HYBRID FORCE/POSITION CONTROL FOR A 3-DOF 1T2R PARALLEL ROBOT: IMPLEMENTATION, SIMULATIONS AND EXPERIMENTS.

José Cazalilla[*], Marina Vallés[*], Ángel Valera[*] , Vicente Mata[†], Miguel Díaz-Rodríguez[**],

[*]Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain
e-mail: {jcazalilla, mvalles, giuprog}@isa.upv.es

[†]Centro de Investigación en Ingeniería Mecánica
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain
e-mails: vmata@mcm.upv.es

[**]Departamento de Tecnología y Diseño, Facultad de Ingeniería
Universidad de los Andes, La Hechicera, 5101 Mérida, Venezuela
e-mails: dmiguel@ula.ve

Abstract. *A robot interacting with the environment requires that the end effector position is tracked and that the forces of contact are kept below certain reference values. For instance, in a rehabilitation session using a robotic device, the contact forces are limited by the allowed strength of the human limbs and their complex-joints. In these cases, a control scheme which considers both position and force control is essential to avoid damage to either the end effector or the object interacting with the robot. This paper therefore develops a real-time force/position control scheme for a 3-DOF parallel robot whose end effector holds a DOF one translation (1T) and two rotations (2R). The implemented hybrid force/position control considers, as a reference, the normal force on the mobile platform, which is measured by means of a load cell installed on the platform. The position control is designed to track the orientations of the robot either in joint or task space using a model-based control scheme with identified parameters. Moreover, the force control is based on a PD action. The control scheme is developed through simulations, before being applied to an actual parallel robot. The findings show that with the implemented controller, the actual robot accomplishes the reference values for the normal force on the mobile platform, while at the same time the platform accurately follows the required angular orientation.*

Keywords: Parallel Manipulator, Robot Control, Force Control, Mechatronics, Dynamics.

1 INTRODUCTION

The inherent capabilities of a parallel robot (PR), compared with its counterpart a serial robot, have made it an interesting research topic in the robotic community. These capabilities can be summed up as follows: high stiffness, high load-carrying footprint ratio, high speed and accuracy. In these robots the end effector is attached to a mobile platform which is connected by several kinematic chains (also known as legs) to a fixed based. The advantages over serial robots stem from the fact that the load on the end effector can be shared by the legs. The similarity with the human body is that when we are unable to carry a load precisely with one arm, we use both arms, thus giving us greater precision and stiffness, although we lose workspace. The forward kinematics, dynamics model and control of PMs are also cumbersome. Given that in certain cases the advantages outweigh the disadvantages, researchers have developed several applications based on PRs, such as motion simulators, tire testing machines, flight simulators and medical applications. Different mechanical architectures of PRs can be found in the following references (Gough and Whitehall, 1962; Steward, 1965; Tsai, 1999; Merlet, 2000).

Seminal papers on PRs have focused on platforms with six degrees of freedom (DOF): three translational (3T) and three rotational (3R), an idea that continues to be developed with new 6-DOF architectures nowadays (Cao et al., 2015). However, many applications require fewer than 6 DOF. In fact, one of the fastest PR architectures is based on 3T1R motion (Pierrot et al. 2009). Furthermore, pick-and-place tasks can be accomplished with three translational DOF (3T), the best example of which is the Delta Robot (Clavel, 1988). In addition, medical applications such as cardiopulmonary resuscitation equipment are accomplished with 3T PRs (Li and Xu, 2007), as are tool heads for manufacturing facilities (Tsai, 1999). Other developments with 3T1R motion can be found in (Zarkandi, 2011). For low-cost applications, designers frequently search for robot architectures with fewer DOF. One such application has 1T2R motion. In this case, the 3-PRS (Carretero, Podhorodeski and Nahon, 2000) and the 3-RPS (Lee and Arjunan, 1991) architectures have been proposed. Note that R, P and S stand for revolution, prismatic and spherical joint at each leg, respectively.

On the other hand, independently of the robot's DOF, the trajectory at the end effector is controlled using position control: point-to-point or tracking control, joint or task space. However, a variety of applications require that the forces applied to the end effector are kept below certain reference values. For example, this is the case with robots involving assembly tasks where one or more parts need to be handled, ensuring proper contact, or industrial operations such as milling or deburring (Afonso, Pires and Estrela, 2007). New developments in parallel robots for medical applications involve interaction with delicate parts of the human body where the forces at the end effector must be applied carefully, e.g. robots for *in vivo* biopsy (Garg et al., 2014). For such applications, an accurate robot force control scheme needs to be developed. However, while different force control approaches have been implemented for serial robots (Zeng and Hemami, 1997), the issue of force/position control of PRs has barely been addressed in the robotics literature (Bellakehal et al., 2011).

