MODELLING AND CONTROL OF A SPHERICAL ROBOTIC DEVICE

Luca Carbonari*, Giacomo Palmieri**, Matteo-Claudio Palpacelli*, Donatello Tina*

*Dipartimento di Ingegneria Industriale e Scienze Matematiche, Università Politecnica delle Marche, Ancona (AN), Italy (e-mail: {l.carbonari, m.palpacelli, d.tina}@univpm.it).
**Università degli Studi e-Campus, Novedrate (CO), Italy (e-mail: giacomo.palmieri@uniecampus.it)

ABSTRACT
The researchers at the Polytechnic University of Marche developed a spherical parallel manipulator designed for the orientation of parts or tools: the paper presents the first experimental results on such innovative machine. A model of the prototype robot has been realised by means of commercial multibody software then both open-loop and closed loop dynamics has been studied. The relative simplicity of machine kinematics allowed also to experiment the use of controllers with compensation of gravitational terms.

KEY WORDS
Robot control, Control system analysis, Mechanisms, Robot dynamics, Model-based control, PID control.

1. Introduction
The issues of modularity and reconfigurability are assuming more and more importance in present production systems [1,2] and also in the field of robotics researchers are recently trying to exploit such concepts to enhance performances while still keeping costs under control [3,4]. Parallel kinematics machines are characterised by a modular mechanical architecture, but their use has been limited in past years due to some well-known drawbacks, among which the limited workspace and the complex kinematics, at least for 6-axes machines. Such issues are even more important in the case of mini-or micro-robotics, where the problems of small-scale realisations urge for new concepts and solutions, especially for assembly applications [5-7]. The researchers at the Laboratory of Robotics of the Polytechnic University of Marche developed two 3 axes robots that shall cooperate in order to perform full mobility assembly tasks [8]: the kinematics of both machines is based upon the same 3-CPU parallel topology but the joints are differently assembled so as to obtain a Translating Parallel Machine (TPM) with one mechanism and a Spherical Parallel Machine (SPM) with the other. The two robots are presently available at the prototypical stage and the present article reports some experiments on the motion control of the orienting device, which was previously tested with impedance control algorithms [9,10].

2. Robot Architecture and Kinematics
2.1 Mechanical Architecture
The detailed description of machine kinematics and prototype design has been provided already by Callegari et al. [11], therefore the present section only outlines the most relevant aspects. The spherical parallel machine under study is made of three identical serial chains connecting the moving platform to the fixed base, as shown in Figure 1; each leg is composed by two links: the first one is connected to the frame by a cylindrical joint (C) while the second link is connected to the first one by a prismatic joint (P) and to the end-effector by a universal joint (U); for this reason its mechanical architecture is commonly called 3-CPU. A few manufacturing conditions must be fulfilled in order to constraint the end-effector to a spherical motion:

- the axes of the cylindrical joints \(a_i, i=1,2,3\) are aligned along the \(x, y, z\) axes of the base frame and intersect at the centre \(O\) of the spherical motion;
- the axis \(b_i\) of each prismatic pair is perpendicular to the axis of the respective cylindrical joint \(a_i\);
- the first axis of each universal joint is perpendicular to the plane of the corresponding leg (plane identified by the axes \(a_i\) and \(b_i\));
- the second axis of the 3 universal joints (respectively for the leg 1, 2 and 3) are aligned along the \(y_1, z_1, x_1\) axes of a local frame centred in \(P\) (coincident with \(O\)) and attached to the mobile platform.

For a successful operation of the mechanism, one mounting condition must be satisfied too: assembly should be operated in such a way that the two frames \(O(x_0, y_0,z_0)\) and \(P(x_1,y_1,z_1)\) come to coincide.
2.2 Machine Kinematics

The platform is actuated by driving the strokes of the 3 cylindrical joints, gathered into the vector \( \mathbf{a} \):

\[
\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}
\] (1)

The position kinematics of the robot expresses the relation between the orientation of the mobile platform and the displacements of the actuators; the attitude of the machine in space is fully provided by the rotation matrix \( \mathbf{R} \), that can also be conveniently expressed as a composition of elemental rotations. In the development of robot’s kinematics, the following Cardan angles set is used:

\[
\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma) = \\
\begin{bmatrix}
c\beta c\gamma + s\alpha s\beta & -s\beta c\gamma + c\alpha s\beta & s\gamma s\beta c\alpha - c\gamma s\alpha c\beta \\
-c\beta s\gamma + c\alpha s\beta & s\beta s\gamma + c\alpha s\beta & c\gamma s\beta c\alpha + s\gamma s\alpha c\beta \\
-s\alpha c\gamma & -s\beta c\gamma & c\gamma s\beta c\alpha + s\gamma s\alpha c\beta
\end{bmatrix}
\] (2)

where \( c\alpha \) is a shorthand of \( \cos(\alpha) \), \( s\alpha \) for \( \sin(\alpha) \) and so on.

