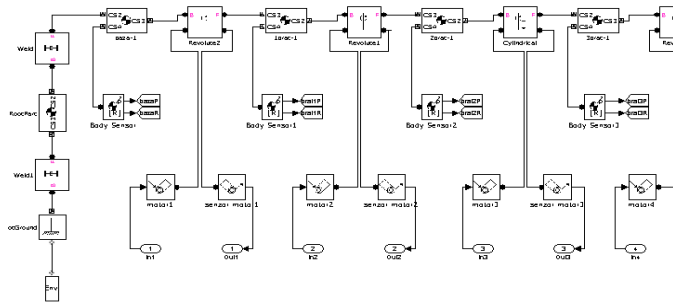


Fig. 7 A part of the robot model equipped with sensors and actuators

C. Motion control of the model

For controlling the robot the equations of the inverse problem are put in an embedded function block that calculates the angular positions of the joints. This block also limit's the positions that are outside the robot working range. In case that such a position is introduced as an input the output will be a position nearest to these inputs. The code that has been writhed in this block begins whit a statement that sets the name of the block, the input variable and the output variable:

```
function [t1,t2,t3,t4,t5p]= fcn(x,y,z,t5)
```

The limitations that where added consist in limiting the argument of the asin and acos functions as follow:

```
if m1>1
    m1=1;
else
end
if m1<-1
    m1=-1;
else
end
```

The resulting code in the embedded function block is:

```
function [t1,t2,t3,t4,t5p]= fcn(x,y,z,t5)
l4=70;
ABi=sqrt(x^2+y^2);
z1=z-297.5;
tz=atan(z1/ABi);
AB=ABi/cos(tz);
dd=l4*sin(t5+tz);
ad=sqrt(l4^2+AB^2-2*AB*l4*cos(t5+tz));
m1=( 36900+ad^2)/(2*250*ad);
if m1>1
    m1=1;
else
end
if m1<-1
    m1=-1;
else
end
t1p=acos(m1);
m2=( 88100-ad^2)/(500*160);
if m2>1
    m2=1;
else
end
if m2<-1
    m2=-1;
else
end
t2p=acos(m2);
m3=(-36900+ad^2)/(320*ad);
if m3>1
    m3=1;
else
```

```

end
if m3 < -1
    m3 = -1;
else
    end
t3p = acos(m3);
t1 = atan2(-y, -x);
m5 = dd/ad;
if m5 > 1
    m5 = 1;
else
    end
if m5 < -1
    m5 = -1;
else
    end
t2 = tz + t1p + asin(m5) + pi/2;
t2 = -t2;
t3 = t2p;
t5p = t5 - tz;
t4 = t3p - t5 + acos(m5) + pi/2 - tz;

```

These values are then processed so that they can be used as inputs for the joints actuators blocks. The making of signal consists in obtaining for each joint a control signal consisting in the position, speed and acceleration of the actuator. These signals are created by deriving the position and then deriving the speed resulted. The blocks that are used to make this are presented in Fig. 8

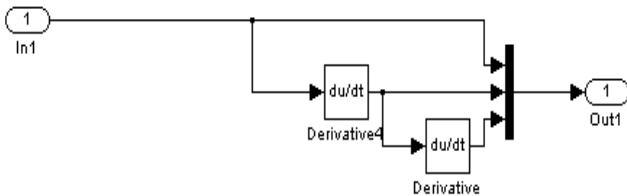


Fig. 8 Signal building for actuator blocks

Beside this the signals where limited so that their values do not exceed the angles that the real robot can achieve. The values that joints of the real robot can reach are presented below:

- Actuator 1 : $-150^0, +150^0$
- Actuator 2 : $-60^0, 120^0$
- Actuator 3 : $-110^0, +120^0$
- Actuator 4 : $-90^0, 90^0$
- Actuator 5 : $-200^0, +200^0$

The block that enters these limitations and constructs the signal for the motion control is presented in Fig. 9