In a previous work, the authors developed a low-cost PR with 3-PRS, its end effector having one translation (1T) and two rotations (2R) DOF (Valles et al., 2012). This robot has the advantage that the control architecture is open, thus allowing model-based control schemes to be implemented and tested (Diaz-Rodriguez et al., 2013). One interesting application of 1T2R robots is an ankle rehabilitation device, but in a rehabilitation session using a robotic device the contact forces are limited by the allowed strength of the human limbs and joints undergoing rehabilitation. Therefore, the aim of this paper is to develop a real-time hybrid force/position control so that the normal force on the mobile platform follows a reference signal while simultaneously tracking the orientation of the platform. In order to implement the control scheme, the force is measured by means of a load cell installed on the end effector, with the force controller part based on a PD action. The position control is designed to track the orientations of the robot either in joint or task space using a model based control scheme (Rosillo et al., 2011), using an identified model (Farhat, 2008). Simulations conducted in a Matlab/Simulink environment have made it possible to develop the control scheme which is then applied to the actual PR.

The paper is organized as follows: Section 2 briefly describes the test bed PR and develops the kinematic and dynamic models. Section 3 discusses the different control schemes using a simulated robot. Section 4 shows the results of the force/position control when it is applied to the actual prototype, and Section 4 presents the conclusions.

2 THE TEST BED ROBOT

The choice of the test bed robot architecture was determined by the need to develop a low-cost robot which is capable of generating angular rotation on two axes, as well as a prismatic movement. Two different architectures were considered: 3-RPS and 3-PRS. The second was chosen after comparing the advantages and disadvantages of each proposal. For example, one advantage of the PRS architecture is that the actuators are located on the fixed base, while in the RPS architecture the actuators move with revolute joints.

Fig. 1 shows the actual prototype. The physical system consists of three legs connecting the moving platform to the base. Each leg consists of a direct-drive ball screw (prismatic joints) and a coupler, as well as the motor.

Figure 1: The test bed low-cost Parallel Robot

The motors in each leg are brushless DC servomotors equipped with power amplifiers. The actuators are Aerotech BMS465 AH brushless servomotors. Aerotech BA10 power amplifiers operate the motors. The control system was developed on an industrial PC, which has several advantages: the first is that it is a completely open system, providing a powerful platform in which to program high-level tasks. Therefore, applications such as trajectory generation or control strategies based on external sensing (machine vision and force sensors, etc.) can be developed. The second significant advantage of the control system is its cost, given that the software architecture is based on free and open software.

2.1 Kinematic model

Robot force or position control algorithms involve establishing the direct and inverse kinematic models. The inverse kinematic model is used to obtain the motion at the actuated joints corresponding to the motions required at the end effector. On the other hand, since the actuated joints are usually the variables that can be measured through incremental encoders, the direct kinematic model is needed to determine the actual motion at the end effector. In order to develop both kinematic models, nine joint generalized coordinates are used to model the robot (q_i, where $i=1...9$). Moreover, the end-effector motion is modeled though three Cartesian coordinates (x,y,z) and three Euler angles (α,β,γ). Fig. 2 shows a kinematic diagram of the robot.

Figure 2: Kinematic diagram of the 3-PRS Parallel Manipulator, type of joints and generalized coordinates.

The modified Denavit-Hartenberg (D-H) convention is used for modeling purpose. The robot is cut open at the spherical joints shown in Fig. 2 as $P_2$and P_3, thus obtaining three legs. One leg consists of three links: a sliding rod, a coupler and the platform (coordinates $q_1..q_5$), and the two other legs consist of two links: a sliding rod and a coupler (coordinates $q_6..q_7$ and $q_8..q_9$). Numbers 1, 4 and 6 in the figure apply to the sliding rod which is attached to the coupler (numbers 2, 5 and 7) by a revolute joint (R), and to the base by an actuated prismatic joint (P). The active coordinates q_1, q_6 and q_8 are associated with the actuated prismatic joints, while the passive coordinates q_2, q_7 and q_9 are associated with the revolute joints. Coordinates q_3, q_4 and q_5 are associated with the spherical joint located at P_1. Note that the spherical joint is modeled, according to the D-H parameters, as three orthogonal axes intersecting at the center of an ideal sphere (representing the spherical joint).

In the forward position problem, the position of the actuators is known (q_1, q_6 and q_8 are known) but the remaining coordinates need to be found. Due to the fact that the length between the locations of the spherical joints P_i and P_{i+1} is constant and equal to l_m (rigid body assumption), the following equation can be written as follows:

$$\left\|\vec{r}_{A_iB_i} + \vec{r}_{B_iP_i} - \left(\vec{r}_{A_iA_{i+1}} + \vec{r}_{A_{i+1}B_{i+1}} + \vec{r}_{B_{i+1}P_{i+1}}\right)\right\| - l_m^2 = 0 \tag{1}$$

with i=1,2,3 and when i=3 then $i+1$=1. $\vec{r}_{AB}$ denotes the vector from point A to B.

