Adaptive Control of a 3-DOF Parallel Manipulator Considering Payload Handling and Relevant Parameter Identification

J. Cazalilla¹, M. Vallés¹, V. Mata², M. Díaz-Rodríguez³, A. Valera¹

¹Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Valencia, CO 46022 Spain. (e-mail: {jcazalilla, mvalles, giuprog}@ai2.upv.es).
²Centro de Investigación de Tecnología de Vehículos, Universidad Politécnica de Valencia, Valencia, CO 46022 Spain. (e-mail: vmata@mcm.upv.es).
³Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida, CO 5101 Venezuela (e-mail: dmiguel@ula.ve).

Abstract

A model-based control system may produce a substantial increase in the overall performance of parallel robots, thus allowing less expensive manufacturing. Although dynamic parameter identification is an experimental technique that significantly improves data feeding dynamic models, some difficulties arise: the effect of the parallel robot’s unmodeled dynamics, the variation of some dynamic parameters over time (e.g. friction parameters) and, even more importantly from an operational point of view, the payload, which is normally not included in the identification process for practical reasons.

Adaptive control techniques offer the possibility to deal with these uncertainties. Therefore, an adaptive control in joint space has been developed in this paper in order to control a low cost 3-DOF parallel manipulator. The novelty of the proposed controller is that it considers, as a starting point, simplified dynamic models based solely on experimentally identified relevant dynamic parameters. In addition, an adaptive strategy has been applied to various scenarios where it is assumed that the uncertainties affect: 1) rigid body parameters, 2) friction parameters, 3) actuator dynamics, and 4) all the aforementioned. Simulation and experiments on an actual parallel robot were conducted to evaluate the performance of the controller, which was
implemented in a modular way using Open RObot Control Software (OROCOS). Finally, experiments which consider the placement of a payload onto the platform were conducted over the actual prototype. Results demonstrate that the proposed adaptive schemes improve position-tracking performances when a payload is added to the platform.

**Keywords** — Parallel robot, model-based control, adaptive control, dynamic parameter identification.

1. INTRODUCTION

Parallel Manipulators (PMs) have been a very active topic of research that started in academia, and have since been extended to real world applications such as medical applications, e.g. surgery systems [1], [2]. The use of PMs is due to the advantages they hold over their serial counterparts in terms of high stiffness, accuracy, speed, and payload handling. However, as explained in [3], these advantages can only be qualified as potential. To reach these theoretical performances, PMs still require improvements in design, modeling and control, which are the aspects that should be developed for any given robot. Regarding the modeling of PMs, although there is a vast literature dealing with kinematics and dynamic modeling, the control of PMs is a field where great potential remains for improvement in the accuracy of a robot’s performance [4].

In most industrial robots, PD or PID controls are used, (see [5-7] among others), as well as the implementation of control strategies based on fuzzy logic [8]. Given that PD or PID control schemes do not consider the dynamic model of the robot, these strategies are less accurate than model-based control when fast and accurate motion is required [5]. Computed Torque Control (CTC) uses the robot dynamics in the feedback loop for linearization and decoupling system dynamics. This control strategy has been implemented for PMs in [4], [9]-[11]. However, CTC
controllers need the dynamic model to be calculated in real time, which increases the computational burden on the control system. In addition to the CTC controller, feedforward with PD control or Augmented PD control (APD) has been developed for PMs in [12]. APD control improves the controller trajectory by compensating the system dynamics, which are computed offline, and as a result, computational complexity of the controller is lower than CTC. However, these control schemes use the desired robot trajectory to compute the dynamic model, thus ADP control does not take into account changes in the actual system state variables.

The aforementioned control strategies require accurate values of dynamic model parameters. Generally, the dynamic parameters are determined through experimental parameter identification techniques [13]. Nevertheless, due to the topology of PMs, not all the links of the robot can move with enough excitation in order to properly identify its dynamics parameters, especially when the PM has been constrained with less than 6 DOF [14]. In addition, the payload may be unknown and some of the system parameters are time-varying (e.g. friction parameters). The disadvantage of such uncertainties means that model-based control could lose its performance potential and therefore controllers that take into account time variant parameters must be considered.

