Robotics and Computer-Integrated Manufacturing 30 (2014) 468–477

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Adaptive control of a 3-DOF parallel manipulator considering payload handling and relevant parameter models

J. Cazalilla [a], M. Vallés [a], V. Mata [b], M. Díaz-Rodríguez [c], A. Valera [a,*]

[a] Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Valencia, CO 46022, Spain
[b] Centro de Investigación de Tecnología de Vehículos, Universidad Politécnica de Valencia, Valencia, CO 46022, Spain
[c] Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida, CO 5101, Venezuela

ARTICLE INFO

Article history:
Received 5 December 2013
Received in revised form
11 February 2014
Accepted 26 February 2014
Available online 3 April 2014

Keywords:
Parallel robot
Model-based control
Adaptive control
Dynamic parameter identification

ABSTRACT

Model-based control improves robot performance provided that the dynamics parameters are estimated accurately. However, some of the model parameters change with time, e.g. friction parameters and unknown payload. Particularly, off-line identification approaches omit the payload estimation (due to practical reasons). Adaptive control copes with some of these structural uncertainties. Thus, this work implements an adaptive control scheme for a 3-DOF parallel manipulator. The controller relies on a novel relevant-parameter dynamic model that permits to study the cases in where the uncertainties affect: (1) rigid body parameters, (2) friction parameters, (3) actuator dynamics, and (4) a combination of the former cases. The simulations and experiments verify the performance of the proposed controller. The control scheme is implemented on the modular programming environment Open Robot Control Software (OROCOS). Finally, an experimental setup evaluates the controller performance when the robot handles a payload.

1. Introduction

Researchers began to focus on parallel manipulators because of the advantages they hold over their serial counterparts: high stiffness, accuracy, speed, and payload handling. New architectures of PM have been proposed in academia while some of them has been implemented to real world applications e.g. medical applications [1,2]. However, in order to reach potential advantages over serial manipulator, PMs still require improvement in their design, modeling and control [3]. A vast literature deals with the kinematics and dynamic of a PMs, while the development of control schemes provides a field for improving a robot's performance [4] which is particularly in the interest of this paper.

One of the most common control technique applied to PMs is the family of PID controllers which are mainly designed for trajectory tracking control (see [5–8] among others). as well as the implementation of control strategies based on fuzzy logic [8]. Nowadays, the demand for fast and accurate robots points out the need for the implementation of model-based control [5]. For instance, computed torque control (CTC) linearizes and decouples the system dynamics through feedback loop compensation [4,9–11]. However, CTC computes the dynamic model in real time, which increases the computational burden on the control system. On the other hand, Augmented PD control (APD) improves the tracking control by compensating the system dynamics which is computed off-line [12]. APD computes the dynamic model with the desired trajectory values ignoring the changes in the actual system state variables. CTC and APD require accurate values of dynamic model parameters, which are found through experimental parameter identification techniques [13]. Nevertheless, due to the topology of PMs – especially for lower-mobility PMs (less than 6 DOF) – some of the robot's links moves with poor excitation which affects the identification of the parameters [14]. In addition, the payload may be unknown and some of the system parameters may change (e.g. friction parameters) such that model-based control loses performance.

Robust control deals with time variant parameters, thus, in [15] a nonlinear task, space control is applied to 6 DOF PM. However, the control design is based on practical implementation aspects discussed in [16]. Another approach for dealing with time-varying uncertainties is adaptive control. In [17], and adaptive control is developed, considering a linear feedback controller with a dynamic feedforward compensator, for a 6 DOF PM. The control approach identifies the parameters of the dynamic model through a nonlinear scheme. However, the authors used a simplified model without considering the identifiability of the dynamics parameters.

In [18,19], an adaptive computed torque control and nonlinear adaptive control have been developed on a planar PM. The controllers

* Corresponding author.
E-mail addresses: jcazalilla@ai2.upv.es (J. Cazalilla), mvalles@ai2.upv.es (M. Vallés), vmata@mcm.upv.es (V. Mata), dmiguel@ula.ve (M. Díaz-Rodríguez), giuprog@ai2.upv.es (A. Valera).

http://dx.doi.org/10.1016/j.rcim.2014.02.003
0736-5845/

are implemented in task space which requires extra sensors (the control signal and the measurements are normally applied in actuator space). Thus, task space coordinates were found from the actuated joint measurement through forward kinematics which increases the computational burden and could be impractical for a spatial PM.

The adaptive controllers applied thus far lack of the following consideration: (1) to adapt the model for only a subset of the parameters of a relevant parameters model and (2) to setup an experiment where the robot handles a payload while moving. Motivated by this consideration, this paper develops an adaptive control scheme for a low-cost 3-DOF spatial PM. The control scheme updates the model's parameters on line due to the implementation of a relevant parameter model [20] that can be computed in real time [21]. The model includes 12 parameters: three rigid bodies, three actuator dynamics and six frictions parameters (in order to have a model linear in parameters, the friction in the actuated joints considers a Coulomb plus viscous friction model). Simulations and experiments evaluate four study cases of adaptive control, as regard to the subset of parameters identified online. The first case identifies on-line only the rigid body parameters. As will be seen in the modeling section, the rigid body parameters include the platform's mass and therefore this model compensates for an unknown payload. The second case explores friction parameters, the third considers actuator dynamics uncertainties and the fourth case combines the three former cases.

Different adaptive controls and a fixed passivity-based controller (PD+) [22] are implemented in the modular programming middleware OROCOS. The tracking trajectory performances of the adaptive controllers and the PD+ controller were compared through simulations and experiments. Moreover, the experiments section includes a novel setup where a payload is placed onto the platform while the robot is executing a task.

This paper is organized as follows: Section 2 shows the PM design, while Section 3 deals with the model-based control schemes. Section 4 shows the results from the simulations and the experiments. Finally, Section 5 summarizes the conclusions.

2. The 3-DOF parallel manipulator

2.1. Physical description of the low-cost PM

As mentioned before, a 3-DOF spatial PM was used to address the controller design problem. The robot (see Fig. 1) consists of three kinematic chains, with each chain having a PRS configuration (P, R and S standing for prismatic, revolute and spherical joint, respectively). The underline format (P) stands for the actuated joint. The choice of the PRS configuration was guided by the need to develop a low-cost robotic platform with two DOF of angular rotation in two axes (rolling and pitching) and one DOF of translation motion (heave). In [23], a complete description of the mechatronic development process of the PM is presented.

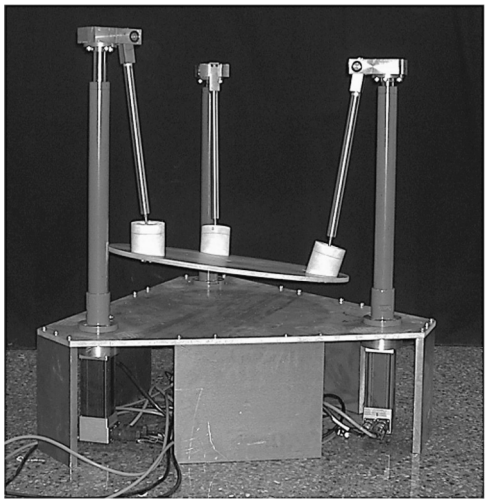

Fig. 1. Parallel robot used in the experiment.

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. The lower part of the coupler is connected with a revolute joint to the ball screw, while the upper part is connected to the moving platform through a spherical joint. The lower part of the ball screws are perpendicularly attached to the platform's base and are positioned at the base in an equilateral triangular configuration. The ball screw transforms the rotational movement of the motor into linear motion.

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.

2.2. Kinematic and dynamic model

For modeling purposes, mobile reference systems have been attached to the robot links using Denavit–Hartenberg's notation; a detailed explanation is provided in [23]. Fig. 2 shows a kinematic sketch of the robot. Nine generalized coordinates are used to model the robot (q_i, where $i=1\ldots9$). The active coordinates; q_1, q_6 and q_8 are associated with the actuated prismatic joints (P). The passive coordinates q_2, q_7 and q_9, are associated with the revolute joints (R), and q_3, q_4 and q_5 correspond to only one of the spherical joints (S, located at P_1 in Fig. 2). The spherical joint has been modeled by means of three mutually perpendicular rotational joints.