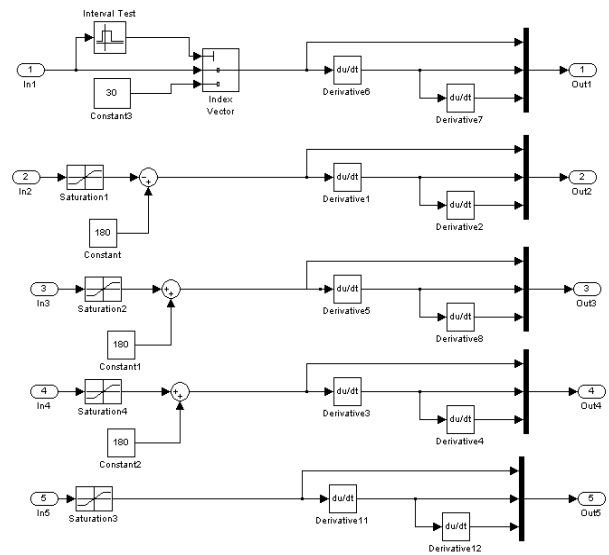


Fig. 9 Signal processing block

The signal processing block also adds or subtracts, where is needed, because of the differences between the reference system that is use in the calculating the inverse cinematic and the reference position that the virtual model has.

As inputs sources for the block that calculates the inverse cinematic, meaning X, Y, Z and θ_5 , it was used two methods. The first was using a user defined block that generated a specific trajectory (circle, square, spiral etc.). The second way was an interactive method that consists in using a joystick to control it. The signal from the joystick where filtered, amplified use and then to control the speed of the end-effector for each axe, and his orientation. The blocks that do that are illustrated in the Fig. 10.

Beside these blocks it was added to the model a virtual scope to better visualize the trajectory of the end effector. These results where compared with the input signals to verify the model. In this case the input signals and the position of the end-effector are identical at all time, because the actuators are considered to be ideal. This means that they move instantly to the command signals and they don't need time to reach the commanded position and orientation.

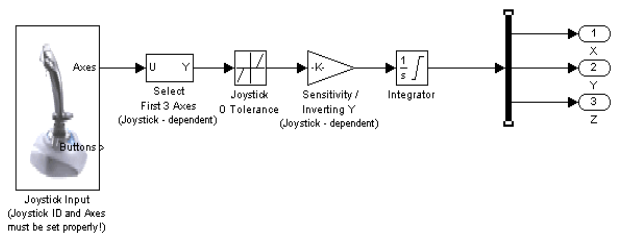


Fig. 10 System use to collect and use joysticks as inputs

IV. CONCLUSION

The need of serial robots is very high due to their large workspace and high speed. Because there is such a large variety of models and the high prices the need for their

simulation has increased. The simulation process reduces the costs of testing and makes later control to much easier. Using simulated models in the designing faze makes the observation of the advantages and disadvantages, of a certain model, to be much more easy. The modifications on a structure made in a virtual environment can be very easily parameterized and the differences that appear can be visualized to be analyzed. The visualization in a virtual reality of the robot model it is very useful in the early stages of design, because it helps to understand better the processes that are simulated.

The virtual simulation of the robot is also the first step in actually controlling the mechanical structure. With the simulation, the control signals that are going to be used to physically control the robot can be obtained. Also the control algorithm that is going to be used to achieve the desired speed and precision is going to be implemented in the simulation. The input signals that are going to be use for controlling to command the way the robot moves are going to be also processed in the simulation. These signals can come from an external device connected to the computer, a joystick, keyboard, mouse or any other peripheral device. Also in the simulation other types of signal can be processed. For example a video processing block can be introduced in Matlab. This block will analyze the signals received from a video camera and following a predefined algorithm will generate the movement trajectory and the control signal for the robot so that the necessary operations to be successfully executed