One approach for dealing with time-varying uncertainties is robust control. In [15] a nonlinear task, space control has been applied to 6 DOF PM. However, robust control has difficulty in tuning the control parameters, with some practical aspects of its implementation discussed in [16]. Another approach for dealing with time-varying uncertainties is adaptive control, especially if the payload is considered in the model. This control strategy has been developed for a 6 DOF PM in [17], where a dynamic feedforward compensator was implemented together with a linear feedback controller. The parameters of the dynamic model were identified through a nonlinear adaptive control. However, linear dependencies in model parameters, base parameters [5], were
not studied, and the simplification of the model is carried out without considering the identifiability of the dynamic parameters.

Adaptive computed torque control and nonlinear adaptive control have been developed [18-19], where the controller is implemented on a planar PM. The controllers were designed in task space, which requires extra sensors. Thus, task space coordinates were found from the actuated joint measurement through forward kinematics. In order to determine the remaining generalized coordinates of the dynamic model, inverse kinematics was performed. The computational burden of this approach is high. In addition, the authors have not considered the simplification of the dynamic model.

In the aforementioned adaptive controllers, the following topics have not been considered: 1) Applying the adaptive schemes solely on a subset of the identified parameters, e.g. only rigid body parameters, 2) An adaptive model considering a simplified model based on a set of properly identified parameters (relevant parameters [20]), and 3) Real experiments where the robot handles a payload. Motivated by these three points, an adaptive control scheme was developed in this paper for a low-cost 3-DOF spatial PM. The dynamics parameters identification methodology proposed in [20] was applied to develop the model for the proposed control scheme. The identified model is called the relevant parameter model and is a simplified one that contains physically feasible parameters, which are properly identified in terms of the standard deviation of parameters. It is worth noting that for the considered robot, the relevant parameters model can be computed in real time [21]. The relevant parameter model includes 12 parameters: 3 rigid bodies, 3 actuator dynamics and 6 frictions parameters. So as to keep the model linear in parameters, the friction in the actuated joints was modeled as a Coulomb plus viscous friction model. Different adaptive schemes have been developed. Each scheme differs in
terms of the subset of relevant parameters for online identification. The first case only considers
the online calculation of the rigid body parameters. As will be seen in the modeling section, rigid
body parameters included the platform mass and therefore this model could adapt the parameters
to compensate for an unknown payload. The second case explores friction parameters, the third
considers actuator dynamics uncertainties and the fourth the variation of all the relevant
parameters.

Based on these cases, different adaptive control schemes and a fixed passivity-based controller
(PD+) [22] have been implemented by means of real-time middleware OROCOS. The tracking
trajectory performances of the adaptive controllers and the PD+ controller were compared
through simulations and experiments conducted over an actual 3-DOF parallel robot. The
experiments considered a payload variation where a robot followed a specific trajectory.

The results demonstrate that in comparison to the PD+ controller, the proposed controller can
obtain a better trajectory tracking accuracy when a payload is added to the platform.

This paper is organized as follows: Section II shows the PM design, while Section III deals
with the model-based control schemes. Section IV presents some experiments conducted on
simulated and actual parallel manipulators, showing the results of the different control schemes.
Finally, some conclusions are presented in Section V.

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.

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 $i=1..9$). 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 $P_i$ 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,

\begin{align}
\Psi_1(q_1, q_2, q_6, q_7) &= \left\| \vec{r}_{A_1B_1} + \vec{r}_{B_1P_1} \right\| - \left( \vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2} \right) - l_m = 0 \\
\Psi_2(q_1, q_2, q_8, q_9) &= \left\| \vec{r}_{A_1B_1} + \vec{r}_{B_1P_1} \right\| - \left( \vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3} \right) - l_m = 0 \\
\Psi_3(q_6, q_7, q_8, q_9) &= \left\| \vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3} \right\| - \left( \vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2} \right) - l_m = 0
\end{align}

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 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],

\begin{equation}
\vec{r}_{A_1P_i} - q_1 \cdot \vec{u}_{A_1B_1} = l_m \cdot \vec{u}_{B_1P_1}
\end{equation}

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,
\[ M(q, \Phi) \cdot \ddot{q} + C(q, \dot{q}, \Phi) \cdot \dot{q} + G(q, \Phi) = \tau \]  \hspace{1cm} (5)

In (5), \( M \) represents the system mass matrix, \( C \) is the matrix grouping the centrifugal and Coriolis terms, \( G \) is the vector corresponding to gravitational terms and \( \tau \) 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 \( q = [q_1 \quad q_6 \quad q_8]^T \).