The forward kinematics is solved using a geometric approach. From the rigid body link assumption, Fig. 2 shows that the length between the locations of the spherical joints P_i and P_j is constant and equal to l_m; thus, the constraints equations are as follows:

$$\Psi_1(q_1,q_2,q_6,q_7)=||(\vec{r}_{A_1B_1}+\vec{r}_{B_1P_1})-(\vec{r}_{A_1A_2}+\vec{r}_{A_2B_2}+\vec{r}_{B_2P_2})||-l_m=0 \quad (1)$$

$$\Psi_2(q_1,q_2,q_8,q_9)=||(\vec{r}_{A_1B_1}+\vec{r}_{B_1P_1})-(\vec{r}_{A_1A_3}+\vec{r}_{A_3B_3}+\vec{r}_{B_3P_3})||-l_m=0 \quad (2)$$

$$\Psi_3(q_6,q_7,q_8,q_9)=||(\vec{r}_{A_1A_3}+\vec{r}_{A_3B_3}+\vec{r}_{B_3P_3})-(\vec{r}_{A_1A_2}+\vec{r}_{A_2B_2}+\vec{r}_{B_2P_2})||-l_m=0 \quad (3)$$

In the forward kinematics, the position of the actuators is known (q_1, q_6 and q_8), thus the system (1)–(3) is nonlinear with q_2, q_7 and q_9 as unknown. The Newton–Raphson (N–R) numerical method has

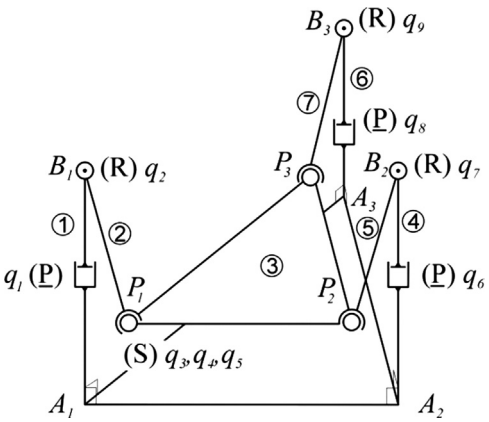

Fig. 2. Kinematic sketch of the parallel robot.

been chosen to solve the nonlinear system. The method converges rather quickly when the initial guess is close to the desired solution [24]. Once those coordinates have been obtained, the location of points P_i is acquired. With these three points, the rotational matrix defining the orientation of the mobile platform with regard to the fixed base is easily obtained. The remaining generalized coordinates (q_3, q_4 and q_5) are found in a straightforward manner from the rotation matrix.

The inverse kinematics analysis consists of finding the actuated generalized coordinates given the roll, the pitch angles, and the heave of the reference system attached to the mobile platform. From these values, the coordinates of points P_i can be obtained following [25]:

$$\vec{r}_{A_1P_1} - q_1 \cdot \vec{u}_{A_1B_1} = l_m \cdot \vec{u}_{B_1P_1} \tag{4}$$

In (4) $\vec{u}_{AB}$ is a unit vector. Analytical expressions for the generalized coordinate q_1 are obtained squaring both sides of (4). A similar procedure can be applied to the other two limbs.

One of the objectives of this paper is to develop an open control architecture allowing the implementation and testing of dynamic control schemes. The dynamic controller requires that the equation of motion be described as follows:

$$M(\vec{q}, \vec{\Phi}) \cdot \ddot{\vec{q}} + C(\vec{q}, \dot{\vec{q}}, \vec{\Phi}) \cdot \dot{\vec{q}} + \vec{G}(\vec{q}, \vec{\Phi}) = \vec{\tau} \tag{5}$$

In (5), M represents the system mass matrix, C is the matrix grouping the centrifugal and Coriolis terms, $\vec{G}$ is the vector corresponding to gravitational terms and $\vec{\tau}$ is the vector of generalized forces. It is worth noting that (5) 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 by a set of independent generalized coordinates. The actuated joint coordinates are the set chosen as the independent coordinates $\vec{q} = [q_1 \quad q_6 \quad q_8]^T$.

The dynamic model parameters are experimentally identified. This set of parameters not only includes the rigid body parameters, but also the rotor and screw dynamics of the robot actuators, as well as the friction in actuated joints. The assumed Coulomb and viscous friction in the ith joint has been modeled as follows:

$$\tau_{f_i} = -(F_{C_i} \cdot \mathrm{sign}(\dot{q}_i) + F_{v_i} \cdot \dot{q}_i) \tag{6}$$

In (6), F_{C_i} and F_{v_i} are the coefficients of the Coulomb and viscous friction respectively. Due to the characteristics of the actuators (ball screws), only friction in the actuated joint is considered. Eq. (5) can be rewritten (see [5,13,20]) as follows:

$$[K_{rb} \quad K_r \quad K_f] \cdot \begin{bmatrix} \vec{\Phi}_{rb} \\ \vec{\Phi}_r \\ \vec{\Phi}_f \end{bmatrix} = \vec{\tau} \tag{7}$$

In (7), the vectors $\vec{\Phi}_{rb}$, $\vec{\Phi}_r$, and $\vec{\Phi}_f$ group the rigid body, rotor and friction parameters. K_i is the part of the coefficients matrix that determines the linear relationship between the corresponding parameters (rigid body, rotor and friction) and the generalized forces.

From (7), a set of base parameters corresponding to the complete base parameter model can be obtained (Tables 1–3). In Table 3, lm is the characteristic length of the mobile platform, lr is the length of the coupler links connecting the platform to the linear actuators, m_i, mx_i, my_i, mz_i, Ixx_i, Ixy_i, Ixz_i, Iyy_i, Iyz_i, and Izz_i are the mass, the first and second moments of inertia of the ith link with regard to its local reference system.

The base parameter model cannot always be properly identified. For this reason, a reduced model containing the relevant parameters is obtained in this paper through a process that considers the robot's leg symmetries, the influence on the dynamic behavior of the robot, the statistical significance of the identified parameters, and the physical feasibility of the parameters [20].

Table 1
Friction base parameters for the 3-PRS PM.

$\vec{\Phi}_f$	Base parameter	Value (SI units)
1	F_{c1}	152.70
2	F_{v1}	3267.45
3	F_{c2}	118.32
4	F_{v2}	2127.48
5	F_{c3}	247.97
6	F_{v3}	2120.09

Table 2
Actuator base parameters for the 3-PRS PM.

$\vec{\Phi}_r$	Base parameter	Value (SI units)
1	J_1	222.75
2	J_2	222.75
3	J_3	222.75

Table 3
Rigid body base parameters for the 3-PRS PM.

$\vec{\Phi}_{rb}$	Base parameter	Value (SI units)
1	$I_{zz2} - l_r^2 \sum_{i=1}^{2} m_i$	−2.7068
2	$mx_2 + l_r \sum_{i=1}^{2} m_i$	−4.071
3	my_2	0
4	$I_{xx3} - (sin(2/3\pi) l_m)^2 \sum_{i=1}^{5} m_i$	−5.2431
5	$I_{xy3} + cos(2/3\pi) sin(2/3\pi) {l_m}^2 \sum_{i=1}^{5} m_i$	2.9887
6	I_{xz3}	0.1539
7	$I_{yy3} - (cos(2/3\pi) l_m)^2 \sum_{i=1}^{3} m_i + sin(2/3\pi) l_m \sum_{i=4}^{5} m_i$	1.6483
8	I_{yz3}	0
9	$I_{zz3} - l_r^2 \sum_{i=1}^{3} m_i$	−3.8104
10	$mx_3 - cos(2/3\pi) l_m \sum_{i=1}^{5} m_i + l_m \sum_{i=4}^{5} m_i$	−0.0014
11	$my_3 - sin(2/3\pi) l_m \sum_{i=1}^{5} m_i$	−11.568
12	mz_3	−0.5623
13	$I_{zz5} - lr^2 \sum_{i=4}^{5} m_i$	−2.6436
14	$mx_5 - lr \sum_{i=4}^{5} m_i$	−4.884
15	my_5	0
16	$I_{zz7} + lr^2 \sum_{i=1}^{5} m_i$	8.7408
17	$\sum_{i=1}^{7} m_i$	38.8162
18	$mx_7 + l_r \sum_{i=6}^{7} m_i$	16.3594
19	my_7	0

The relevant parameters model consists of the friction parameters from Table 1, the actuator parameters from Table 2 and the rigid body parameters 11, 17 and 18 from Table 3.

$$\Omega_1 = my_3 - sin(2/3\pi) l_m \sum_{i=1}^{5} m_i \tag{8}$$

$$\Omega_2 = \sum_{i=1}^{7} m_i \tag{9}$$

$$\Omega_3 = mx_7 + lr \sum_{i=6}^{7} m_i \tag{10}$$

In (8)–(10) m_3 is the mass of the platform. It can be seen that if a payload is placed in the center of the platform, its mass can be identified together with the mass of the platform.

