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

Ciprian-Radu Rad¹, Milos Manic², Member, IEEE, Radu Balan¹, and Sergiu-Dan Stan¹, Member, IEEE
¹Technical University of Cluj-Napoca, Romania,
²University 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 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 integrates 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. 1. 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 DS1104 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 analyzed in Section III. Finally, 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 Grüber’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 \( \mathbf{b}_i \) 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 \( B_i \) (equilateral triangles) are given by:

\[
\mathbf{a}_1 = [g, 0, 0]^T, \tag{2}
\]
\[
\mathbf{a}_2 = [-\frac{1}{2} g, \frac{\sqrt{3}}{2}, 0]^T, \tag{3}
\]
\[
\mathbf{a}_3 = [-\frac{1}{2} g, -\frac{\sqrt{3}}{2}, 0]^T. \tag{4}
\]
\[ \mathbf{b}_2 = \left[ -\frac{1}{2} h, \frac{\sqrt{3}}{2} h, 0 \right]^T, \]  
(6)

\[ \mathbf{b}_3 = \left[ -\frac{1}{2} h, -\frac{\sqrt{3}}{2} h, 0 \right]^T. \]
(7)

The inverse kinematics problem is 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 \( \alpha \) and the nutation angle \( \beta \). The cartesian translatory freedom is in the \( Z \) direction. The spin angle \( \gamma \) 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:

\[ \gamma = -\alpha, \]  
(8)

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

\[ p_y = \frac{1}{2} h(1 - \cos(\beta)) \sin(2\alpha). \]
(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].

\[ d_1 = g^2 + h^2 + p_1^2 + p_i^2 + z^2 + 2g p_x + 2h(C_\alpha C_\beta + S_\alpha S_\beta)(p_3 - g) + \ldots + h(C_\alpha - 1)S_\alpha p_y - 2hS_\beta C_\alpha p_z, \]
(11)

\[ d_2 = g^2 + h^2 + p_1^2 + p_i^2 + z^2 + g p_x - \frac{3}{2} g p_y - h(C_\alpha C_\beta + S_\alpha S_\beta)(p_3 - g) + \ldots - \sqrt{3} C_\alpha S_\alpha (C_\beta - 1)(p_3 - \frac{1}{2} g) - h(S_\alpha C_\beta (C_\beta - 1) - \sqrt{3} (S_\beta C_\alpha + C_\beta)) + \ldots + (p_3 - \frac{\sqrt{3}}{2} g) h S_\beta (C_\alpha - \sqrt{3} S_\alpha) p_z, \]  
(12)

\[ d_3 = g^2 + h^2 + p_1^2 + p_i^2 + z^2 + g p_x + \frac{3}{2} g p_y - h(C_\alpha C_\beta + S_\alpha S_\beta)(p_3 - g) + \ldots + \sqrt{3} C_\alpha S_\alpha (C_\beta - 1)(p_3 - \frac{1}{2} g) - h(S_\alpha C_\beta (C_\beta - 1) + \sqrt{3} (S_\beta C_\alpha + C_\beta)) + \ldots + (p_3 + \frac{\sqrt{3}}{2} g) h S_\beta (C_\alpha + \sqrt{3} S_\alpha) p_z. \]
(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]:

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

(14)

where: \( C_\gamma \) is short for \( \cos(\gamma) \) and \( S_\gamma \) is short for \( \sin(\gamma) \).

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:

\[ ^A B_i = ^A T_B \cdot ^B B_i. \]
(15)

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

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

For a prescribed position and orientation of the moving platform, the dependent variables are defined by (8) – (10), and 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](image-url)
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