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 \( i \)-th joint has been modeled as follows,

\[ \tau_{f_i} = -\left( F_{C_i} \cdot \text{sign}(\dot{q_i}) + F_{v_i} \cdot \dot{q_i} \right) \]  \hspace{1cm} (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. Equation (5) can be rewritten (see [5], [13], [20]) as follows,

\[ \begin{bmatrix} K_{rb} & K_r & K_f \end{bmatrix} \begin{bmatrix} \Phi_{rb} \\ \Phi_r \\ \Phi_f \end{bmatrix} = \tau \]  \hspace{1cm} (7)

In (7), the vectors \( \Phi_{rb}, \Phi_r, \Phi_f \) group the rigid body, rotor and friction parameters. \( K_i \) is the part of the regressor 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 to 3). In Table 3, \( l_m \) is the characteristic length of the mobile platform, \( l_r \) is the length of the coupler links connecting the platform to the linear actuators, \( m_i, m_{x_i}, m_{y_i}, m_{z_i}, I_{xx_i}, I_{xy_i}, I_{xz_i}, I_{yy_i}, I_{yz_i}, I_{zz_i} \) are the mass, the first and second moments of inertia of the \( i \)-th 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].

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.

\[
\Omega_1 = m y_3 - \sin(2/3\pi) \cdot \sum_{i=1}^{5} m_i \\
\Omega_2 = \sum_{i=1}^{7} m_i \\
\Omega_3 = m x_7 + l_r \cdot \sum_{i=6}^{7} m_i
\]

(8) (9) (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

Equation (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 \([p x 1]\) vector \( \Phi \) that rewrites the system dynamics linearly in parameters.
Therefore, the following holds,

\[ M(q, \Phi) \dot{q} + C(q, \ddot{q}, \Phi) \ddot{q} + G(q, \Phi) = M_0(q) \dot{q} + C_0(q, \ddot{q}) \ddot{q} + G_0(q) + Y(q, \dddot{q}) \Phi \quad (11) \]

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

\[
C(q, \dddot{q}) \dddot{q} = \begin{bmatrix} F_{v1} \dddot{q}_1 + F_{c1} \text{sign}(q_1) \\ F_{v2} \dddot{q}_6 + F_{c2} \text{sign}(q_6) \\ F_{v3} \dddot{q}_8 + F_{c3} \text{sign}(q_8) \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dddot{q}) & C_{12}(q, \dddot{q}) & C_{13}(q, \dddot{q}) \\ C_{21}(q, \dddot{q}) & C_{22}(q, \dddot{q}) & C_{23}(q, \dddot{q}) \\ C_{31}(q, \dddot{q}) & C_{32}(q, \dddot{q}) & C_{33}(q, \dddot{q}) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad (12)
\]

\[
M(\dddot{q}) \dddot{q} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \dddot{q} + \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad (13)
\]

\[
G(\dddot{q}) = g \begin{bmatrix} G_{11}(\dddot{q}) & G_{12}(\dddot{q}) & G_{13}(\dddot{q}) \\ G_{21}(\dddot{q}) & G_{22}(\dddot{q}) & G_{23}(\dddot{q}) \\ G_{31}(\dddot{q}) & G_{32}(\dddot{q}) & G_{33}(\dddot{q}) \end{bmatrix} \Omega \quad (14)
\]

In addition, \( M_n(), C_0(), G_0() \) represent the known part of the system dynamics, and \( Y(q, \dddot{q}) \) is a regressor matrix with dimensions \([n \times p]\) 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,

\[ M_0(q) \dot{q} + C_0(q, \dddot{q}) \dddot{q} + G_0(q) + Y(q, \dddot{q}) \Phi = \tau \quad (15) \]

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,

\[
\tau = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_6 \\ \dot{q}_8 \end{bmatrix} + \begin{bmatrix} F_{v_1} \dot{q}_1 + F_{c_1} \text{sign}(\dot{q}_1) \\ F_{v_2} \dot{q}_6 + F_{c_2} \text{sign}(\dot{q}_6) \\ F_{v_3} \dot{q}_8 + F_{c_3} \text{sign}(\dot{q}_8) \end{bmatrix} + Y_1(q, \dot{q}, \ddot{q}) \ddot{q}
\]

(16)

where,

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

(17)

\[
\ddot{q} = [\Omega_1 \quad \Omega_2 \quad \Omega_3]^T
\]

(18)