3. Model-based position control schemes

Eq. (5) has several properties that can be exploited to facilitate dynamic controller designs. One of the most useful properties is that there is a reparametrization of all unknown parameters into a parameter [$px1$] vector $\vec{\Phi}$ that rewrites the system dynamics linearly in parameters. Therefore, the following holds:

$$M(\vec{q},\vec{\Phi})\cdot\ddot{\vec{q}}+C(\vec{q},\dot{\vec{q}},\vec{\Phi})\cdot\dot{\vec{q}}+\vec{G}(\vec{q},\vec{\Phi})\equiv M_0(\vec{q})\cdot\ddot{\vec{q}}+C_0(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}}+\vec{G}_0(\vec{q})+Y(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\cdot\vec{\Phi} \quad (11)$$

The term in (11) for the actual 3-PRS parallel robot based on relevant parameters can be written as follows:

$$C(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}}=\begin{bmatrix}F_{v_1}\dot{q}_1+F_{c_1}\mathrm{sign}(\dot{q}_1)\\ F_{v_2}\dot{q}_6+F_{c_2}\mathrm{sign}(\dot{q}_6)\\ F_{v_3}\dot{q}_8+F_{c_3}\mathrm{sign}(\dot{q}_8)\end{bmatrix}+\begin{bmatrix}C_{11}&C_{12}&C_{13}\\ C_{21}&C_{22}&C_{23}\\ C_{31}&C_{32}&C_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix} \quad (12)$$

$$M(\vec{q})\cdot\ddot{\vec{q}}=\begin{bmatrix}J_1&0&0\\ 0&J_2&0\\ 0&0&J_3\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_6\\ \ddot{q}_8\end{bmatrix}+\begin{bmatrix}M_{11}&M_{12}&M_{13}\\ M_{21}&M_{22}&M_{23}\\ M_{31}&M_{32}&M_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix} \quad (13)$$

$$G(\vec{q})=g\begin{bmatrix}G_{11}&G_{12}&G_{13}\\ G_{21}&G_{22}&G_{23}\\ G_{31}&G_{32}&G_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix} \quad (14)$$

In addition, $M_0(.)$, $C_0(.)$, and $\vec{G}_0(.)$ represent the known part of the system dynamics, and $Y(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})$ is a regressor matrix with dimensions [nxp] containing nonlinear but known functions.

Having a dynamic model that is linear in parameters, the left-hand side of (5) can be written as follows:

$$M_0(\vec{q})\cdot\ddot{\vec{q}}+C_0(\vec{q},\dot{\vec{q}})\cdot\dot{\vec{q}}+\vec{G}_0(\vec{q})+Y(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\cdot\vec{\Phi}=\vec{\tau} \quad (15)$$

In the following, different adaptive scenarios are presented: (1) rigid body parameters, (2) friction parameters, (3) actuator dynamics, and (4) all the aforementioned.

3.1. Adaptive Model I

If the rigid body parameters constituting the reduced model are assumed to be unknown, then (15) can be written as follows:

$$\vec{\tau}=\begin{bmatrix}J_1&0&0\\ 0&J_2&0\\ 0&0&J_3\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_6\\ \ddot{q}_8\end{bmatrix}+\begin{bmatrix}F_{v_1}\dot{q}_1+F_{c_1}\mathrm{sign}(\dot{q}_1)\\ F_{v_2}\dot{q}_6+F_{c_2}\mathrm{sign}(\dot{q}_6)\\ F_{v_3}\dot{q}_8+F_{c_3}\mathrm{sign}(\dot{q}_8)\end{bmatrix}+Y_1(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\vec{\theta}_1 \quad (16)$$

where

$$Y_1(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})=\begin{bmatrix}M_{11}&M_{12}&M_{13}\\ M_{21}&M_{22}&M_{23}\\ M_{31}&M_{32}&M_{33}\end{bmatrix}+\begin{bmatrix}C_{11}&C_{12}&C_{13}\\ C_{21}&C_{22}&C_{23}\\ C_{31}&C_{32}&C_{33}\end{bmatrix}+g\begin{bmatrix}G_{11}&G_{12}&G_{13}\\ G_{21}&G_{22}&G_{23}\\ G_{31}&G_{32}&G_{33}\end{bmatrix} \quad (17)$$

$$\vec{\theta}_1=[\Omega_1\quad\Omega_2\quad\Omega_3]^T \quad (18)$$

3.2. Adaptive Model II

In this case Coulomb and viscous friction parameters are unknown, thus,

$$\vec{\tau}=\begin{bmatrix}J_1&0&0\\ 0&J_2&0\\ 0&0&J_3\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_6\\ \ddot{q}_8\end{bmatrix}+\begin{bmatrix}M_{11}&M_{12}&M_{13}\\ M_{21}&M_{22}&M_{23}\\ M_{31}&M_{32}&M_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+\begin{bmatrix}C_{11}&C_{12}&C_{13}\\ C_{21}&C_{22}&C_{23}\\ C_{31}&C_{32}&C_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+g\begin{bmatrix}G_{11}&G_{12}&G_{13}\\ G_{21}&G_{22}&G_{23}\\ G_{31}&G_{32}&G_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+Y_2(\vec{q},\dot{\vec{q}})\vec{\theta}_2 \quad (19)$$

where

$$Y_2(\vec{q},\dot{\vec{q}})=\begin{bmatrix}\dot{q}_1&0&0&\mathrm{sign}(\dot{q}_1)&0&0\\ 0&\dot{q}_6&0&0&\mathrm{sign}(\dot{q}_6)&0\\ 0&0&\dot{q}_8&0&0&\mathrm{sign}(\dot{q}_8)\end{bmatrix} \quad (20)$$

$$\vec{\theta}_2=[Fv_1\quad Fv_2\quad Fv_3\quad Fc_1\quad Fc_2\quad Fc_3]^T \quad (21)$$