The position kinematics of the robot is simply expressed by:

\[
\begin{align*}
\mathbf{r}_{12} &= -c\beta s\gamma = \frac{e-a_1}{d} \\
\mathbf{r}_{23} &= -s\alpha c\gamma = \frac{e-a_2}{d} \\
\mathbf{r}_{31} &= -c\alpha c\beta s\gamma + s\gamma s\beta c\alpha = \frac{e-a_3}{d}
\end{align*}
\] (3)

where \( r_{ij} \) is the element at the \( i \)th row and \( j \)th column of rotation matrix \( \mathbf{R} \). The solution of the direct position kinematics (DPK) problem requires the computation of the rotation matrix \( \mathbf{R} \) as a function of internal coordinates \( a \); a maximum number of 8 different configurations can be worked out, however a single feasible solution is found when the actual mobility of the joints is taken into consideration.

Of course, inverse position kinematic (IPK) problem admits just one solution and it is easily solved by working out joint displacements \( \mathbf{a} \) in (3).

Turning to differential kinematics, the expression of the analytic Jacobian \( \mathbf{J}_A \) is immediately obtained as a function of the Cardan angles and their rates:

\[
\begin{bmatrix}
\dot{a}_1 \\
\dot{a}_2 \\
\dot{a}_3
\end{bmatrix} = \mathbf{J}_A \\
\begin{bmatrix}
\dot{\alpha} \\
\dot{\beta} \\
\dot{\gamma}
\end{bmatrix}
\] (4)

\[
\mathbf{J}_A = d \\
\begin{bmatrix}
0 & -s\beta s\gamma & c\beta c\gamma \\
-sa s\beta c\gamma - c\beta s\gamma & c\gamma s\beta c\alpha + s\gamma s\alpha c\beta & -c\gamma s\beta c\alpha - s\gamma s\alpha c\beta \\
c\gamma s\beta c\alpha + s\gamma s\alpha c\beta & -c\gamma s\beta c\alpha - s\gamma s\alpha c\beta & 0
\end{bmatrix}
\] (5)

By taking into account the linear mapping \( \mathbf{T} \) between the derivatives of the Cardan angles and the angular velocity \( \boldsymbol{\omega} \), reported here below

\[
\begin{bmatrix}
\dot{\alpha} \\
\dot{\beta} \\
\dot{\gamma}
\end{bmatrix} = \mathbf{T} \\
\begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\] (6)

the geometric Jacobian \( \mathbf{J}_G \) is easily obtained too:

\[
\begin{bmatrix}
\dot{a}_1 \\
\dot{a}_2 \\
\dot{a}_3
\end{bmatrix} = \mathbf{J}_A \mathbf{T}^{-1} \\
\begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\] (7)

with:

\[
\mathbf{J}_G = d \\
\begin{bmatrix}
0 & -c\alpha c\beta s\gamma + s\gamma s\alpha c\beta & -s\alpha c\gamma \\
-c\alpha c\beta c\gamma + s\gamma s\alpha c\beta & c\alpha c\beta s\gamma + c\gamma s\alpha c\beta & -c\gamma s\beta c\alpha - s\gamma s\alpha c\beta \\
-s\alpha c\gamma & -s\alpha c\beta s\gamma - c\gamma s\beta c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha c\beta
\end{bmatrix}
\] (8)

2.3 User Frames

In order to better define the assigned tasks and visualize the obtained results, it is necessary to choose a different set of reference frames, as shown in Figure 2. The fixed frame \( O^* (x_0^*, y_0^*, z_0^*) \) is defined as follows:

- the origin is located at the centre of the moving platform when it assumes its initial configuration
- the \( z_0^* \) axis is directed as the vector \( \mathbf{g} \) of gravity acceleration
- the \( x_0^* \) axis lies on the horizontal plane passing through the reference system origin and on the vertical plane which contains the axis \( a_1 \) of the cylindrical joint of the first leg
• the $y_0^*$ axis is placed according to the right-hand rule.

The mobile frame $P^*(x_1^*,y_1^*,z_1^*)$ is coincident with the fixed frame $O^*(x_0^*,y_0^*,z_0^*)$ when the platform is in its initial configuration. Of course, since the frames are not placed at the centre of the spherical motion, the two origins $O^*$ and $P^*$ will be coincident only in the home configuration.