Equation (1) constitutes a set of three nonlinear equations relating the known variables q_1, q_6 and q_8 to the unknown (q_2, q_7 and q_9). The Newton-Raphson (N-R) numerical method is chosen to solve this nonlinear problem. The method converges rather quickly when the initial guess is close to the desired solution (Jalón and Bayo, 1994). After finding coordinates q_2, q_7 and q_9, the position of each point P_i can be found. Thus, the coordinates associated with the spherical joints P_1 can be obtained in a straightforward manner, making it possible to find angles β, γ at the end effector. Since the location of the spherical joints forms an equilateral triangle, the heave (z) of the end effector is determined by the mean values of the points $P_1..P_3$.

On the other hand, the inverse kinematics problem consists of finding the movement of the linear actuators from the roll (γ), pitch (β) and height (z) of the platform. Following a similar procedure to that presented for the case of an RPS parallel robot (Tsai, 1999,), the positions of the actuators for the PRS parallel robot can be obtained with the following expression:

$$q_1 = x^2 + y^2 + z^2 + 2h\left(xu_x + yu_y + zu_z\right) - 2gx - 2ghu_x + g^2 + h^2 \tag{2}$$

$$\begin{aligned} q_6 = {} & x^2 + y^2 + z^2 - h\left(zu_x + yu_y + zu_z\right) + \sqrt{3}h\left(xv_x + yv_y + zv_z\right) \\ & + g\left(x - \sqrt{3}y\right) + gh\left(u_x - \sqrt{3}u_y\right)/2 + gh\left(v_x - \sqrt{3}v_y\right)/2 + g^2 + h^2 \end{aligned} \tag{3}$$

$$\begin{aligned} q_8 = {} & x^2 + y^2 + z^2 - h\left(xu_x + yu_y + zu_z\right) - \sqrt{3}h\left(xv_x + yv_y + zv_z\right) \\ & + g\left(x - \sqrt{3}y\right) - gh\left(u_x - \sqrt{3}u_y\right)/2 + gh\left(v_x - \sqrt{3}v_y\right)/2 + g^2 + h^2 \end{aligned} \tag{4}$$

In equations (2-4), $h = l_m / \sqrt{3}$, $g = l_b / \sqrt{3}$, $x = -hu_y$, $y = -h(u_x - v_y)$, l_b is the distance between $A_i A_{i+1}$, and *u* and *v* can be obtained from the rotation matrix of the moving platform. That is:

$$ {}^{0}R_p = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} = \begin{bmatrix} cos(\alpha)cos(\beta) & cos(\alpha)sin(\beta)sin(\gamma)-sin(\alpha)cos(\gamma) & cos(\alpha)sin(\beta)cos(\gamma)-sin(\alpha)sin(\gamma) \\ sin(\alpha)cos(\beta) & sin(\alpha)sin(\beta)sin(\gamma)+cos(\alpha)cos(\gamma) & sin(\alpha)sin(\beta)cos(\gamma)-cos(\alpha)sin(\gamma) \\ -sin(\beta) & sin(\gamma)cos(\beta) & cos(\gamma)cos(\beta) \end{bmatrix} \quad (5) $$

2.2 Jacobian matrix

The kinematic model of a robot seeks relationships between end-effector coordinates and the actuated joint position. The relationship between the velocities of the joint coordinates and the position and orientation of the robot is obtained through the Jacobian matrix. Furthermore, transposing this matrix provides the relationship between the external forces applied to the end effector (often called the wrench) and the actuated joint forces. Thus, the matrix is useful for both the velocity and the force problem. In order to obtain the Jacobian, the following vector expression can be obtained from Fig. 2:

$$ \vec{r}_{OP_c} + \vec{r}_{P_cP_i} = \vec{r}_{OA_i} + q_1 \cdot \vec{r}_{1i} + l_r \cdot \vec{r}_{2i} \quad (6) $$

where $\vec{r}_{1,i}$ and $\vec{r}_{2,i}$ are unit vectors in the direction of A_iB_i and B_iP_i, respectively, l_r is the length of the coupler, $\vec{r}_{OP_c}$ is the position vector from the global coordinate system to the end-effector location, and $\vec{r}_{P_cP_i}$ is the position vector between the end effector and the spherical joint P_i. Finally, $\vec{r}_{OA_i}$ defines the position vector between the intersection of the line defining the prismatic joint motion with the base plane and the global coordinate system.