3.3. Adaptive Model III

When the parameters of the actuator dynamics are assumed to be unknown, the following equations hold:

$$\vec{\tau}=\begin{bmatrix}M_{11}&M_{12}&M_{13}\\ M_{21}&M_{22}&M_{23}\\ M_{31}&M_{32}&M_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+\begin{bmatrix}F_{v_1}\dot{q}_1+F_{c_1}\mathrm{sign}(\dot{q}_1)\\ F_{v_2}\dot{q}_6+F_{c_2}\mathrm{sign}(\dot{q}_6)\\ F_{v_3}\dot{q}_8+F_{c_3}\mathrm{sign}(\dot{q}_8)\end{bmatrix}+\begin{bmatrix}C_{11}&C_{12}&C_{13}\\ C_{21}&C_{22}&C_{23}\\ C_{31}&C_{32}&C_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+g\begin{bmatrix}G_{11}&G_{12}&G_{13}\\ G_{21}&G_{22}&G_{23}\\ G_{31}&G_{32}&G_{33}\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}+Y_3(\ddot{\vec{q}})\vec{\theta}_3 \quad (22)$$

where

$$Y_3(\ddot{\vec{q}})=\begin{bmatrix}\ddot{q}_1&0&0\\ 0&\ddot{q}_6&0\\ 0&0&\ddot{q}_8\end{bmatrix} \quad (23)$$

$$\vec{\theta}_3=[J_1\quad J_2\quad J_3]^T \quad (24)$$

3.4. Adaptive Model IV

In the same way, combinations where all the relevant parameters are unknown can be considered. For example, if all the dynamic parameters are unknown

$$\vec{\tau}=Y_4\left(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\right)\vec{\theta}_4 \quad (25)$$

where

$$Y_4(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})=\begin{bmatrix}M_{11}&M_{12}&M_{13}&\ddot{q}_1&0&0&0&0&0&0&0&0\\ M_{21}&M_{22}&M_{23}&0&\ddot{q}_6&0&0&0&0&0&0&0\\ M_{31}&M_{32}&M_{33}&0&0&\ddot{q}_8&0&0&0&0&0&0\end{bmatrix}+\begin{bmatrix}C_{11}&C_{12}&C_{13}&0&0&0&\dot{q}_1&0&0&\mathrm{sign}(\dot{q}_1)&0&0\\ C_{21}&C_{22}&C_{23}&0&0&0&0&\dot{q}_6&0&0&\mathrm{sign}(\dot{q}_6)&0\\ C_{31}&C_{32}&C_{33}&0&0&0&0&0&\dot{q}_8&0&0&\mathrm{sign}(\dot{q}_8)\end{bmatrix}+g\begin{bmatrix}G_{11}&G_{12}&G_{13}&0&0&0&0&0&0&0&0&0\\ G_{21}&G_{22}&G_{23}&0&0&0&0&0&0&0&0&0\\ G_{31}&G_{32}&G_{33}&0&0&0&0&0&0&0&0&0\end{bmatrix} \quad (26)$$

$$\vec{\theta}_4=[\Omega_1\quad\Omega_2\quad\Omega_3\quad J_1\quad J_2\quad J_3\quad F_{v_1}\quad F_{v_2}\quad F_{v_3}\quad F_{c_1}\quad F_{c_2}\quad F_{c_3}]^T \quad (27)$$

3.5. Control scheme

In recent years, the passivity-based approach to robot control has received a lot of attention. This approach solves the robot

control problem by exploiting the robot system's physical structure, and specifically its passivity property. The design philosophy of these controllers is to reshape the system's natural (kinetic and potential) energy in such a way that the control objective is achieved.

Bayard and Wen proposed a number of adaptive passivity-based control schemes that do not suffer from the parameter drift problem [26]. These authors have developed a class of adaptive controllers of robot motion, but in this paper a different one has been developed for the parallel robot,

$$\tau_c = M_0(\vec{q})\,\vec{\ddot{q}}_d + C_0(\vec{q},\vec{\dot{q}}_d)\,\vec{\dot{q}}_d + \vec{G}_0(\vec{q}) + Y(\vec{q},\vec{\dot{q}}_d,\vec{\ddot{q}}_d,\vec{\dddot{q}}_d)\cdot\vec{\hat{\theta}} - K_d\,\vec{\dot{e}} - K_p\,\vec{e} \tag{28}$$

$$\frac{d}{dt}\left\{\hat{\theta}(t)\right\} = -\Gamma_0\cdot Y^T(\vec{q},\vec{\dot{q}}_d,\vec{\ddot{q}}_d,\vec{\dddot{q}}_d)\cdot\vec{s}_1 \tag{29}$$

where $\vec{s}_1 = \vec{\dot{e}} + \Lambda_1\vec{e}$, with $\Lambda_1 = \lambda_1 I$, $\lambda_1 > 0$ and $M_0(.), C_0(.), \vec{G}_0(.)$ represent the known part of the system dynamics, and $Y^T(.)$ is the regressor matrix.

The closed-loop system (5)–(28) and (29) is convergent (see Appendix A); that is, the tracking error asymptotically converges to zero and all internal signals remain bounded under suitable conditions on the controller gains K_p and K_d.

In order to compare and validate the adaptive controller, another passivity-based controller has been implemented and tested. The PD+ controller proposed in [22] is implemented and can be written as follows:

$$\tau_c = M\left(\vec{q}\right)\vec{\ddot{q}}_d + C\left(\vec{q},\vec{\dot{q}}\right)\vec{\dot{q}}_d + \vec{G}\left(\vec{q}\right) - K_d\,\vec{\dot{e}} - K_p\,\vec{e} \tag{30}$$

As in (5), $M(.), C(.), \vec{G}(.)$ represent the mass matrix, the centrifugal and Coriolis forces, and the gravitational forces of the robot's dynamic equation. This controller has been chosen because it has very good robust properties. In addition, since both are passivity-based controllers, their expressions are similar and therefore it is easy to compare and analyze their characteristics.

4. Results

In order to validate the performance of the adaptive control algorithm, first of all, several Matlab/Simulink schemes for the parallel robot simulation have been developed. The proposed controller is then implemented in an actual 3-DOF PM.

4.1. 3-PRS PM simulations

As seen in the previous section, the reduced PM model has 12 parameters: three rigid bodies, three actuator dynamics and six corresponding to Coulomb and viscous friction. Depending on which parameters are assumed to be unknown, different adaptive schemes can be developed. In addition, the selection of the adaptive scheme also depends on which parameters are properly identified. This can be measured by the standard deviation of the identified dynamic parameters. A previous study shows that the rigid body parameters are those with a higher standard deviation in comparison with the other identified parameters [20]. Therefore, an adaptive controller based on the Adaptive Model I (Eqs. 16–18) has been simulated. This model only considers the rigid body parameters to be unknown.

Fig. 3a shows the reference and the simulated position for the first joint obtained with the adaptive controller. The control action is calculated by Eqs. (28) and (29) using the regressor matrix (17). In order to compare the control performance, the PD+ controller Eq. (30) has also been simulated. The simulations consider that a

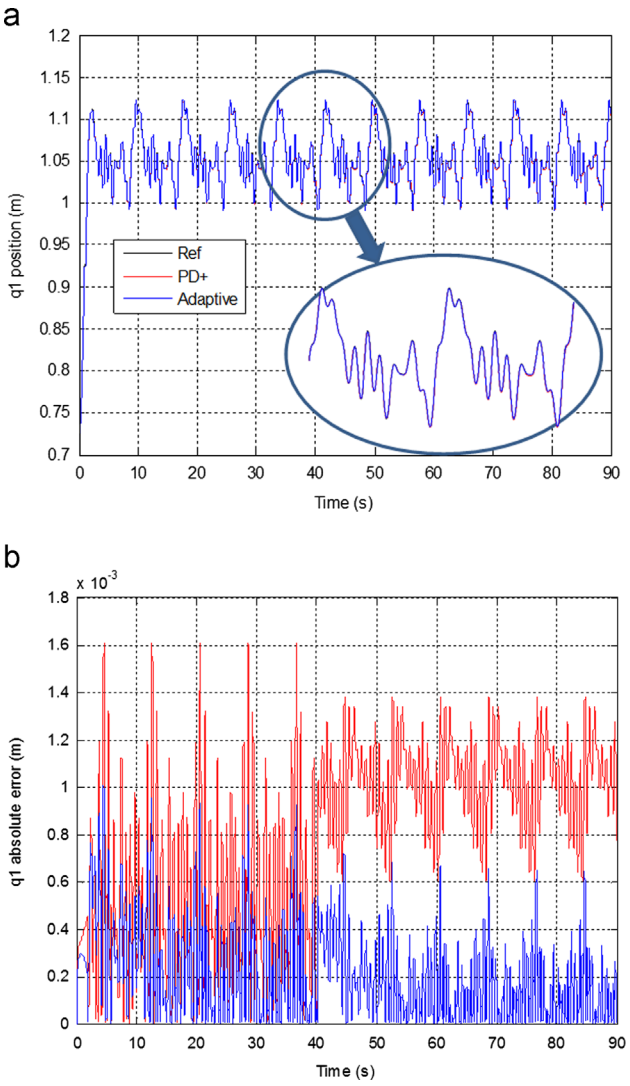

Fig. 3. Reference and simulated position for the first joint with a PD+ and an adaptive controller (a). Absolute error for the first joint with a PD+ and an adaptive controller (b).

mass of 30 kg is placed on the mobile platform m_3 at $t=40$ s. The overall value of the mass changes from 12 kg to 42 kg.