Once the location of the new frame $O^*$ has been defined by means of the $\mathbf{R}_{O^*O}$ rotation matrix, the orientation of the mobile platform can be described in the new frames by:

$$\mathbf{R}_P = \mathbf{R}_{O^*O} \mathbf{R}_{P^*P}$$

(9)

where it has been used the identity $\mathbf{R}_{O^*O} = \mathbf{R}_{P^*P}$. Of course, having changed the mobile and fixed frames, also the Cardan angles $\varphi_x, \varphi_y, \varphi_z$ that yield the rotation matrix $\mathbf{R}_P$ are different from the previously described set ($\alpha, \beta, \gamma$):

$$\mathbf{R}_P(\varphi_x, \varphi_y, \varphi_z) = \mathbf{R}_{x^*}(\varphi_x) \mathbf{R}_{y^*}(\varphi_y) \mathbf{R}_{z^*}(\varphi_z)$$

(10)

Henceforth these angles are used to describe the orientation of the manipulator and to assign the tasks of the mobile platform; since they are assumed as external coordinates for the computation of the differential kinematics, the analytic and the geometric Jacobians are worked out again as previously described, providing similar but more complex relations.

3. Prototype Design

The design of a first prototype has been developed with the aim of obtaining high dynamic performances in cooperative assembly tasks. Since Jacobian matrix (8) does not depend on geometric parameters (leaving out the scale factor $d$), it was not possible to drive the mechanical design based on the optimization of kinematic properties: therefore heuristic considerations and several computer simulations, based on a multibody model, have allowed to refine the design.

Figure 3a sketches the CAD model of the prototype while in Figure 3b a picture of the machine is presented. The limbs are made of avional in order to join good mechanical properties with a lightweight construction; the mobile platform is made of bronze, therefore allowing the precise machining in a single placement of the 3 journal bearings that have to meet orthogonally in a single point.

<table>
<thead>
<tr>
<th>Geometrical data</th>
<th>Mass data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$ 210 mm</td>
<td>slider 7.15 kg</td>
</tr>
<tr>
<td>$d$ 490 mm</td>
<td>link 1 1.90 kg</td>
</tr>
<tr>
<td>$a_{i_{\text{min}}}$ 319 mm</td>
<td>link 2 2.21 kg</td>
</tr>
<tr>
<td>$a_{i_{\text{max}}}$ 661 mm</td>
<td>platform 11.73 kg</td>
</tr>
<tr>
<td>$b_{i_{\text{min}}}$ 130 mm</td>
<td></td>
</tr>
<tr>
<td>$b_{i_{\text{max}}}$ 210 mm</td>
<td></td>
</tr>
</tbody>
</table>
4. Control Algorithms

The prototype of the robot has been tested by the execution of some simple closed-loop motions. The first set of trials has been driven by a conventional PID controller [12]. The error signal $\tilde{a}$ is computed as difference between actual position of cylindrical joints $a$ and their desired values $a_D$:

$$\tilde{a} = a_D - a$$

Since planning is programmed in the orientation space by assigning the desired configuration of the robot $\varphi_D$, the corresponding positions of the actuated joints are computed by means of the inverse kinematics. According to the joint resolved PID algorithm, see Figure 4a, the actuation effort of the motors is:

$$f_i = k_{pi} \left[ \tilde{a}_i + \frac{1}{T_i} \int \tilde{a}_i dt + T_i \frac{d\tilde{a}_i}{dt} \right]$$

(12)

In the second set of tests, the effects of gravitational field have been compensated by adding the term:

$$f_g = -J^T \sum_i m_i J_i^T g$$

(13)

where $J_i$ is the analytic Jacobian matrix, $m_i$ is the mass of the $i^{th}$ member, $J_i$ is the Jacobian that links the velocity of the centre of gravity of the $i^{th}$ member to the vector $\tilde{a}$; $g$ is the gravity acceleration. The related control scheme is shown in Figure 4b.

5. Multibody Model

A few test cases have been set up to evaluate in simulation the performances of the two PID controllers (12) and (13). The following figures show system response when the robot started at rest at the home configuration ($\varphi_x = \varphi_y = \varphi_z = 0$) and was required to attain the set point:

$$\varphi_D = \begin{bmatrix} \varphi_{x,D} \\ \varphi_{y,D} \\ \varphi_{z,D} \end{bmatrix} = \begin{bmatrix} 15^\circ \\ 15^\circ \\ 15^\circ \end{bmatrix}$$

$$\varphi_D = 0$$

(14)

Such task is very challenging for machine’s controller because the set point lies close to a singular configuration of the robot and algorithm (13) requires an inversion of the Jacobian matrix. Figure 5 shows the different performances, in simulation, between the two different controllers: C1 and C2 represent respectively the joint resolved PID and the joint resolved PID with gravity compensation [13]. It is noted that the robot is not kept at its home position by means of the brakes but only the motors are used to this aim instead; then the set point has been applied in all trials at the time instant $t=0.5$ s. The orientation trajectories in the task space show the better behaviour of the closed-loop system when it is equipped with the conventional PID algorithm, due to the mentioned proximity to a singular configuration.