Deriving equation (6) and multiplying both sides of the equation by $\vec{r}_{2,i}$:

$$ \vec{r}_{2,i} \cdot \vec{v}_{Pc} + \vec{r}_{2i} \cdot \left(\vec{\omega}_{Pc} \times \vec{r}_{P_cP_i} \right) = \vec{r}_{2,i} \cdot \vec{r}_{1,i}\ \dot{q}_i \quad (7) $$

where $\vec{v}_{Pc} = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$ contains the components of the linear velocity at the end effector and $\vec{\omega}_{Pc} = [\omega_x \quad \omega_y \quad \omega_z]^T$ contains the components of the angular velocity.

Equation (7) can be applied to each leg. Defining $\vec{b}_i = \vec{r}_{P_cP_i}$ the following equations hold:

$$ \begin{bmatrix} \vec{r}_{21} \cdot \vec{r}_{11} & 0 & 0 \\ 0 & \vec{r}_{22} \cdot \vec{r}_{12} & 0 \\ 0 & 0 & \vec{r}_{23} \cdot \vec{r}_{13} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_6 \\ \dot{q}_8 \end{bmatrix} = \begin{bmatrix} \vec{r}_{21}^{\,t} & (\vec{b}_1 \times \vec{r}_{21})^t \\ \vec{r}_{22}^{\,t} & (\vec{b}_2 \times \vec{r}_{22})^t \\ \vec{r}_{23}^{\,t} & (\vec{b}_3 \times \vec{r}_{23})^t \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (8) $$

We can therefore verify that:

$$J_q \cdot \dot{\vec{q}} = J_x \cdot \dot{\vec{x}}_{Pc} \tag{9}$$

However, it is important to bear in mind that not all the variables of the $\dot{\vec{x}}_{Pc}$ vector are independent given that only three of the coordinates at the end effector are independent. Consequently, the relationship between the end-effector coordinates has to be determined. The revolute joints, at points B_i, impose the constraint that each leg moves on a plane whose normal vector corresponds to the axis of the revolute joint. Taking this into account, the following set of equations can therefore be written:

$$x = -\frac{l_m}{\sqrt{3}} sin(\alpha) cos(\beta) \tag{10}$$

$$y = -\frac{2l_m}{\sqrt{3}} \left(cos(\alpha) cos(\beta) - sin(\alpha) sin(\beta) sin(\gamma) - cos(\alpha) cos(\gamma) \right) \tag{11}$$

$$tan(\alpha) = \frac{sin(\gamma) cos(\beta)}{cos(\gamma) + cos(\beta)} \tag{12}$$

The partial derivatives of equations (10)-(12) provide the 6 x 3 Jacobian matrix relating the dependent end-effector coordinates to the independent ones. That is:

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{bmatrix}^T = J_r \cdot \begin{bmatrix} \dot{z} & \dot{\beta} & \dot{\gamma} \end{bmatrix} \tag{13}$$

where:

$$J_r = \begin{bmatrix} \frac{\delta x}{\delta z} & \frac{\delta x}{\delta \gamma} & \frac{\delta x}{\delta \beta} \\ \frac{\delta y}{\delta z} & \frac{\delta y}{\delta \gamma} & \frac{\delta y}{\delta \beta} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{\delta \alpha}{\delta z} & \frac{\delta \alpha}{\delta \gamma} & \frac{\delta \alpha}{\delta \beta} \end{bmatrix} \tag{14}$$

The above equations provide the relationship, including the change in platform angle over time. Nevertheless, the Jacobian matrix needed by the wrench has to be written with respect to the angular velocity at the end effector. Based on the fact that the angular velocity is related to the change in the angles over time, this matrix can be obtained by the following equations:

$$\begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T = \begin{bmatrix} cos(\alpha) cos(\beta) & -sin(\alpha) & 0 \\ sin(\alpha) cos(\beta) & cos(\alpha) & 0 \\ -sin(\beta) & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{bmatrix}^T \tag{15}$$

Thus, from equations (9), (13) and (15), the following relationship can be written as follows:

$$\dot{\vec{q}} = J_q^{-1} \cdot J_x \cdot J_r \cdot \begin{bmatrix} \dot{z} & \omega_x & \omega_y \end{bmatrix}^T = J_p \cdot \begin{bmatrix} \dot{z} & \omega_x & \omega_y \end{bmatrix}^T \tag{16}$$

where J_p is the Jacobian matrix relating the coordinates of the end effector to the actuated joint space. The equation relating the wrench at the end effector with the actuation forces can be written as follows:

$$\vec{\tau} = {J_p}^T \cdot \begin{bmatrix} F_z & N_x & N_y \end{bmatrix}^T \tag{17}$$

Equation (17) can be used to find the required actuation forces caused by the external force applied to the end effector, which is useful for developing force control.