As can readily be appreciated in Fig. 3b, the error presented with both controllers before modifying m_3 is very similar. However, the PD+ controller response is much worse after modifying this mass on the mobile platform. This is because the PD+ controller uses unbiased values for its dynamic parameters. Given that the adaptive controller calculates an estimation of the rigid body parameters, it can take into account and compensate for changes in the platform mass.

In order to analyze the response of the different adaptive controllers considered, another test has been carried out. In this case, three different adaptive controllers have been simulated, which are based on Adaptive Model II (Eqs. (19)–(21)), Adaptive Model III (Eqs. (22)–(24)) and Adaptive Model IV (Eqs. (25)–(27)). As in the previous case, for these simulations a mass of 30 kg has been added to the platform at $t=40$ s.

Fig. 4 illustrates the absolute error obtained for the two periods: before and after adding the 30 kg mass. In this figure, the error obtained with the Adaptive Model I, Adaptive Model II, Adaptive Model III, Adaptive Model IV and PD+ model can be seen. As can be appreciated in Fig. 4, the controllers based on Adaptive Model II and Model III present a mean error that is twice

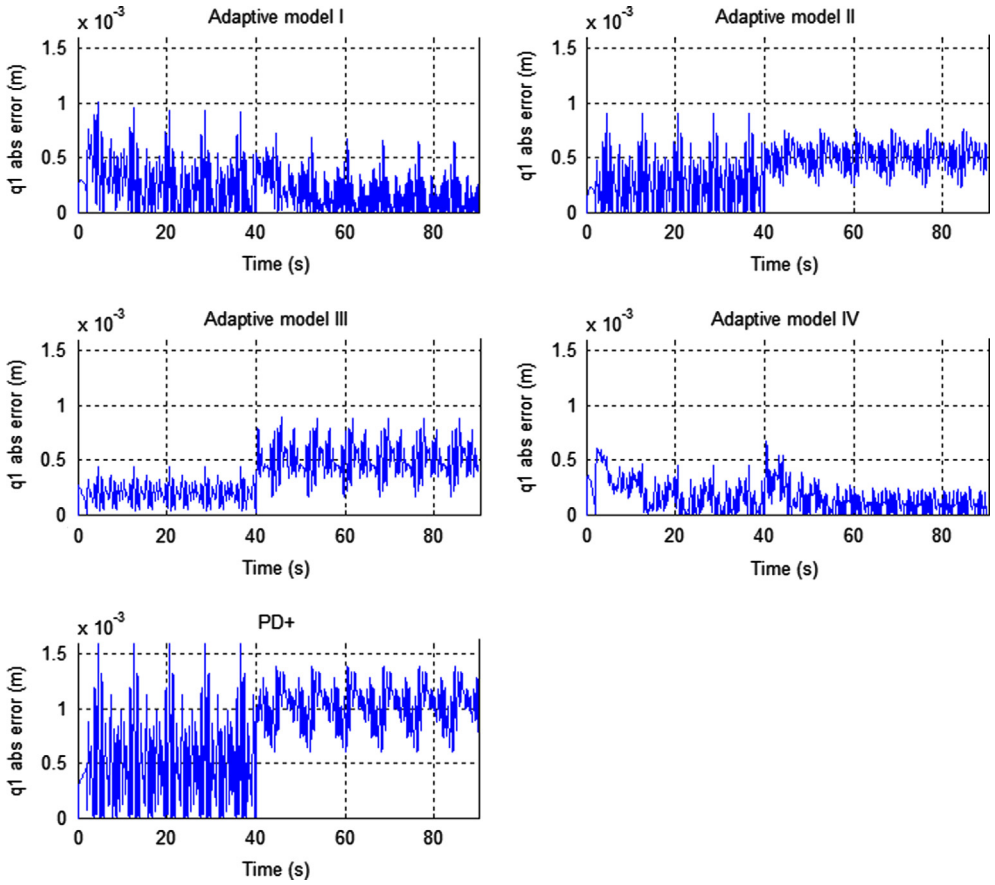

Fig. 4. Absolute first joint position error obtained for a reference in which a mass of 30 kg is added over the platform at the middle of the simulated execution, with Adaptive I–IV and PD+ models.

that of the period where the load is added to the robot platform. This is because such controllers do not calculate an estimation of the parameters that change when the mass is added to the platform and thus their responses are very similar to the PD+ controller. However, given that the adaptive controller based on Model IV computes an estimation of the rigid body, actuator dynamics and friction parameters, it obtains a similar error before and after adding the mass to the platform. In addition, their response is virtually the same as the response of the adaptive controller based on Model I.

The selection of the adaptive model depends, among other factors, on which parameters are assumed to be unknown or may vary due to changes in the operating conditions. Clearly, the larger the number of unknown parameters, the greater the size of some of the constants of the adaptive controller. This may hamper the correct tuning of these constants. For this reason, in the case of variations in the mass added to the platform for example, control based on Adaptive Model I is recommended given that this is the simplest controller able to deal with the variation in parameters.

4.2. Experimental results on the actual 3-PRS PM

In addition to the simulation schemes, the PD+ and adaptive controllers described in this paper have been implemented in a modular way, using real-time middleware Open Robot Control Software (OROCOS). These controllers have been used and analyzed with the actual parallel robot presented in Section 2.

OROCOS [27] is implemented entirely in C++ and, because it is a component-based middleware (being closely linked with a component-based software development), it allows the creation of modules that can operate in real time [28]. Furthermore, within the OROCOS environment there are libraries that are very useful for creating these components. In particular, one of the most relevant is the "OROCOS Toolchain", which is very important as it includes the "Real Time Toolkit (RTT)" and the "OROCOS Component Library (OCL)". The first one deals with everything related to the real-time execution of components, as well as the connection between them, while the second provides the basic primitives for building components. Therefore, through the component-based software development that provides the OROCOS environment, the following advantages can be seen [28]

1. Through the modular design, the program execution flow can very easily be monitored, facilitating the creation of new components and their insertion into the model to obtain new features.
2. The modular structure allows the execution of multiple modules in a distributed manner, obtaining a lower running-time than if they were running serially.
3. The code is fully reusable, which allows an unlimited number of examples to be created for each module.
4. Once loaded, modules are configurable and reconfigurable both in setup time and in running-time, being able not only to change specific parameters for each module, but also general parameters such as the execution priority, etc.

Thus, when a number of modules are implemented and a control scheme is required, it is as simple as inserting the necessary modules to configure them, making connections with each other, and setting up the connections with each other, and making them run. Therefore, given that the different control schemes have common parts due to the development of

