Real-Time Evaluation of Inverse Kinematics for a 3-RPS Medical Parallel Robot using dSPACE Platform

Ciprian-Radu Rad[1], Milos Manic[2], *Member, IEEE*, Radu Balan[1], and Sergiu-Dan Stan[1], *Member, IEEE*
[1]Technical University of Cluj-Napoca, Romania,
[2]University of Idaho, Idaho Falls, United States of America,
cipri_rad@yahoo.com, misko@uidaho.edu, radubalan@yahoo.com, sergiustan@ieee.org

Abstract. **Parallel robots find many applications in human-systems interaction, medical robots, rehabilitation, exoskeletons, to name a few. In a real-world applications it is very important for the robots to compute certain tasks in a given sample time. Hence, the time allocated for computing inverse kinematics must be as small as possible. This paper presents a dSPACE 1104 based real-time experiment of two inverse kinematics methods for a 3-DOF medical parallel robot with R-P-S (revolute-prismatic-spherical) joint structure; meanwhile a demo experiment with MATLAB/Simulink diagram and its results are given.**

Keywords**: 3-RPS, dSPACE, Inverse Kinematics, MATLAB/Simulink, Parallel Robots, Real-time Simulations.**

I. Introduction

PARALLEL robots find many applications in human-systems interaction [1], medical environment [2-6] rehabilitation, exoskeletons, to name a few. They have great potential in the enumerated applications, thank to their fine characteristics of stiffness, positioning accuracy, etc. Parallel robots with 3-RPS structure have been intensively studied in the literature. Zhang used it for the balancing of a spaceship arm, as a micro-manipulator (under the name of *Artisan*) by Waldron and Khatib and as an entertainment motion base [7]. Kinematics and dynamics characteristics have been studied by Lee and Shah [8], [9] and Song and Zhang [10]. Amann [11] has made a simulation of inverse kinematics for this structure and verified the results with a real-world model to show the relevance of modelling and simulation robots in modern mechatronics. In real-world applications it is very important for the robots to compute certain tasks in a given sample time. Hence, the time allocated for computing inverse kinematics must be as small as possible.

This work was supported in part by CNMP through the grants no. 3280 (PARTENERIATE type), title of the project: “Complex mechatronics systems for medical applications” and CNCSIS 1072 (IDEI type), title of the project: “Researches regarding the advanced control with applications in mechatronics”.

MSc. Student Ciprian-Radu Rad, Dr. Radu Balan and Dr. Sergiu-Dan Stan is with the Technical University of Cluj-Napoca, Dept. of Mechanisms, Precision Mechanics and Mechatronics, Romania (e-mail: cipri_rad@yahoo.com, radubalan@yahoo.com and sergiustan@ieee.org).

Dr. Milos Manic is with University of Idaho, Dept. of Computer Science, USA (e-mail: misko@ieee.org).

Fig. 1. DS1104 R&D Controller Board

This paper is focused on the experimental research of two inverse kinematics methods to see which one is better in a real-time application. The experiments were realised using Simulink/Real-Time Workshop based on dSPACE real-time simulation system.

The Simulink/Real-Time Workshop based on dSPACE real-time simulation system is a famous development kit in mechatronics industry, and many international automobile manufacturers have adopted this system. The platform in this paper adopts DS1104 R&D controller board [12] from Fig. 1 and it’s resources is described in Fig. 2. DS1104 R&D Controller Board is a cost-effective system for controller development. It connects to the PC machine through PCI bus, all the real-time calculation is implemented in DS1104. The real-time hardware based on PowerPC technology and its set of I/O interfaces makes the controller board an ideal solution for developing controllers in various fields, such as drives, robotics, aerospace and automotives. ControlDesk, dSPACE's well-established experiment software, provides all the functions to control, monitor and automate experiments and make the development of controllers more efficient. Real-Time Interface (RTI) is the real-time realization software for dSPACE system, which extend the real-time C-code automate, generate software Real-Time Workshop, seamlessly intergrates the dSPACE system's Real-Time Kernel and I/O hardware model, automatically build, compile, link, download and execute the real-time C-code from Simulink model [13]. Further more, RTI