3.2. Adaptive Model II

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

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

(19)

with,

\[
Y_2(q, \dot{q}) = \begin{bmatrix} q_1 & 0 & 0 & \text{sign}(q_1) & 0 & 0 \\ 0 & q_6 & 0 & 0 & \text{sign}(q_6) & 0 \\ 0 & 0 & q_8 & 0 & 0 & \text{sign}(q_8) \end{bmatrix}
\]

(20)

\[
\ddot{\theta}_2 = [F_{v_1} \quad F_{v_2} \quad F_{v_3} \quad F_{c_1} \quad F_{c_2} \quad F_{c_3}]^T
\]

(21)
3.3. Adaptive Model III

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

\[
\tau = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} \Omega + \begin{bmatrix} F_{v1} \dot{q}_1 + F_{c1} \text{sign}(\dot{q}_1) \\ F_{v2} \dot{q}_6 + F_{c2} \text{sign}(\dot{q}_6) \\ F_{v3} \dot{q}_8 + F_{c3} \text{sign}(\dot{q}_8) \end{bmatrix}
\]

\[
+ \begin{bmatrix} C_{11}(q, \bar{q}) & C_{12}(q, \bar{q}) & C_{13}(q, \bar{q}) \\ C_{21}(q, \bar{q}) & C_{22}(q, \bar{q}) & C_{23}(q, \bar{q}) \\ C_{31}(q, \bar{q}) & C_{32}(q, \bar{q}) & C_{33}(q, \bar{q}) \end{bmatrix} \Omega + g \begin{bmatrix} G_{11}(q) & G_{12}(q) & G_{13}(q) \\ G_{21}(q) & G_{22}(q) & G_{23}(q) \\ G_{31}(q) & G_{32}(q) & G_{33}(q) \end{bmatrix} \Omega + Y_3(\bar{q})\bar{\theta}_3
\]

with,

\[
Y_3(\bar{q}) = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_6 \\ \ddot{q}_8 \end{bmatrix}
\]

\[
\bar{\theta}_3 = \begin{bmatrix} J_1 & J_2 & J_3 \end{bmatrix}^T
\]

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,

\[
\tau = Y_4(q, \dot{q}, \ddot{q})\bar{\theta}_4
\]

where,

\[
Y_4(q, \dot{q}, \ddot{q}) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} \Omega + \begin{bmatrix} C_{11}(q) & C_{12}(q) & C_{13}(q) \\ C_{21}(q) & C_{22}(q) & C_{23}(q) \\ C_{31}(q) & C_{32}(q) & C_{33}(q) \end{bmatrix} \Omega + g \begin{bmatrix} G_{11}(q) & G_{12}(q) & G_{13}(q) \\ G_{21}(q) & G_{22}(q) & G_{23}(q) \\ G_{31}(q) & G_{32}(q) & G_{33}(q) \end{bmatrix} \Omega + \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_6 \\ \ddot{q}_8 \end{bmatrix}
\]

\[
\bar{\theta}_4 = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & J_1 & J_2 & J_3 & F_{v1} & F_{v2} & F_{v3} & F_{q1} & F_{q2} & F_{q3} \end{bmatrix}^T
\]
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,

\[
\tau_c = M_0(q)\ddot{q}_d + C_0(q,\dot{q}_d)\dot{q}_d + G_0(q) + Y(q,\dot{q}_d,\ddot{q}_d,\dddot{q}_d)\cdot \ddot{\theta} - K_d\ddot{e} - K_p\dddot{e}
\]  

(28)

\[
\frac{d}{dt}\{\dot{\theta}(t)\} = -\Gamma_0 \cdot Y^T(q,\dot{q}_d,\ddot{q}_d,\dddot{q}_d) \cdot \dot{s}_1
\]  

(29)

where \(s_1 = \dot{e} + \Lambda_1\ddot{e}\), with \(\Lambda_1 = \lambda I\), \(\lambda > 0\) and \(M_0(), C_0(), G_0()\) represent the known part of the system dynamics, and \(Y^T()\) is the regressor matrix.