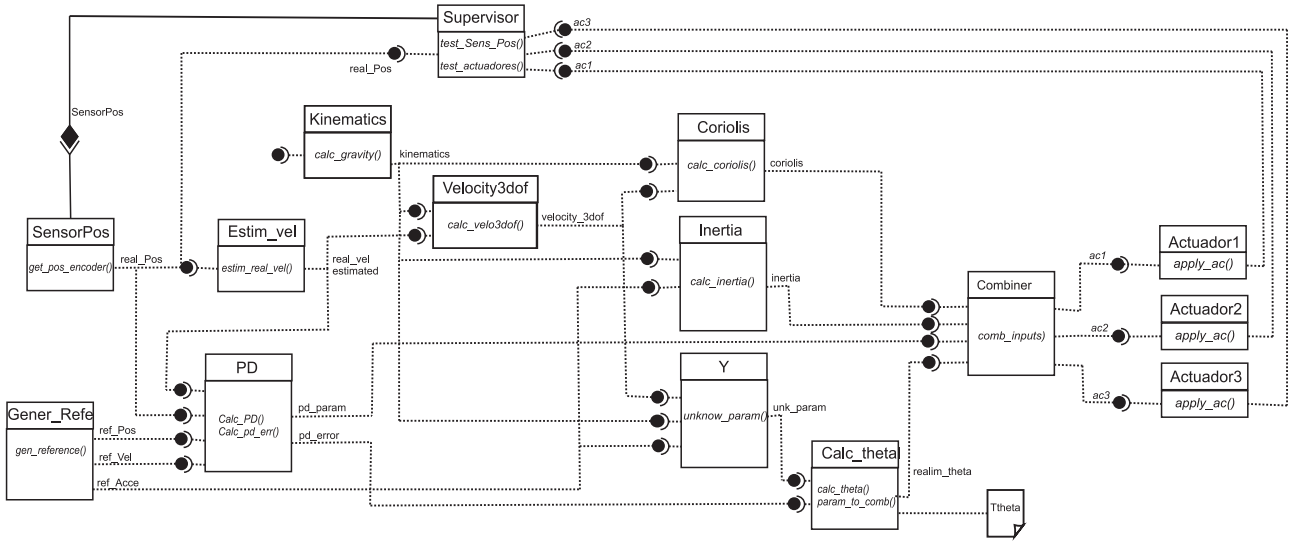

Fig. 5. Component diagram implemented in order to perform an adaptive controller using the middleware Open Robot Control Software (OROCOS).

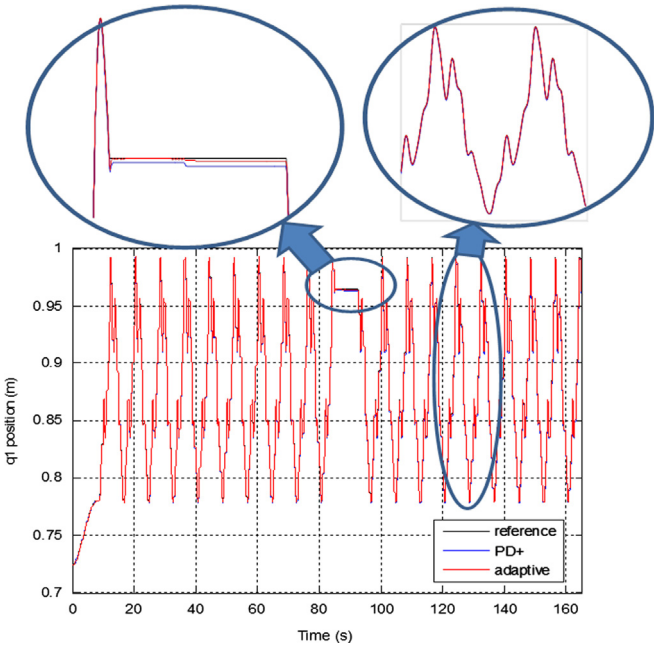

Fig. 6. Reference and real position for the first joint of the parallel robot for both adaptive and PD+ controllers.

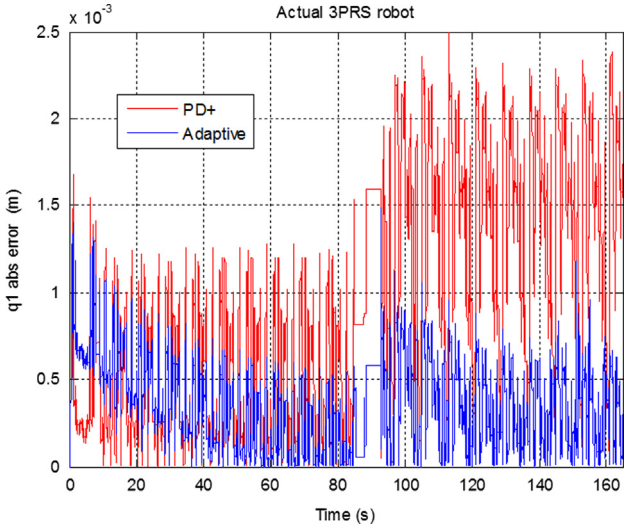

Fig. 7. Absolute first position error, obtained with the real robot, for a reference in which a mass of 30 kg is added over the platform at the middle of the real execution, with adaptive and PD+ controllers.

several modules; these modules are reused to implement different controllers.

Finally, note that although the development of component-based software can be a complicated task at first, in the long run it facilitates the programmer's work because if a module works correctly in one particular scheme, it will certainly work correctly in another control scheme. Therefore, as well as the advantages discussed above, this approach minimizes the chance of possible programming errors in the implementation of any module.

Fig. 5 shows the OROCOS diagram for the adaptive controller implemented based on Adaptive Model I. The *SensorPos* module provides the three joint positions of the robot using the Advantech's PCL-833 encoders card. *Gener_Refe* module calculates the movement references for the robot joints. *PID* module implements a proportional-derivative-integral type controller. *Coriolis and Inertia* modules calculate the robot's fixed dynamic terms of the equation of motion (16). *Y* module calculates the regressor matrix of Eq. (17). The *Calc_theta* module calculates the parameter estimation of Eq. (18). The *combinator* module calculates the control action depending on the control strategy selected. Finally, the *actuator* modules are responsible for carrying out digital-analog conversions through Advantech's PCI-1720 cards. The control scheme also provides the *supervisor* module, which is responsible for monitoring the correct operation of the system. This module is in charge of deactivating the control unit and stops the robot in the case of detecting a malfunction in the system. Due to the distributed execution of the control, the sample time in this experiment has been 10 ms.

Fig. 6 illustrates the reference and the robot q_1 position with PD+ and adaptive controllers. The actual robot motion reference is the same as the reference used in simulation. The only difference is that in the middle of execution, the robot remains in the same position for 8 s (between $t=85$ and $t=93$ s). This time allows a load of 30 kg to be placed onto the robot platform. Fig. 7 illustrates the absolute error values, and the absolute position error for the platform heave is shown in Fig. 8. The control action provided by the adaptive controller is shown in Fig. 9. Finally, Fig. 10 shows the real robot with the added mass.

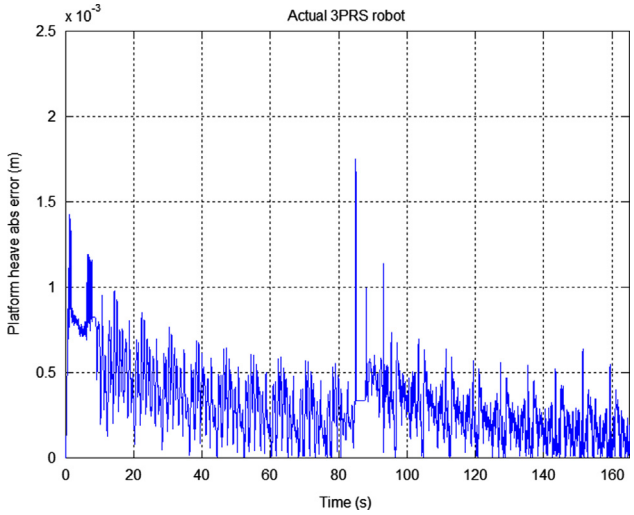

Fig. 8. Absolute heave position error, obtained with the real robot, with an adaptive controller.

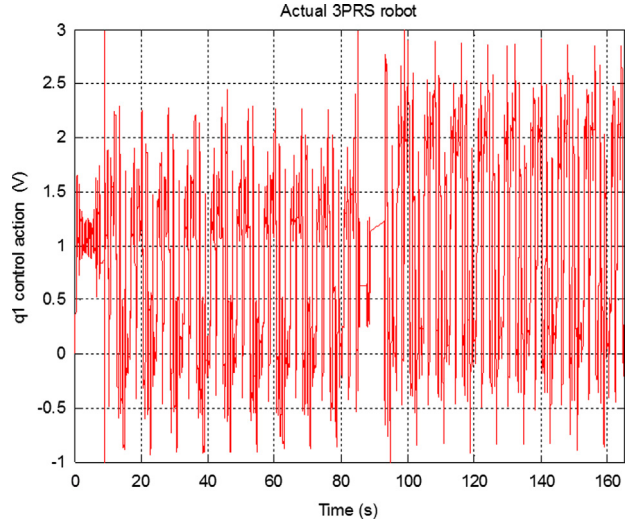

Fig. 9. Control action applied to the robot provided by the adaptive controller

The results obtained with the actual robot are consistent with those obtained in the simulation: because of the estimation of the online dynamic parameters, the change in the load means that robot response using an adaptive controller is significantly better than that obtained with the PD+ controller.