2.3 Dynamic model

In order to implement dynamic controllers, the equation of motion of the parallel robot can be described as follows:

$$M\left(\vec{q},\vec{\Phi}\right)\cdot \ddot{\vec{q}} + \vec{C}\left(\vec{q},\dot{\vec{q}},\vec{\Phi}\right)\cdot \dot{\vec{q}} + \vec{G}\left(\vec{q},\vec{\Phi}\right) = \vec{\tau} \tag{18}$$

From equation (18), it can be seen that the system mass matrix *M*, the vectors corresponding to the centrifugal and Coriolis forces $\vec{C}$, and the gravitational forces $\vec{G}$ depend on the dynamic parameters Φ and the external generalized forces $\vec{\tau}$. It is worth noting that (18) is valid only when the system is modeled by a set of independent generalized coordinates. In this paper, a coordinate partitioning procedure has been considered to model the system using a set of independent generalized coordinates. The actuated joint coordinates are the set chosen as the independent coordinates $\vec{q} = \begin{bmatrix} q_1 & q_6 & q_8 \end{bmatrix}^T$.

In order to identify the dynamic parameters, the model needs to be written in linear parameters (Díaz-Rodriguez et al., 2010):

$$\mathrm{K}\left(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\right)\cdot \vec{\Phi} = \vec{\tau} \tag{19}$$

Equation (19) represents a set of three linear equations, thus $\mathrm{K}\left(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\right)$ is a 3 x 25 matrix which is a function of the positions, velocities and accelerations of the generalized coordinates. In addition, $\overrightarrow{\Phi}$ is the 25 x 1 vector of dynamic parameters. As can be seen, the systems of equation hold a number of equation which is less than the unknown parameters. Thus, in order to identify the dynamic parameters, equation (19) has to be applied to a set of several robot states so that an overdetermined system is built up, which can be solved using Least-Square Techniques. The states of the robot which are needed to build the overdetermined system are obtained by measuring the position and forces of the robot when it follows a prescribed trajectory. The trajectory is designed so that all the parameters in the model contribute to the actuation torques (Díaz-Rodriguez et al., 2010). However, even with a well-designed trajectory, some of the dynamic parameters are irrelevant in terms of their contribution to the actuation torques. In this paper, only those relevant parameters which contribute to the dynamics model are identified and used for model-based control. This approach was used in the author's previous work, of which the procedure followed and the dynamics parameters can be seen in (Díaz-Rodriguez et al., 2013).

The gravitational term $\vec{G}$ is obtained, taking into account that it depends only on the generalized coordinates. Thus, by zeroing the generalized velocities and accelerations, the following expression is obtained:

$$K\left(\vec{q},\vec{\dot{q}}=0,\vec{\ddot{q}}=0\right)\cdot\vec{\Phi}=\vec{G}\left(\vec{q}\right) \tag{20}$$

The columns in the system mass matrix can be determined as follows:

$$K\left(\vec{q},\vec{\dot{q}}=0,\vec{e}_i\right)\cdot\vec{\Phi}-\vec{G}\left(\vec{q}\right)=M_i\left(\vec{q}\right) \tag{21}$$

In the last equation, $\mathbf{M}_i\left(\vec{q}\right)$ is the *i*-th column of the mass matrix and $\vec{e}_i=\begin{bmatrix}0 & \dots & 1 & \dots & 0\end{bmatrix}^T$ a column vector with 1 in the *i*-th position.

Finally, by once again zeroing the generalized accelerations in equation (19), the centrifugal and Coriolis terms (which depend on the generalized coordinates and velocities) can be obtained.

3 DEVELOPMENT OF THE SIMULATED PARALLEL ROBOT

3.1 Position joint space control

After the dynamic problem has been solved and the dynamic parameters have been obtained and validated through parameter identification of the parallel robot, the real-time control can be addressed. In this work, various joint space control strategies based on inverse dynamics have been implemented. This type of control is discussed in more detail in (Craig, 1989; Spong and Vidyasagar, 1989; Yoshikawa, 1990; Canudas, Siciliano and Basting, 1996). These strategies are potentially very useful given that they reduce the nonlinear control problem to the control problem of a linear system, for which many tools are available. Assuming the dynamic model as:

$$\vec{x}^{(n)}=f(\vec{x})+b(\vec{x})\cdot\vec{u} \tag{22}$$

where $f(\vec{x})$ and $b(\vec{x})$ are nonlinear functions and $\vec{u}$ is the control input.

If the control input has the expression:

$$\vec{u}=\frac{1}{b(\vec{x})}\left[\vec{a}-f(\vec{x})\right] \tag{23}$$

the nonlinearities will be cancelled, and the simple input-output relation will be obtained:

$$\vec{x}^{(n)}=\vec{a} \tag{24}$$

where $\vec{a}$ is a new input vector to be defined below.