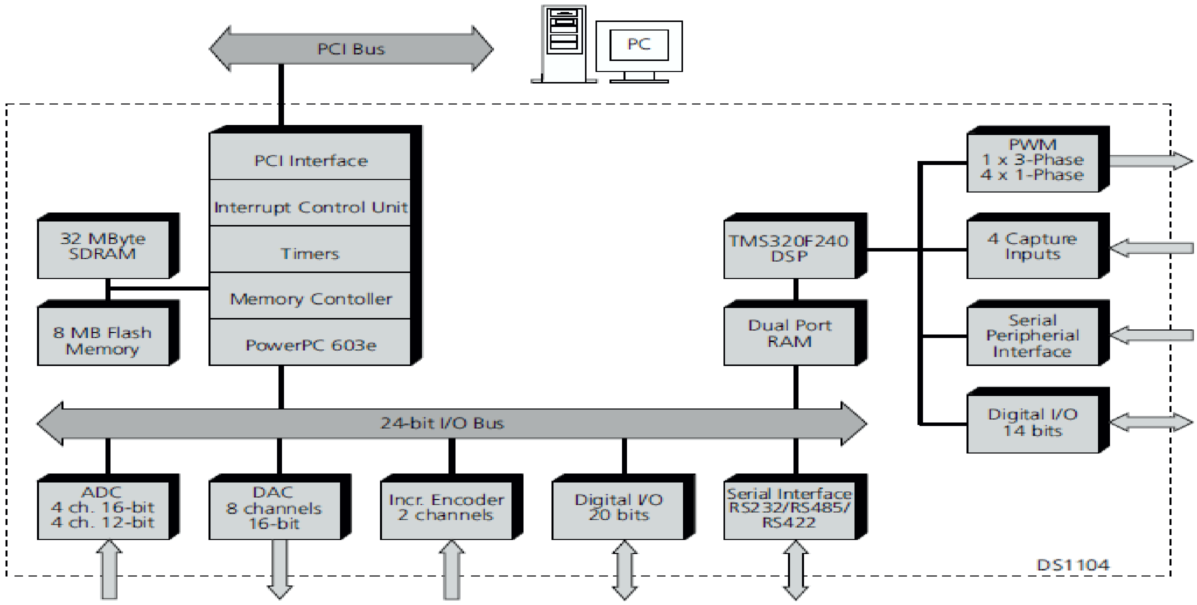

Fig. 2. Block Diagram of DS1104 R&D Controller Board

generate a variables file according to signals and parameters, and ControlDesk will access these variables and edit the parameters [14]. With Real-Time Interface (RTI), you can easily run your function models on the DS1 104 R&D Controller Board. You can configure all I/O graphically by dragging RTI blocks and reduce the implementation time to a minimum.

The remainder of this paper is organized as follows. The inverse kinematics methods are carried out in Section II. Real-time experiments are presented and analized in Section III. Finaly, this paper ends up with conclusions in Section IV.

II. Inverse Kinematics Methods

A. 3-RPS Mechanism

Fig. 3 presents the 3-RPS mechanism schematic. Three identical limbs connect the moving platform at points B_i by spherical joints and to the fixed base at points A_i by revolute joints. Each limb consists of an upper and a lower member connected by a prismatic joint d_i. These three prismatic joints are used as the inputs to the robot. Overall, there are eight links, three revolute joints, three prismatic joints and three spherical joints. Hence, the mechanism has three degrees of freedom which can be verified with *Grübler's* formulas [15]:

$$m = 6\cdot(l-n-1)+\sum_{i=1}^{n} d_i = 3. \tag{1}$$

where:

m = mechanism degrees of freedom;
l = number of rigid bodies (also the base);
n = number of joints;
d_i = joint i degrees of freedom.

Let $\mathbf{a}_i$ and ${}^{B}\mathbf{b}_i$ be the position vectors of points A_i and B_i in the coordinate systems $\{A\}$ and $\{B\}$, respectively. Then, the coordinates of A_i and B_i ($\Delta_{A_1A_2A_3}, \Delta_{B_1B_2B_3}$ equilateral triangles) are given by:

