RAAD 2013

22nd International Workshop on Robotics in Alpe-Adria-Danube Region

Portorož, Slovenia

PROCEEDINGS

September 11th – 13th, 2013

Jozef Stefan Institute
Ljubljana, Slovenia

RAAD 2013

22nd International Workshop

on

Robotics in Alpe-Adria-Danube Region

PROCEEDINGS

September 11 – 13, 2013
Portorož, Slovenia

EDITED BY

Bojan Nemec and Leon Žlajpah

PUBLISHED BY

Jožef Stefan Institute

Ljubljana, 2013

Editors:
Bojan Nemec, Leon Žlajpah Jožef Stefan Institute, Ljubljana

Publisher: Jožef Stefan Institute, Ljubljana

Printing: Infokart d.o.o., Medvode

Impression: 70, 1. edition

CIP -- Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

007.52(082)

INTERNATIONAL Workshop on Robotics in Alpe-Adria-Danube
Region (22 ; 2013 ; Portorož)
Proceedings [Elektronski vir] / 22nd International
Workshop on Robotics in Alpe-Adria-Danube Region [also]
RAAD 2013, September 11-13, Portorož, Slovenia ; edited
by Bojan Nemec and Leon Žlajpah. - 1st ed. - Ljubljana :
Jožef Stefan Institute, 2013

ISBN 978-961-264-064-4

1. Nemec, Bojan

268260608

Bio-inspired robotics

Robot control II

Motion planning

Proceedings of the RAAD 2013
22nd International Workshop on Robotics in Alpe-Adria-Danube Region
September 11-13, 2013, Portorož, Slovenia

Implementation of a 3-DOF Parallel Robot Adaptive Motion Controller

J. Cazalilla[1], M. Vallés[1], V. Mata[2], M. Díaz-Rodríguez[3], A. Valera[1]
[1]*Instituto de Automática e Informática Industrial, Universitat Politècnica de València, Valencia, CO 46022 Spain.*
E-mail: {jcazalilla, mvalles, giuprog}@ai2.upv.es).
[2]*Centro de Investigación de Tecnología de Vehículos, Universitat Politècnica de València, Valencia, CO 46033 Spain, E-mail: vmata@mcm.upv.es.*
[3]*Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida, CO 5101 Venezuela, E-mail: dmiguel@ula.ve.*

Abstract. For fast and accurate motion of a Parallel Manipulator, model-based control needs to be implemented. In general in a model-based controller, exact knowledge of the system dynamics is required. However, the dynamic model has uncertainties not only because of the unmodeled dynamics but also when, for instance, unknown inertial parameters can appear. This kind of uncertainty limits the applicability of model-based controllers. To relax the requirement of exact knowledge, an adaptive controller has been developed. The controller is implemented in a modular way using Orocos, a real-time middleware. The proposed controller is compared with a fixed model passivity-based controller. Both control strategies are tested on a virtual and an actual prototype. From the simulations and experiments, the adaptive controller does not present a loss of accuracy when compared with the fixed controller; moreover, when a payload is handled by the robot, the results show that the adaptive controller improves the trajectory tracking precision.

Keywords. Parallel manipulators, model-based control, adaptive robot control, control applications.

1. Introduction

A Parallel Manipulator (PM) consists of a moving platform connected to a fixed base by at least of two kinematics chains. The end-effector in a PM is attached to the moving platform, so the load is shared by the kinematics chains connecting the moving platform to the fixed base. This fact gives to the PMs high stiffness, high load-carrying capacity and high accuracy. PMs are nowadays an active research field where several prototypes have been developed, for instances: motion simulators, tire-testing machines, flight simulators and medical applications (Stewart, 1965), (Merlet, 2000), (Tsai, 1999), (Li and Xu, 2007), (Chablat, 2003), (Carretero et al., 2000), (Merlet, 2002). In addition, PMs with very high accelerations, such as 200m/s^2 for the PAR4 manipulator (Nabat et al., 2005) or 50m/s^2 for the Urane SX machine tool (Company and Pierrot, 2002) have been proposed.

In spite of the advantages that PMs have over serial robots, the implementation of PMs in real applications is difficult. One of the difficulties lies on the controller design of PMs. Due to the inherent closed-loop constraints, the joints of PMs are tightly coupled and the dynamic characteristics are highly nonlinear (Zhan et al., 2007). The highly coupled dynamics makes it difficult to move a PM along a trajectory accurately and quickly. Moreover, the controller design can be a challenging work, which has aroused the interest many researchers in recent years (Zhan et al., 2007), (Fu and Mills, 2007), (Stan et al., 2009), (Gou et al., 2009), (Abdellatif and Heimann, 2010), (Díaz-Rodriguez et al., 2010).

In this paper, the dynamic controller design problem of a PM is addressed. The controller is implemented on a low-cost three degree-of-freedom (DOF) spatial PM. The robot was developed at Universitat Politècnica de València; its end effector is able to perform two angular rotations (rolling and pitching) and a linear motion (heave). The robot is equipped with an open control unit based-on industrial PC.

For the implementation of the model-based controller, exact knowledge of the system dynamics is required. The dynamic model of the PM implemented into the model-based control is based on a reduced model formulated in the joint space.