The closed-loop system (5)-(28)-(29) is convergent (see Appendix I); 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,

\[
\tau_c = M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d + G(q) - K_d\ddot{e} - K_p\dddot{e}
\]  

(30)
As in (5), \( M(\cdot), C(\cdot), G(\cdot) \) 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. POSITION CONTROL EXPERIMENTS

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: 3 rigid body, 3 actuator dynamics and 6 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 (equations 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 equations (28) and (29) using the regressor matrix (17). In order to compare the control performance, the PD+ controller (equation
30) has also been simulated. The simulations consider that a mass of 30 kg is placed on the mobile platform $m_3$ at $t=40$ sec. The overall value of the mass changes from 12kg to 42kg.

As can readily be appreciated in Fig. 3b, the error that is 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 (equations (19)-(21)), Adaptive Model III (equations (22)-(24)) and Adaptive Model IV (equations ((25)-(27)). As in the previous case, for these simulations a mass of 30kg has been added to the platform at $t=40$ sec.

Fig. 4 illustrates the absolute error obtained for the two periods: before and after adding the 30kg mass. Fig. 4a shows the error obtained with the Adaptive Model II (red line). Fig. 4b shows the error with the Adaptive Model III (red line), and Fig. 4c presents the error obtained with the Adaptive Model IV (red line). All these figures also illustrate the error obtained with the Adaptive Model I (blue line). As can be seen in Fig. 4a and Fig. 4b, the controllers based on Adaptive Model II and Model III present a mean error that is twice 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. Results of experiments 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 Orocos (Open Robot COntrl Software). These controllers have been used and analyzed with the actual parallel robot presented in Section II.

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 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 equation (17). The Calc_theta module calculates the parameter estimation of equation (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.

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 seconds (between $t = 85$ and $t = 93$ sec). This time allows a load of 30 kg to be placed onto the robot platform. Fig. 7 illustrates the absolute error values, and the control action provided by the adaptive controller is shown in Fig. 8. Fig. 9 shows the real robot with the added mass.

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 30kg). 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

An adaptive control scheme in joint space has been proposed for the trajectory tracking control of a low cost 3-DOF parallel manipulator. The proposed controller takes advantage of a simplified dynamic model based on a set of identified relevant parameters. This modeling strategy allowed a dynamic model to be obtained, which can be computed in real time without losing the estimation of the robot’s dynamic behavior. In addition, the adaptive scheme can account for an unknown payload if it is placed on the center of the mobile platform. The adaptive controller was developed for different adaptive models: the online identification of the relevant rigid body parameters, the identification of friction parameters, the identification of the actuator dynamics, and the identification of all the aforementioned. Simulation and real experiments were conducted to evaluate the performance of the controller. For comparison purposes, a fixed PD+ passivity-based controller was also implemented. The controllers were implemented in a modular way using Orocos, real-time middleware. Real experiments were conducted considering the case when a payload is placed onto the platform. To the best of the authors’ knowledge, to date this kind of experiment has not been conducted or shown. The
results demonstrate that when comparing the PD+ controller and the adaptive controller, the latter obtains better trajectory tracking accuracy when a payload is added to the platform.

REFERENCES

APPENDIX I. CONTROLLER STABILITY ANALYSIS

I.1 PD+ CONTROLLER

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

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

Where $K_p$ and $K_d$ are constant and positive definite diagonal matrices.

Proposition 1

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

Proof.-

We selected as energy function:

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

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

$$H_1(e, \dot{e}) = e^T M(q, \theta) \dot{e} + \frac{1}{2} e^T \dot{M}(q, \theta) \dot{e} + e^T K_p \dot{e}$$  \hspace{1cm} (A.3)

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

$$M(q, \theta) \ddot{e} + C(q, \dot{q}, \theta) \dot{e} + K_p e + K_d \dot{e} = 0$$  \hspace{1cm} (A.4)

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

$$H_1(e, \dot{e}) = -e^T (C(q, \dot{q}, \theta)e + K_p e + K_d \dot{e}) + \frac{1}{2} e^T \dot{M}(q, \theta) e + e^T K_p \dot{e}$$  \hspace{1cm} (A.5)

Then equation (A.5) becomes

$$H_1(e, \dot{e}) = e^T (\frac{1}{2} \dot{M}(q, \theta) - C(q, \dot{q}, \theta)) e - e^T K_d \dot{e}$$  \hspace{1cm} (A.6)

Because $\dot{M}(q, \theta) - 2C(q, \dot{q}, \theta)$ is a skew-symmetric matrix,
\[ H_1 = -\dot{e}^T K_d \dot{e} \]

(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 \(K_p = K_p^T > 0, K_d = K_d^T > 0\).