$$\mathbf{a}_1 = [g, 0, 0]^T, \tag{2}$$

$$\mathbf{a}_2 = [-\frac{1}{2}g, \frac{\sqrt{3}}{2}g, 0]^T, \tag{3}$$

$$\mathbf{a}_3 = [-\frac{1}{2}g, -\frac{\sqrt{3}}{2}g, 0]^T. \tag{4}$$

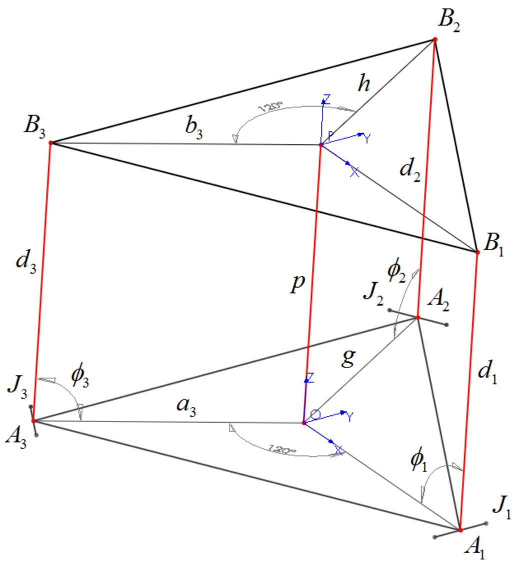

Fig. 3. 3-RPS mechanism schematic

$${}^{B}\mathbf{b}_1 = [h, 0, 0]^T, \tag{5}$$

$$^{B}\mathbf{b}_2 = [-\frac{1}{2}h, \frac{\sqrt{3}}{2}h, 0]^T, \quad (6)$$

$$^{B}\mathbf{b}_3 = [-\frac{1}{2}h, -\frac{\sqrt{3}}{2}h, 0]^T, \quad (7)$$

The inverse kinematics problem it's a common task in robotic systems. This is important in simulations, because given a trajectory to the robot the inverse kinematics returns the length of the actuators. Since the mechanism has only three degrees of freedom, the position and orientation must be specified in accordance with the constraints imposed by the revolute joints [15].

B. ZYZ Euler Angles Method

In terms of *ZYZ* Euler angles [13], it has been determined in [5] that the two orientation freedoms of the moving platform with respect to the base platform are the precession angle α and the nutation angle β. The cartesian translatory freedom is in the *Z* direction. The spin angle γ of the moving platform, with respect to the base platform, and the other two cartesian position variables in the *X* and *Y* has been determined to be:

$$\gamma = -\alpha, , \quad (8)$$

$$p_x = -\frac{1}{2}h(1-\cos(\beta))\cos(2\alpha), \quad (9)$$

$$p_y = \frac{1}{2}h(1-\cos(\beta))\sin(2\alpha). \quad (10)$$

For a prescribed position and orientation of the moving platform, the dependent variables are defined by (8) – (10), and (11) – (13) defines the length d_i of the links [8].

$$d_1^2 = g^2 + h^2 + p_x^2 + p_y^2 + p_z^2 - 2gp_x + 2h(C_\alpha^2 C_\beta + S_\alpha^2)(p_x - g) + \ldots$$
$$\ldots + h(C_\beta - 1)S_{2\alpha} p_y - 2hS_\beta C_\alpha p_z, \quad (11)$$

$$d_2^2 = g^2 + h^2 + p_x^2 + p_y^2 + p_z^2 + gp_x - \sqrt{3}gp_y - h(C_\alpha^2 C_\beta + S_\alpha^2 + \ldots$$
$$\ldots - \sqrt{3}C_\alpha S_\alpha (C_\beta - 1))(p_x + \frac{1}{2}g) - h((S_\alpha C_\alpha (C_\beta - 1) - \sqrt{3}(S_\alpha^2 C_\beta + C_\alpha^2)) + \ldots$$
$$\ldots + (p_y - \frac{\sqrt{3}}{2}g) + hS_\beta (C_\alpha - \sqrt{3}S_\alpha) p_z, \quad (12)$$

$$d_3^2 = g^2 + h^2 + p_x^2 + p_y^2 + p_z^2 + gp_x + \sqrt{3}gp_y - h(C_\alpha^2 C_\beta + S_\alpha^2 + \ldots$$
$$\ldots + \sqrt{3}C_\alpha S_\alpha (C_\beta - 1))(p_x + \frac{1}{2}g) - h((S_\alpha C_\alpha (C_\beta - 1) + \sqrt{3}(S_\alpha^2 C_\beta + C_\alpha^2)) + \ldots$$
$$\ldots + (p_y + \frac{\sqrt{3}}{2}g) + hS_\beta (C_\alpha + \sqrt{3}S_\alpha) p_z. \quad (13)$$

C. Euclidian Distance Method

The coordinate frame $\{B\}$ with respect to coordinate frame $\{A\}$ can also be described using homogeneous transformation matrix $^{A}T_B$ [16]:

$$^{A}T_B = \begin{bmatrix} u_x & v_x & w_x & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad [14]$$

$$\begin{bmatrix} C_\alpha C_\beta C_\gamma - S_\alpha S_\gamma & -C_\alpha C_\beta S_\gamma - S_\alpha C_\gamma & C_\alpha S_\beta & p_x \\ S_\alpha C_\beta C_\gamma + C_\alpha S_\gamma & -S_\alpha C_\beta S_\gamma + C_\alpha C_\gamma & S_\alpha S_\beta & p_y \\ -S_\beta C_\gamma & S_\beta S_\gamma & C_\beta & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

where: C_x is short for $\cos(x)$ and S_x is short for $\sin(x)$.