In the case of robot systems, the dynamic model is expressed by Equation (18). Working with this expression:

$$\ddot{\vec{q}} = M^{-1}(\vec{q})\left(\vec{\tau}_c - \vec{C}(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}} - \vec{G}(\vec{q})\right) \tag{25}$$

the following terms can be defined:

$$f(\vec{x}) \equiv M^{-1}(\vec{q})\left(-\vec{C}(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}} - \vec{G}(\vec{q})\right) \tag{26}$$

$$b(\vec{x}) \equiv M^{-1}(\vec{q}) \tag{27}$$

Using the general expression (23):

$$\vec{\tau}_c = \frac{1}{M^{-1}(\vec{q})}\left[\vec{a} - M^{-1}(\vec{q})\left(-\vec{C}(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}} - \vec{G}(\vec{q})\right)\right] \tag{28}$$

the controllers based on the inverse dynamics could be viewed as particular cases of the following control law:

$$\vec{\tau}_c = M(\vec{q})\cdot\vec{a} + \vec{C}(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}} + \vec{G}(\vec{q}) \tag{29}$$

The inverse dynamics control (29) shows how the nonlinearities, such as Coriolis vector $\vec{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}}$, as well as the gravity term $\vec{G}(\vec{q})$ can be simply compensated by adding these forces to the control input.

Depending on the expression used in $\vec{a}$, different control strategies can be obtained. In this work, a trajectory controller has been implemented. For this controller, the $\vec{a}$ expression is:

$$\vec{a} = \ddot{\vec{q}}_d - K_d\dot{\vec{e}} - K_p\vec{e} \tag{30}$$

where $\vec{e} = \vec{q}$ - $\vec{q}_d$, $\dot{\vec{e}} = \dot{\vec{q}}$ - $\dot{\vec{q}}_d$, K_d, and K_p are diagonal and positive-definite matrices and $\vec{q}_d(t)$, $\dot{\vec{q}}_d(t)$ and $\ddot{\vec{q}}_d(t)$ is the time-varying trajectory and its successive derivatives which describe the desired velocity and acceleration, respectively.

The control law of equations (29) and (30) applied to the robot dynamics gives the close-loop equation:

$$\ddot{\vec{e}} - K_d\dot{\vec{e}} - K_p\vec{e} = 0 \tag{31}$$

The error equation (31) is exponentially stable by a suitable choice of the matrices K_d and K_p. Figure 3 shows the inverse dynamics controller proposed. The figure shows how nonlinearities, such as Coriolis, gravity and inertia terms, can easily be compensated for by adding these forces to the control input. The proportional and derivative terms compose the linear auxiliary control input $\vec{a}$.

Figure 3: Point-to-point inverse dynamic controller

In order to validate the methodology proposed in this paper, various Matlab/Simulink schemes have been developed. Fig. 4 illustrates the reference and real position of the robot joint q_1. As can be seen, a good response is obtained in following the trajectory. The response is quite similar for joints q_6 and q_8.

Figure 4: Reference and robot position

3.2 Position task space control

Control of parallel robots is naturally achieved in the joint space, since the control inputs are the joint torques. Nevertheless, the user specifies a motion in the task space, and thus it is important to extend the control problem to the task space. This can be achieved by following the equation below (Canudas, Siciliano and Basting, 1996), for example:

$$\vec{\tau}_c = J^T(\vec{q})\cdot K_p \cdot \left(\vec{x}_d - \vec{x}\right) - J^T(\vec{q})\cdot K_d \cdot J(\vec{q})\cdot \vec{q} + \vec{G}(\vec{q}) \tag{32}$$

where $\vec{x}_d$ is the desired robot location vector, x is the robot location vector, J is the Jacobian matrix, $\vec{q}$ and $\vec{q}$ are the robot position and velocity, and $\vec{G}(\vec{q})$ is the robot gravity term. Fig. 5 shows the task space controller that has been developed.

Figure 5: Task space controller

As in the previous cases, simulations have also been implemented with this controller. Fig. 6 shows the height (z) reference for the platform and the robot height, while Fig.7 illustrates the parallel robot platform orientation: the *gamma* and *beta* angles.

Figure 6: Response task space controller: robot height

Figure 7: Response task space controller: robot gamma and beta angles

3.3 Force control

In addition to controlling position, force control is increasingly being used in industrial robots. Various types of control strategies can be chosen to establish force control, the first of which is based on typical force control and consists of following a force reference. This control is restricted to a linear (Volpe and Khosla, 1993) control, such as PID force control. The effect of the three parameters of a PID controller is well known and has been extensively described in the literature, e.g. (Astrom and Murray, 2008). The integral term ensures zero tracking error while the function of the derivative term is to dampen the system. The resulting force control law is given in equation (33):

$$\tau_f = K_p(F_{ref} - f) + K_I \int (F_{ref} - f)dt + K_d \frac{d}{dt}(F_{ref} - f) \tag{33}$$

where τ_f is the force control action, F_{ref} is the force reference, f is the measured force and K_p, K_i and K_d are proportional, integral and derivative constants.