1.2 ADAPTIVE CONTROLLER

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

\[
\tau_e = M_0(q)\dot{q}_d + C_0(q, \dot{q}_d)\dot{q}_d + G_0(q) + Y(q, \dot{q}_d, \ddot{q}_d, \dddot{q}_d)\dot{\theta} - K_d \dot{e} - K_p e
\]

(A.8)

\[
\frac{d}{dt} \{ \dot{\theta}(t) \} = -\Gamma \dot{Y}^T (q, \dot{q}_d, \ddot{q}_d, \dddot{q}_d) s_1
\]

(A.9)

where \( s_1 = \dot{e} + \Lambda_1 e \), with \( \Lambda_1 = \lambda_1 I \), \( \lambda_1 > 0 \).

**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

\[
K_{p,m} > \lambda_1 C_M V_M
\]

(A.10)

\[
K_{d,m} > \lambda_1 M_M + 2C_M V_M + \lambda_1 C_M \left( \frac{H_2(e(0), \dot{e}(0), \ddot{\theta}(0))}{K_{p,m} + \lambda_1 K_{d,m} - \lambda_1^2 M_M} \right)^{\frac{1}{2}}
\]

(A.11)

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

**Proof.**

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

\[
M(q, \theta) \ddot{e} + C(q, q, \theta) \dot{e} + C(q, \dot{q}_d, \theta) e + K_d \dddot{e} + K_p e = Y(q, \dot{q}_d, \ddot{q}_d, \dddot{q}_d) \ddot{\theta}
\]

(A.12)
Consider the function

$$H_2(\epsilon) = \frac{1}{2} \epsilon^T M(q, \theta) \epsilon + \lambda_1 \epsilon^T M(q, \theta) \epsilon + \frac{1}{2} \epsilon^T (K_p + \lambda_1 K_d) \epsilon + \frac{1}{2} \hat{\theta} \hat{\theta}^T \Gamma_0^{-1} \hat{\theta}$$

(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

$$\dot{H}_2(\epsilon, \dot{\epsilon}, \hat{\theta}) = -\epsilon^T (K_d - \lambda_1 M(q, \theta)) \dot{\epsilon} - \epsilon^T (\lambda_1 K_p \epsilon) +$$

$$+ \epsilon^T C(q, \dot{\epsilon}, \theta)(\lambda_1 \epsilon) + \epsilon^T C(q, \dot{q}_d, \theta)(\lambda_1 \epsilon) - s_1^T C(q, \dot{q}_d, \theta) \dot{\epsilon}$$

(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

$$\dot{H}_2(\epsilon, \dot{\epsilon}, \hat{\theta}) \leq - (K_{d,m} - \lambda_1 M_M) \| \dot{\epsilon} \|^2 - \lambda_1^{-1} K_{p,m} \| \lambda_1 \epsilon \|^2 +$$

$$+ C_M (\| \dot{\epsilon} \|^2 \| \lambda_1 \epsilon \|^2 + 2 \| \dot{\epsilon} \| \| \lambda_1 \epsilon \| + V_M \| \dot{\epsilon} \|^2$$

(A.15)

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

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

$$2\| \lambda_1 \epsilon \| = -(\alpha \| \dot{\epsilon} \|^2 - \frac{1}{\alpha} \| \lambda_1 \epsilon \|^2) + \alpha^2 \| \epsilon \|^2 + \frac{1}{\alpha^2} \| \lambda_1 \epsilon \|^2, \alpha \neq 0$$

(A.16)

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

$$\dot{H}_2(\epsilon, \dot{\epsilon}, \hat{\theta}) \leq - (K_{d,m} - \lambda_1 M_M - 2 C_M V_M) \| \dot{\epsilon} \|^2 +$$

$$- (\lambda_1^{-1} K_{p,m} - C_M V_M) \| \lambda_1 \epsilon \|^2 + C_M \| \dot{\epsilon} \| \| \lambda_1 \epsilon \|$$

(A.17)

(A.17) contains two second-order terms and one third-order term in $\epsilon$ and $\dot{\epsilon}$. Hence, $H_2(\epsilon, \dot{\epsilon}, \hat{\theta})$ 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 \textit{local} stability properties, that is the initial error state $\begin{bmatrix} e^T (0) & e^T (0) & \theta^T (0) \end{bmatrix}^T$ has to start within a certain region. To complete the proof, a local stability result (the so-called \textit{\beta-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.