The column vector $[p_x\ p_y\ p_z]^T$ denotes the translation from the center point of the base to the center point of the upper platform. The position of the ball joints B_i with respect to the base frame $\{A\}$ can be expressed as:

$$^{A}B_i = {}^{A}T_B \cdot {}^{B}B_i. \quad [15]$$

Eq. (16) defines the inverse kinematics equation by determining the Euclidian distance from $^{A}B_i$ to $^{A}A_i$.

$$d_i = norm(^{A}B_i - {}^{A}A_i) = norm(^{A}T_B \cdot B_i - {}^{A}A_i). \quad [16]$$

For a prescribed position and orientation of the moving platform, the dependent variables are defined by (8) – (10), , Eq. (14) defines the homogeneous transformation matrix and (15) defines the length d_i of the links [11].

III. REAL-TIME EXPERIMENTS

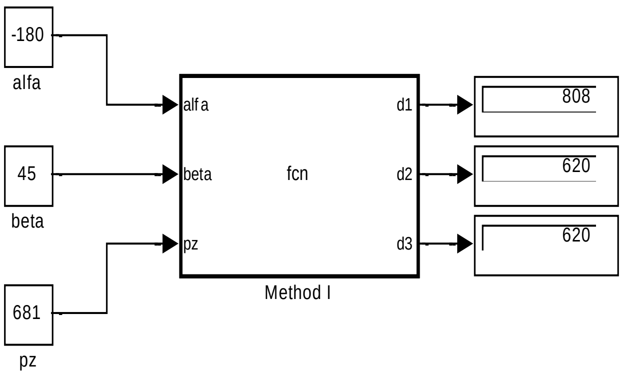

Fig. 4. Method I Simulink model

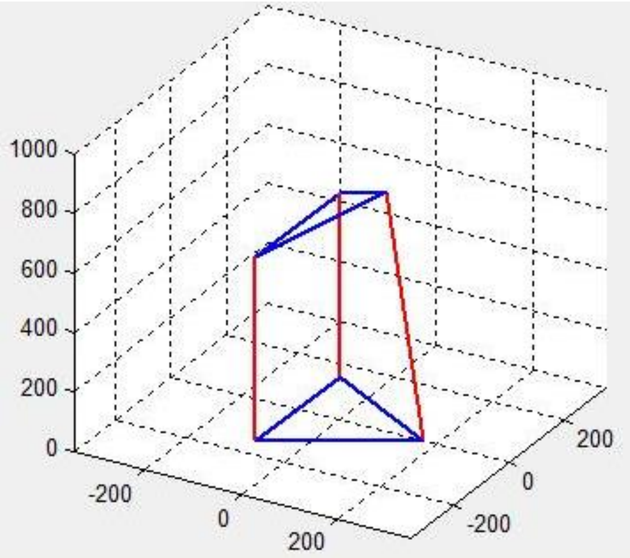

Fig. 5. alfa = -180˚, beta = 45˚, pz = 681 [mm], g = h = 173.205 [mm]

Fig. 4 presents the Simulink model of inverse kinematics Eqs. (11-13). The model was built using Embeded MATLAB Function block, which permit us to write and execute m code in Simulink models. Eq. (16) was implemented in a similar way. Alfa, beta and pz are the inputs of the inverse kinematics functions; first two describes the orientation, while pz the vertical cartesian coordinate of the mobile platform. Outputs of the function are length d_i of the links.

Fig. 5 presents the position and orientation of the mobile platform for prescribed parameters. Using Simulink/Real-Time Workshop based on dSPACE and ControlDesk, both models were compiled and linked to DS1104.

A GUI interface was created to obtain the simulation results (Fig. 6). Table 1 presents the obtained results. It is clear that the second method is more time consuming than the first one. Actually, the first method is 3.6 faster than the second one. In a real-time application this can be a good improvement regarding the computational time of the tasks.