Table 4 demonstrates the absolute mean error and the mean square root error (RMS) between the references and the measured positions of the parallel robot for the two periods (before and after changing the robot load using a mass of 30 kg). As can readily be appreciated, the adaptive controller provides a better response in comparison with the PD+ controller. In fact, for the second period, the error obtained with the PD+ controller has an increment of about 184% greater than the one obtained with adaptive control.

5. Conclusion

Adaptive control was developed for the trajectory tracking control of a 3-DOF parallel manipulator. The controller took advantages of a simplified relevant parameters model that is cost-effective real-time. Four study cases of adaptive control were evaluated considering the on-line identification of: the rigid body parameters, the friction parameters, the actuator dynamics, and a combination of the former parameters. A novel experimental setup was proposed that evaluates the performance of the adaptive controller: a payload is placed onto the platform while the robot is executing a task. The adaptive control and a fixed PD+ control scheme were implemented modularity in OROCOS environment. Simulations and experiments showed that adaptive control outperform, as regard to trajectory tracking accuracy, the PD+ control when a payload is added onto the platform. The on-line update of the relevant parameters takes into account for an unknown payload and changes in friction coefficients, thus, overcoming the limitation of the PD+ that relies on fixed parameters.

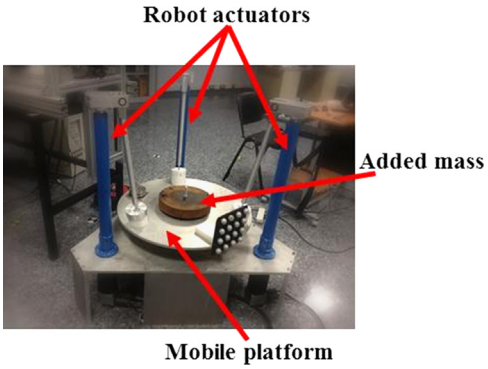

Fig. 10. Parallel robot used in this experiment, with the 30 kg added mass over the platform.

Table 4
Error values (m).

Period	$\frac{\sum_{j=i}^{n}\sum_{i=1}^{\mathrm{DOF}_j} abs(e_{i,j})}{n\mathrm{DOF}}$		$\sqrt{\frac{\sum_{j=i}^{n}\sum_{i=1}^{\mathrm{DOF}_j}(e_{i,j})^2}{n\mathrm{DOF}}}$	
	PD+ C.	Adaptive C.	PD+ C.	Adaptive C.
1	0.000504	0.000337	0.000622	0.000412
2	0.001432	0.000378	0.001541	0.000446

Acknowledgments

This work was partially financed by Fondo Nacional de Ciencia y Tecnología e Innovación (FONACIT, Venezuela), CDCHT-ULA Grant I-1286-11-02-B and Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (FEDER-CICYT, Spain) under the research projects DPI2011-28507-C02-01 and DPI2010-20814-C02-02.

APPENDIX A. Controller stability analysis

A.1. PD+ controller

The motion controller proposed by [26] has the following control law:

$$\tau_e = M(q,\theta)\ddot{q}_d + C(q,\dot{q},\theta)\dot{q}_d + G(q,\theta) - K_p e - K_d \dot{e} \quad (A.1)$$

where K_p and K_d are constant and positive definite diagonal matrices.

Proposition 1. *The control input of Eq. (A.1) stabilizes the robot globally and asymptotically at the equilibrium point* $(e,\dot{e}) = (0,0)$.

Proof. We selected as energy function:

$$H_1(e,\dot{e}) = \frac{1}{2}\dot{e}^T M(q,\theta)\dot{e} + \frac{1}{2}e^T K_p e \tag{A.2}$$

The time derivative of (A.2) is given by

$$\dot{H}_1(e,\dot{e}) = \dot{e}^T M(q,\theta)\ddot{e} + \frac{1}{2}\dot{e}^T \dot{M}(q,\theta)\dot{e} + \dot{e}^T K_p e \tag{A.3}$$

Substituting (A.1) into the robot dynamics leads to the following error equation:

$$M(q,\theta)\ddot{e} + C(q,\dot{q},\theta)\dot{e} + K_p e + K_d\dot{e} = 0 \tag{A.4}$$

Hence, plugging (A.4) into (A.3) yields

$$\dot{H}_1(e,\dot{e}) = -\dot{e}^T(C(q,\dot{q},\theta)\dot{e} + K_p e + K_d\dot{e}) + \frac{1}{2}\dot{e}^T\dot{M}(q,\theta)\dot{e} + \dot{e}^T K_p e \tag{A.5}$$

Then Eq. (A.5) becomes

$$\dot{H}_1(e,\dot{e}) = \dot{e}^T\left(\frac{1}{2}\dot{M}(q,\theta\right) - C(q,\dot{q},\theta))\dot{e} - \dot{e}^T K_d\dot{e} \tag{A.6}$$

Because $\dot{M}(q,\theta) - 2C(q,\dot{q},\theta)$ is a skew-symmetric matrix,

$$\dot{H}_1 = -\dot{e}^T K_d\dot{e} \tag{A.7}$$

This time-derivative is only negative semi-definite. In the case of position control, LaSalle's invariance theorem is applied in order to establish asymptotic stability of the closed-loop system. Unfortunately, this theorem is only valid for autonomous systems. However, using Matrosov's stability theorem of, asymptotically stabilizes the robot system at $(e,\dot{e}) = (0,0)$ for arbitrary $K_p = K_p^T > 0, \quad K_d = K_d^T > 0$.

A.2. Adaptive controller

Bayard and Wen (1988) proposed an adaptive robot motion controller

$$\tau_e = M_0(q)\ddot{q}_d + C_0(q,\dot{q}_d)\dot{q}_d + G_0(q) + Y(q,\dot{q}_d,\dot{q}_d,\ddot{q}_d)\hat{\theta} - K_d\dot{e} - K_p e \tag{A.8}$$

$$\frac{d}{dt}\{\hat{\theta}(t)\} = -\Gamma_0 Y^T(q,\dot{q}_d,\dot{q}_d,\ddot{q}_d)s_1 \tag{A.9}$$

where $s_1 = \dot{e} + \Lambda_1 e \quad$ with $\Lambda_1 = \lambda_1 I,\ y\lambda_1 > 0$.

Proposition 2. *The closed-loop system obtained with (A.8) and (A.9) is convergent, that is the tracking error asymptotically converges to zero and all internal signals remain bounded if*

$$K_{p,m} > \lambda_1 C_M V_M \tag{A.10}$$

$$K_{d,m} > \lambda_1 M_M + 2C_M V_M + \lambda_1 C_M\left(\frac{H_2(e(0),\dot{e}(0),\tilde{\theta}(0))}{K_{p,m} + \lambda_1 K_{d,m} - \lambda_1^2 M_M}\right)^{1/2} \tag{A.11}$$

where $H_2(e,\dot{e},\tilde{\theta})$ is defined in (A.13) below.