A Matlab/Simulink scheme has been developed in order to simulate and analyze force control. This scheme uses the force control of equation (33) and follows the control architecture presented in (Valera et al., 2011). As mentioned above, due to the configuration of the parallel robot, it has three degrees of freedom: the height (z) of the platform and the orientation (γ and β). Thus, the robot has an established control force/position: force control in z axis and orientation control of the robot's platform. The figure below illustrates a control scheme developed for this work.

Figure 8: Force control scheme developed

Fig. 8 shows the controller developed in this work. The position reference for the parallel robo is x_d, while f_d is the force reference that the robot shall apply to the environment. The specific force control algorithm is programmed in the "Force Controller" block using the error signal between the force reference and the force measured on the robot's platform. Finally, new position references q_d are computed and sent to the joint space robot controller. These references are obtained through the outputs of the parallel robot's inverse kinematics ($\hat{q}_d$) and the force controller (Δq).

In Figure 9 the force applied by the robot (in red) follows the reference with a very small degree of error.

Figure 9: Parallel robot force controller response.

4 DEVELOPMENT OF FORCE CONTROL OVER THE ACTUAL PROTOTYPE

An industrial PC has been used to implement the control architecture for the parallel robot. It is based on a high performance 4U Rackmount industrial system with 7 PCI slots and 7 ISA slots. It has a 2.5 GHz Intel® Pentium® Core 2 Quad/Duo processor and 4 GB SDRAM.

The industrial PC is equipped with two Advantech™ data acquisition cards: a PCI-1720 and a PCL-833. The first is used to supply the control actions for each parallel robot actuator and the second is used to read the encoder measurements.

The force control is established with the ATI sensor Delta SI-330-30. This is a sensor made of silicon strain gauges with 6 DOF which can measure forces and moments on the XYZ axes, providing a measurement error of nearly zero. The control unit measures these signals using the Network Force/Torque sensor system. It provides an Ethernet/IP communication interface. Fig. 10 illustrates the control architecture developed for this robot.

Figure 10: Actual robot positions

The parallel robot is controlled using the C++ programming language (with Orocos middleware). The PC is equipped with the Linux Ubuntu system, patched with Xenomai (a real-time kernel). Thus, real-time features are available. Using this environment, joint space control strategies based on inverse dynamics have been implemented. The sample time used for the control algorithm is 10mS.

Fig 11a shows the reference and the real position (for the joint q1) of the parallel robot. As can be seen in Fig. 11b, a high accuracy is obtained.

(a) (b)

Figure 11: Actual robot position and error

Based on the calculated Jacobian matrix, a task space position controller has also been implemented using the actual parallel robot and Orocos middleware. The sample time is the same as for the joint space controller: 10mS. Fig. 12 shows the robot height while Fig. 13 shows the platform orientation (roll in Fig 13a and pitch in Fig. 13b). As occurs with joint space control, errors in the executions are very small.

Figure 12: Height of the robot platform

(a) (b)

Figure 13: Robot platform orientation

Finally, the force robot control has been implemented with the parallel robot. The sample time of this controller is 10mS. Fig. 14 illustrates the reference and the force applied by the robot on the *z* axis. For the first seven seconds (approximately), the force applied by the robot is zero as during that time the platform is moving down and has no contact with the environment. From that point on, the terminal element collides with the environment and the control force is established, using a sinusoidal force reference. As can be seen in the figure, the force applied by the robot follows the reference very accurately. Fig. 15 shows the control action for the actuator of coordinate q_1.

Figure 14: Reference and actual robot force

Figure 15: q1 control action

5 CONCLUSIONS

This paper studied the position and force control of a parallel robot with 3 DOF. In addition to designing the robot, a completely open and flexible control system (based on an industrial PC) was developed. The PC was equipped with card D/A converters, encoder reading and a force sensor. Programs and control applications were developed with the programming language C++.

In order to establish control of the robot, direct and inverse kinematic models of the robot were solved, as well as the dynamic model and the Jacobian needed to implement force control.

A model-based controller was chosen to develop the position control. The response obtained with this position control, both in simulations and with the real robot, showed accurate response in terms of position error.

A hybrid force/position controller was also implemented in this paper. A force sensor was installed on the platform in order to measure the forces and torques applied by the robot. The performance obtained from simulations and with the actual prototype demonstrated that the robot can accomplish the required task.

ACKNOWLEDGEMENTS

The authors wish to thank the Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (FEDER-CICYT) for the partial funding of this study under the projects DPI2011-28507-C02-01 and DPI2013-44227-R. This work was also partially supported by the Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT-Venezuela).