REFERENCES

- [1] C. Gosselin. "Determination of the workspace of 6-d.o.f. parallel manipulators". *ASME Journal of Mechanical Design*, 112:331-336, 1990. 9.
- [2] Radu Balan, Vistrian Maties, Olimpiu Hancu, Sergiu Stan, A model based predictive control algorithm applied to Nonlinear processes, The 5th Edition of IFAC Knowledge and Technology Transfer Conference Series on Automation for Building the Infrastructure in Developing Countries, May 17-19, 2007 – Golden Dolphin Hotel, Cesme, Turkey, pag. 345-350.
- [3] Stan, Sergiu-Dan; Maties, Vistrian; Balan, Radu; Multiple-Goal Kinematic Optimization of 2 DOF Micro Parallel Robots, *Mechatronics and Automation*, 2007. ICMA 2007. International Conference on 5-8 Aug. 2007 Page(s):2535 - 2540.
- [4] Balan, Radu; Maties, Vistrian; Stan, Sergiu-Dan; Optimization of the Electrode Control System Using On Line Simulation and Rule Based Control, *Mechatronics and Automation*, 2007. ICMA 2007. International Conference on 5-8 Aug. 2007 Page(s):2939 – 2944.
- [5] Stan, Sergiu-Dan; Maties, Vistrian; Balan, Radu; Workspace Optimal Design of a 2 DOF Micro Parallel Robot Using Genetic Algorithms and Simulated Annealing Optimization Methods; *Mechatronics and Automation*, 2007. ICMA 2007. International Conference on 5-8 Aug. 2007 Page(s):1108 - 1113.
- [6] Balan, R. Maties, V.Hancu, O., Stan S.: "The measure of humidity and temperature using smart sensors", pp 329..336 în "Proceedings of the 34 International Symposium on Agricultural Engineering" Opatija, Croatia, 21-24 February 2006.
- [7] Balan,R., Vistrian Maties, Olimpiu Hancu, Sergiu Stan, A Predictive Control Approach for the Inverse Pendulum on a Cart Problem, *IEEE-ICMA 2005* pag. 2026-2031, CD-ROM Proceedings July 29 - August 1, 2005 Niagara Falls, Ontario, Canada.
- [8] Balan, R., V. Maties, S. Stan, A Solution of the Inverse Pendulum on a Cart Problem Using Predictive Control, *IEEE ISIE 2005*, June 20-23, 2005, Dubrovnik, Croatia, pag. 63-68.
- [9] Balan, R. V. Maties, S. Stan, Motion Control Using On-Line Simulation and Rule Based Control, *IEEE ISIE 2005*, June 20-23, 2005, Dubrovnik, Croatia, pag. 167-172.
- [10] J. P. Merlet. "Determination of the orientation workspace of parallel manipulators". *Journal of intelligent and robotic systems*, 13:143-160, 1995.
- [11] Gupta, A., et al., "Design, Control and Performance of RiceWrist: A Force Feedback Wrist Exoskeleton for Rehabilitation and Training", *The Int. J. of Robotics Research*, Vol. 27, No. 2, 233-251 (2008).
- [12] J. P. Merlet. "Determination of the orientation workspace of parallel manipulators". *Journal of intelligent and robotic systems*, 13:143-160, 1995.
- [13] A. Kumar, KJ. Waldron. "The workspace of mechanical manipulators". *ASME J. Mech. Des.* 1981; 103:665-672.
- [14] YC. Tsai, AH. Soni. "Accessible region and synthesis of robot arm". *ASME J. Mech. Des.* 1981, 103: 803-811.
- [15] KG. Gupta, Roth B., "Design considerations for manipulator workspace". *ASME J. Mech. Des.* 1982, 104(4), 704-711.
- [16] K. Sugimoto, Duffy J, Hunt KH, "Special configurations of spatial mechanisms and robot arms". *Mech Mach Theory* 1982, 117(2); 119-132.
- [17] KC. Gupta. "On the nature of robot workspaces", *Int. J. Rob. Res.* 1986; 5(2): 112-121
- [18] JK. Davidson, KH. Hunt, "Rigid body location and robot workspace: some alternative manipulator forms". *ASME J. Mech. Transmissions Automat Des* 1987, 109(2); 224-232.
- [19] SK. Agrawal, "Workspace boundaries of in-parallel manipulator systems". *Int. J. Robotics Automat* 1990, 6(3) 281-290.
- [20] C. Gosselin, Angeles J. "Singularities analysis of closed loop kinematic chains". *IEEE-T.Robotics Automat* 1990; 6(3) 281-290.
- [21] M. Cecarelli, "A synthesis algorithm for three-revolute manipulators by using an algebraic formulation of workspace boundary". *ASME J. Mech. Des.* 1995; 117(2(A)): 298-302.
- [22] S. K. Agrawal. "Workspace boundaries of in-parallel manipulator systems". *IEEE Transactions on Robotics and Automation*, 7(2):94-99, 1991.