Proof. The controller (A.8) results in the closed-loop error system

$$M(q,\theta)\ddot{e} + C(q,\dot{q},\theta)\dot{e} + C(q,\dot{q}_d,\theta)\dot{e} + K_d\dot{e} + K_p e = Y(q,\dot{q}_d,\dot{q}_d,\ddot{q}_d)\tilde{\theta} \tag{A.12}$$

Consider the function

$$H_2(.) = \frac{1}{2}\dot{e}^T M(q,\theta)\dot{e} + \lambda_1\dot{e}^T M(q,\theta)e + \frac{1}{2}e^T(K_p + \lambda_1 K_d)e + \frac{1}{2}\tilde{\theta}^T\Gamma_0^{-1}\tilde{\theta} \tag{A.13}$$

This function contains bilinear cross terms in the position and velocity error, and it can be shown that (A.11) is a sufficient condition for (A.13) to be a strict Lyapunov function candidate. The time-derivative of (A.13) satisfies

$$\begin{aligned}\dot{H}_2(e,\dot{e},\tilde{\theta}) = {} & -\dot{e}^T(K_d - \lambda_1 M(q,\theta))\dot{e} - e^T(\lambda_1 K_p e) + \dot{e}^T C(q,\dot{e},\theta)(\lambda_1 e) \\ & + \dot{e}^T C(q,\dot{q}_d,\theta)(\lambda_1 e) - s_1^T C(q,\dot{q}_d,\theta)\dot{e}\end{aligned} \tag{A.14}$$

Consequently, taking into consideration that the matrices $M(q,\theta)$, $C(q,\dot{q},\theta)$ and $G(q,\theta)$ are bounded with respect to q, and the desired trajectory signals $\dot{q}_d(t)$ and $\ddot{q}_d(t)$ are also bounded

$$\begin{aligned}\dot{H}_2(e,\dot{e},\tilde{\theta}) \le {} & -(K_{d,m} - \lambda_1 M_M)||\dot{e}||^2 - \lambda_1^{-1}K_{p,m}||\lambda_1 e||^2 \\ & + C_M(||\dot{e}||^2||\lambda_1 e|| + 2V_M||\dot{e}||||\lambda_1 e|| + V_M||\dot{e}||^2)\end{aligned} \tag{A.15}$$

where $V_M = \sup||\dot{q}_d(t)||$ and $A_M = \sup||\ddot{q}_d(t)||$

This can be written more conveniently by rewriting the bilinear cross term as the *sum of perfect squares*, that is

$$2||e||||\lambda_1 e|| = -(\alpha||\dot{e}|| - \frac{1}{\alpha}||\lambda_1 e||)^2 + \alpha^2||\dot{e}||^2 + \frac{1}{\alpha^2}||\lambda_1 e||^2,\ \alpha \neq 0 \tag{A.16}$$

So (A.14) can be changed into ($\alpha \equiv 1$)

$$\begin{aligned}\dot{H}_2(e,\dot{e},\tilde{\theta}) \le {} & -(K_{d,m} - \lambda_1 M_M - 2C_M V_M)\|\dot{e}\|^2 \\ & -(\lambda_1^{-1}K_{p,m} - C_M V_M)||\lambda_1 e||^2 + C_M||\dot{e}||^2||\lambda_1 e||\end{aligned} \tag{A.17}$$

(A.17) contains two second-order terms and one third-order term in e and $\dot{e}$. Hence, $\dot{H}_2(e,\dot{e},\tilde{\theta})$ can be guaranteed to be negative semi-definite in the error state only if the second-order terms over bound the third-order one. This naturally implies *local* stability properties, that is the initial error state $[e^T(0) \quad \dot{e}^T(0)\ \tilde{\theta}^T(0)]^T$ has to start within a certain region. To complete the proof, a local stability result (the so-called *β-ball* lemma) can be employed. According to this lemma, $\dot{H}_2(e,\dot{e},\tilde{\theta})$ is negative semi-definite in e and $\dot{e}$ if the controller gains K_p and K_d satisfy (A.10) and (A.11). Application of Barbalat's lemma completes the proof.

References

[1] Dalvand MM, Shirinzadeh B. Motion control analysis of a parallel robot assisted minimally invasive surgery/microsurgery system (PRAMiSS). Robot Compu-Integr Manuf 2012;29:318–27.
[2] Pisla D, Gherman B, Vaida C, Suciu M, Plitea N. An active hybrid parallel robot for minimally invasivesurgery. Robot Compu-Integr Manuf 2013;29:203–21.
[3] Merlet J. Still a long way to go on the road for parallel mechanisms. In: Proceedings of the ASME 27th biennial mechanisms and robotics conference; 2002.
[4] Paccot F, Andreff N, Martinet P. A review on the dynamic control of parallel kinematic machines: theory and experiments. Int J Mach Tools Manuf 2009;28:395–416.
[5] Khalil W, Dombre E. Modeling, identification and control of robots; 2004.
[6] Zhiyong Y, Tian H. A new method for tuning PID parameters of a 3 DOF reconfigurable parallel kinematic machine; 2004.
[7] Li Q, Ouyang P. Integrated design and PD control of high-speed closed-loop mechanisms. J. Dyn. Syst. Meas. Contr. 2002;124:522–8.
[8] Stan S, Balan R, Maties V, Rad C. Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot. Mechanika 2009;1:62–6.
[9] Yang C, Huang Q, Han J. Decoupling control for spatial six-degree-of-freedom electro-hydraulic parallel robot. Robot Compu-Integr Manuf 2012;9:14–23.
[10] Yen P, Lai C. Dynamic modeling and control of a 3-DOF Cartesian parallel manipulator. Mechatronics 2009;19:390–8.
[11] Shang W, Cong S. Nonlinear computed torque control for a high-speed planar parallel manipulator. Mechatronics 2009;19:987–92.
[12] Jiang S. Modeling, identification and control of a redundant planar 2-DOF parallel manipulator. Int. J. Control Autom. Syst. 2007;5:559–69.
[13] Wu J, Wang J, You Z. An overview of dynamic parameter identification of robots. Robot Compu-Integr Manuf 2010;26:414–9.
[14] Díaz-Rodríguez M, Mata V, Farhat N, Provenzano S. Identifiability of the dynamic parameters of a class of parallel robots in the presence of measurement noise and modeling discrepancy. Mech Based Des Struct Mach 2008;36:478–98.
[15] Kim H, Cho Y, Lee K. Robust nonlinear task space control for 6 DOF parallel manipulator. Automatica 2005;41:1591–600.
[16] Abdellatif H, Kotlarski J, Ortmaier T, Heimann B. Practical Model-based and Robust Control of Parallel Manipulators Using Passivity and Sliding Mode Theory, Robotics 2010 Current and Future Challenges.

[17] Honegger M, Codourey A. Adaptive control of the Hexaglide, a 6 DOF parallel manipulator. Robot Autom 1997;1:543–8.
[18] Shang W, Cong S. Nonlinear adaptive task space control for a 2-DOF redundantly actuated parallel manipulator. Nonlinear Dyn 2010;59:61–72.
[19] Shang W, Cong S, Ge Y. Adaptive computed torque control for a parallel manipulator with redundant actuation. Robotica 2012;30:457–66.
[20] Díaz-Rodríguez M, Mata V, Valera Á, Page Á. A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. Mech Mach 2010;45:1337–56.
[21] Diaz-Rodriguez M, Valera A, Mata V, Valles M. Model-based control of a 3-DOF parallel robot based on identified relevant parameters. IEEE/ASME Trans Mechatron 2013;18:1737–44.
[22] Paden B, Panja R. Globally asymptotically stable "PD+"controller for robot manipulators. Int J Control 1988;47:1697–712.
[23] Vallés M, Díaz-Rodríguez M, Valera Á, Mata V, Page Á. Mechatronic development and dynamic control of a 3-DOF parallel manipulator. Mech Based Des Struct Mach 2012;40:434–52.
[24] de Jalon J, Bayo E. Kinemat Dyn Simul Multibody Syst. The Real Time ChallengeSpringer–Verlag, New York: Mechanical Engineering Series; 1993.
[25] Li Y, Xu Q. Kinematic analysis of a 3-PRS parallel manipulator. Robot Comput-Integr Manuf 1988;23:395–408.
[26] Bayard D, Wen J. New class of control laws for robotic manipulators Part 2. Adaptive case. Int J Control 1988;47:1387–406.
[27] Bruyninckx H. Open Robot Control Software: the OROCOS project. Robot Autom 2001;3:2523–8.
[28] Alonso D, Pastor J, Sánchez P, Álvarez B. Generación automática de software para sistemas de tiempo real: un enfoque basado en componentes. Model/Framew, De Autom E 2012;9:170–81.