REFERENCES

1. Afonso, G., Pires, J. N., Estrela, N., (2007). Force control experiments for industrial applications: A test case using an industrial deburring example. *Assem. Autom. J.* 26(2):148–156.
2. Astrom, K. J., Murray, R., (2008). *Feedback Systems*. Princeton, NJ: Princeton Univ. Press.
3. Bellakehal, S., Andreff, N., Mezouar, Y., Tadjine. (2011). Force/position control of parallel robots using exteroceptive pose measurements. *Meccanica* 46(1):195-205.
4. Canudas, C., Siciliano, B., Basting, G., (1996). *Theory of Robot Control*. Ed. Springer.
5. Cao, R., Gao, F., Zhang, Y., Pan, D., Chen, W. (2015). A new parameter design method of a 6-DOF Parallel Motion Simulator for a Given Workspace. *Mechanics Based Design of Structures and Machines* 43(1):1-18.
6. Carretero, J.A., Podhorodeski, R.O., Nahon, M.A. (2000). Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. *Journal of Mechanical Design* 122(1):17-25.
7. Clavel, R., (1988). DELTA, a fast robot with parallel geometry. Proceedings of 18th International Symposium on Industrial Robot, Lausanne, April, 91-100.
8. Craig, J., (1989). *Introduction to robotics: Mechanics and Control*. Ed. Addison-Wesley.
9. Díaz-Rodriguez M., Mata, V., Valera A., Page, A., (2010). A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. *Mechanism and Machine Theory* 45:1337-1356.
10. Díaz-Rodriguez M., Valera, A., Mata, V., Valles, M., (2013). Model-Based Control of a 3-DOF Parallel Robot Based on Identified Relevant Parameters. *IEEE/ASME Transaction on Mechatronics*. 18(6):1737-1744.
11. Farhat, N., Mata, V., Page, A., Valero, F. (2008). Identification of dynamic parameters of a 3-DOF RPS parallel manipulator. *Mechanism and Machine Theory* 43(1):1-17.
12. Garg, A., Vikram, C.S., Gupta, S., Sutar, M.K., Pathak, P.M., Mehta, N.K., Shamar, A.K. Gupta, V.K. (2014). Design and development of in vivo robot for biopsy. *Mechanics Based Design of Structures and Machines* 42(3):278-295.

13. Gough, V.E., Whitehall, S.G., (1962). Universal tire test machine. Proceedings of 9th International Technical Congress FISITA, 117-135.
14. Jalón, J.G., Bayo, E., (1994). *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge*. Springer-Verlag, New-York.
15. Lee, K.M., Arjunan, S., (1991). A three-degrees-of-freedom micromotion in parallel actuated manipulator. *IEEE Transactions on Robotics and Automation* 7(5):634–641.
16. Li, Y., Xu, Q. (2007). Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation. *IEEE/ASME Transaction on Mechatronics* 12(3):265–27.
17. Merlet, J.P., (2000). *Parallel Robots*. Kluwer, London, U.K.
18. Pierrot, F., Nabat, V., Company, O., Krut, S., Poignet, P., (2009). Optimal design of a 4-dof parallel manipulator: From academia to industry. *IEEE Transactions on Robotics* 25(2):213-224.
19. Rosillo, N., Valera, A., Benimeli, F., Mata, V., Valero, F., (2011). Real-time solving of Dynamic Problem in Industrial Robots. *Industrial Robot* 38(2):119-129.
20. Spong, M., Vidyasagar, M., (1989). *Robot dynamics and control*. Ed. John Wiley & Sons.
21. Steward, D. A. (1965). A platform with 6 degrees of freedom. Proceedings of the Institution of mechanical engineers. Part 1 vol. 15, 371-386.
22. Tsai, L. W., (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulator*. Wiley Interscience, Canada.
23. Valera, A., Benimeli, F., Solaz, J., De Rosario H., Robertsson, A., Nilsson, K., Zotovic, R., Mellado, M. (2011). A car-seat example of automated anthropomorphic testing of fabrics using force-controlled robot motions. *IEEE Trans. on Automation Science and Engineering* 8(2):280-291.
24. Valles, M., Díaz-Rodríguez, M., Valera, A., Mata, V., Page, A. (2012). Mechatronic Development and Dynamic Control of a 3-DOF Parallel Manipulator. *Mechanics Based Design of Structures and Machines* 40(4):434-452.
25. Volpe, R. and Khosla, P., (1993). A theoretical and experimental investigation of explicit force control strategies for robot manipulators. *IEEE Trans. Automatic Control* 38(11):1634-1650.
26. Yoshikawa, T., (1990). *Foundations of Robotics: Analysis and control*. Ed. The MIT Press.
27. Zarkandi, S. (2011). Kinematics and Singularity Analysis of a Parallel Manipulator with Three Rotational and one Translational DOFs. *Mechanics Based Design of Structures and Machines* 39(3):392-407.
28. Zeng G., Hemami, A. (1997). An overview of robot force control. *Robotica* 15(5):473-482.