TABLE 1: SIMULATION RESULTS

Method	***Time[us]***
Method I	*2.92*
Method II	*10.64*

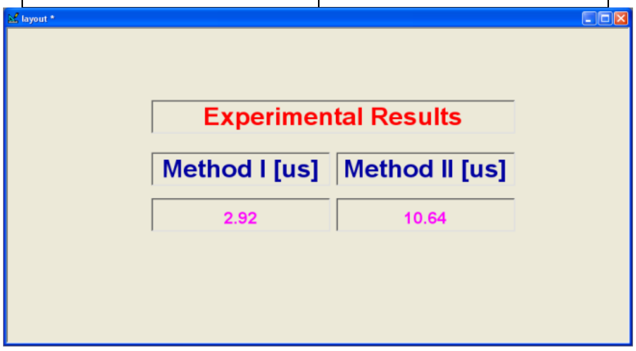

Fig. 6. ControlDesk GUI Simulation Results

IV. CONCLUSION

The paper presented a dSPACE 1104 based real-time experiment of two inverse kinematics methods for a 3-DOF medical parallel robot with R-P-S (revolute-prismatic-spherical) joint structure. The inverse kinematics methods were presented and Eqs. that defines the length d_i of the links were implemented in MATLAB/Simulink. Using Simulink/Real-Time Workshop based on dSPACE, the models were compiled and linked to DS1104 controller board. A GUI interface was realized with ControlDesk to obtain the execution times of the algorithms. In a real-world applications it is very important for the robots to compute certain tasks in a given sample time. Experimental results have revealed that the first method is actually 3.6 faster than the second one which makes it suitable for real-world applications.

REFERENCES

[1] Xu, W.L.; Pap, J.-S.; Bronlund, J.; , "Design of a Biologically Inspired Parallel Robot for Foods Chewing," *Industrial Electronics, IEEE Transactions on* , vol.55, no.2, pp.832-841, Feb. 2008.

[2] Stan, S.-D.; Manic, M.; Maties, V.; Balan, R.; , "Evolutionary approach to optimal design of 3 DOF translation exoskeleton and medical parallel robots," *Human System Interactions, 2008 Conference on* , vol., no., pp.720-725, 25-27 May 2008.

[3] Yangmin Li; Qingsong Xu; , "Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation," *Mechatronics, IEEE/ASME Transactions on* , vol.12, no.3, pp.265-273, June 2007.

[4] Szep, C.; Stan, S.-D.; Csibi, V.; Manic, M.; Balan, R.; , "Kinematics, workspace, design and accuracy analysis of RPRPR medical parallel robot," Human System Interactions, 2009. HSI '09. 2nd Conference on , vol., no., pp.75-80, 21-23 May 2009.

[5] Verdes, D.; Stan, S.-D.; Manic, M.; Balan, R.; Maties, V.; , "Kinematics analysis, workspace, design and control of 3-RPS and TRIGLIDE medical parallel robots," Human System Interactions, 2009. HSI '09. 2nd Conference on , vol., no., pp.103-108, 21-23 May 2009.

[6] Szep, C.; Stan, S.-D.; Csibi, V.; Manic, M.; Balan, R.; , "New approach for accuracy and kinematics analysis of 2 DOF medical parallel robot," Digital Ecosystems and Technologies, 2009. DEST '09. 3rd IEEE International Conference on , vol., no., pp.524-529, 1-3 June 2009.

[7] J-P. Merlet. *Parallel robots, 2nd Edition*. Springer, Heidelberg, 2005.

[8] Lee, K.-M.; Shah, D.K.; , "Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator," *Robotics and Automation, IEEE Journal of* , vol.4, no.3, pp.354-360, Jun 1988.

[9] Lee, K.-M.; Shah, D.K.; , "Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator," *Robotics and Automation, IEEE Journal of* , vol.4, no.3, pp.361-367, Jun 1988.

[10] Zhang M.D. and Song S.M. Study of three-degree-of-freedom parallelplatforms for reactional compensation. In *ISRAM*, pages 373–378, Hawai, August, 14-18, 1994.

[11] R. Amann; F. Geiger; , “Simulation and Code Generation for a Parallel Kinematic Manipulator with three Degrees of Freedom,” *Electronics and electrical engineering, Automation, Robotics* , 2008, No. 1(81), ISSN 1392-1215.

[12] dSPACE, DS1 104 R&D Controller Board Installation and Configuration Guide, 2003.

[13] dSPACE, Real-time interface (RTI and RTI-MP) implementation guide (For Release 3.5), 2003.

[14] dSPACE, ControlDesk Experiment Guide (For Release 3.5), 2003.

[15] Lung-Wen Tsai, Robot Analysis: The mechanics of serial and parallel manipulators, John Wiley & Sons, Inc., 1999.

[16] J. J. Craig, Introduction to Robotics, Mechanics and Control. Reading, MA: Addison-Wesley, 1985.