The simulations also return useful information for what regards the control effort forces, which are plot in Figure 6: in all cases the application of the set point causes a peak in the required forces, which saturates the actuators.
In the end, it is noted that the tasks space PID with gravity compensation is more sensitive to parameter variation. This is due to the intrinsic characteristics of robot prototype, which has no external sensor and many singular configurations: in this way, all the information about the task space is obtained through the direct kinematics and the robot Jacobian. Small errors in the computation may cause heavy problems to the control system.

6. Experimental Tests Results

Two different set of tests (case studies A and B) have been performed in order to asses and compare the linear PID and the gravity compensation control. The tuning of the controller has been set up through the conventional Ziegler-Nichols rules [14]; the values of the gains that have been found at the end of the tuning work are collected in Table 2.

<table>
<thead>
<tr>
<th>PID parameters</th>
<th>Kp,i</th>
<th>V/mm</th>
<th>Proportional gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>2</td>
<td>ms</td>
<td>Integration time</td>
</tr>
<tr>
<td>TD</td>
<td>0.5</td>
<td>ms</td>
<td>Derivative time</td>
</tr>
</tbody>
</table>

**Case study A:** starting from the home configuration the following set point $\mathbf{\Phi}_D, \dot{\mathbf{\Phi}}_D$ has been imposed to the wrist:

$$\mathbf{\Phi}_D = \begin{bmatrix} \phi_{x,D} \\ \phi_{y,D} \\ \phi_{z,D} \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix} [\text{rad}] \quad (\Phi_D = 0) \quad (15)$$

The requested configuration represents a tilt of about $30^\circ$ of the platform around an axis parallel to its plane and corresponds to the following motor strokes:

$$\mathbf{a}_D = \begin{bmatrix} a_{1,D} \\ a_{2,D} \\ a_{3,D} \end{bmatrix} = \begin{bmatrix} 457.5 \\ 567.9 \\ 457.5 \end{bmatrix} [\text{mm}] \quad (\dot{a}_D = 0) \quad (16)$$

Figure 7 presents a superposition between the two different controllers C1 and C2 during this test case. Data of Figure 7 are the averages of different tests; the time history of the orientation angles of the platform are plotted. Fig. 8 focuses on the time history of joint space errors: steady state is achieved in about half a second without appreciable oscillations for both controllers; this is mainly due to the damping of the system, which is pretty high in relation with the set PID parameters. Figure 9 shows the forces $F_i$ exerted by the motors in order to reach the set point; such forces are estimated by measuring feed currents $I_i$ of the motors, which relate to forces by means of the torque constant:

$$F_i = K_t I_i$$

(17)
Figure 7. Platform’s trajectory in workspace. Comparison of C1 and C2 controllers.

Figure 8. Joint space errors for both controllers: joint resolved PID (a) and PID with gravity compensation (b).

Figure 9. Actuation forces

Case study B: the set point has been fixed at

$$\varphi_D = \begin{bmatrix} \varphi_{x,D} \\ \varphi_{y,D} \\ \varphi_{z,D} \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} \text{[rad]} \quad (\dot{\varphi}_D = 0) \quad (18)$$

The requested configuration represents a tilt of about 15° of the platform around all the three coordinate axes (executed in the proper order) and corresponds to the following motor strokes:

$$a_D = \begin{bmatrix} a_{1,D} \\ a_{2,D} \\ a_{3,D} \end{bmatrix} = \begin{bmatrix} 460.5 \\ 571.0 \\ 534.9 \end{bmatrix} \text{[mm]} \quad (\dot{a}_D = 0) \quad (19)$$

Figure 10 plots again the orientation of the platform versus time: steady state is reached in a longer time than in the previous case because of larger rotations commanded to the manipulator. The response is free of relevant oscillations; this is particularly appreciable if the set point is, as in this case, near to workspace borders and excessive overshoots and oscillations may damage the robot. Figure 11(a) and Figure 11(b) show once again the joints pace error in the two different controls. Figure 12 shows the control effort represented by forces $F_i$ exerted by the motors.
7. Conclusion

First experimental results on motion control of a prototypal SPM developed at Polytechnic University of Marche have been presented. Control systems have been designed and virtually tested by means of multibody software and model based programming. Two different well-known controllers have been investigated. In particular, the advantages of compensating the gravity terms have been studied: as a matter of fact, such controller provided poorer performances than conventional PID but in Author’s opinion it could be due to calibration errors, that would heavily affect the computation of direct kinematics, which is required with the present sensing equipment.

The simplicity of direct kinematics of this machine (in comparison with the usual complexity of PKMs) will allow to test in future algorithms with loop closures in the task space. These features and the possible use of visual servoing suggest a possible implementation of control schemes based on force control, where machine’s dynamics has to be computed in task-space coordinates, which is rather “natural” for parallel kinematics machines.

References