The model is obtained by Gibbs-Appell equation and then the model is the reduced to a subset of identified relevant parameters (Díaz-Rodriguez et al., 2010). The relevant parameters considered only those dynamic parameters with have significant influences on the robot dynamics; in addition the identified parameters are physically feasible. Due to this fact, the dynamic model has uncertainties. Moreover, an unknown inertial parameters can appears when a payload mass is grasped by the robot. Particularly, this kind of uncertainty limits the applicability of model-based controllers. To relax the requirement of exact knowledge, an adaptive controller for the PM has been adopted. The chosen control strategy is an adaptive passivity-based controller. One of the useful properties of these passivity-based tracking controllers is that the controller can easily be modified to account for parametric uncertainty of the robot dynamics. The controller takes advantages of the real-time middleware Orocos, allowing the control implementation in a modular way. To verify and validate the proper operation of the adaptive controller, a fixed passivity-based tracking controller has also developed and a comparison of these controllers is presented.

2. The low-cost 3-dof parallel robot

2.1. Physical description of the low-cost PM

As mentioned before, a 3-DOF spatial PM was used for addressing the controller design problem. The robot consists of three kinematics chains; each chain has a PRS configuration (P, R, and S standing for prismatic, revolute, and spherical joint, respectively), The underlying format (P) stand for the actuated joint. The choice of the PRS configuration was guided by the need of developing a low-cost robot with 2 DOF of angular rotation in two axes (rolling and pitching) and 1-DOF translation motion (heave). In (Vallés et al., 2012) a completed 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 motor driving a ball screw (prismatic joints) and a link with is lower part connected by a revolute joint to the ball screw. 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 base platform. The positions of the ball screws at the base are in equilateral triangle configuration. The ball screw transforms the rotational movement of the motor into linear motion.

The motors in each leg are brushless DC servomotor equipped with power amplifiers. The actuators are Aerotech BMS465 AH brushless servomotors. The motors are operated by Aerotech BA10 power amplifiers.

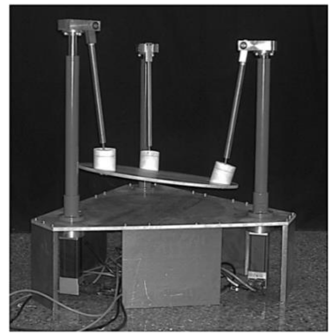


Fig. 1.3-PRS parallel robot implemented

The control system was developed on an industrial PC. The PC-based control system has two main advantages: First, it is a totally open and it gives a powerful platform for programming high level tasks based-on Ubuntu 12.04 operating system. Thus, any controller and/or control technique can be programmed and implemented, such as automatic trajectory generation, control based on external sensing using a force sensor or artificial vision, etc. The second advantage is adopting an industrial PC for the control system the cost of the robot decreases.

2.2. Kinematic model

For control purpose both direct and inverse kinematics problem has to be solved. Given the actuators' linear motions, the direct kinematics of a PM consists of finding the roll (γ) and pitch (β) angles and the heave (z). The kinematic model is established by means of Denavit-Hartenbert (D-H), thus, 9 generalized coordinates are defined for modelling robot kinematics. The location of the coordinate systems is shown in Figure 2.

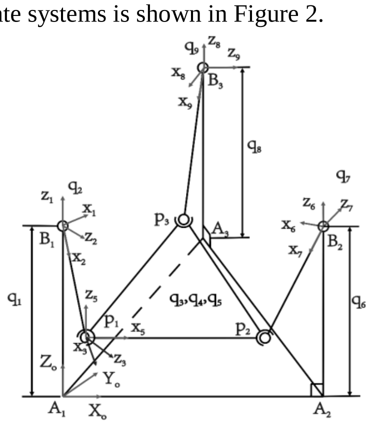


Fig. 2.Location of the coordinate systems

From the figure it can be seen that the length between p_i and p_j is constant and equal to l_m. Thus, applying the geometric approach the kinematics model can be established as follows,

$$f_1(q_1,q_2,q_6,q_7)=\left\|\left(\vec{r}_{A_1B_1}+\vec{r}_{B_1p_1}\right)-\left(\vec{r}_{A_1A_2}+\vec{r}_{A_2B_2}+\vec{r}_{B_2p_2}\right)\right\|-l_m=0 \quad (1)$$

$$f_2(q_1,q_2,q_8,q_9)=\left\|\left(\vec{r}_{A_1B_1}+\vec{r}_{B_1p_1}\right)-\left(\vec{r}_{A_1A_3}+\vec{r}_{A_3B_3}+\vec{r}_{B_3p_3}\right)\right\|-l_m=0 \quad (2)$$

$$f_3(q_6,q_7,q_8,q_9)=\left\|\left(\vec{r}_{A_1A_3}+\vec{r}_{A_3B_3}+\vec{r}_{B_3p_3}\right)-\left(\vec{r}_{A_1A_2}+\vec{r}_{A_2B_2}+\vec{r}_{B_2p_2}\right)\right\|-l_m=0 \quad (3)$$

In the forward kinematics the position of the actuators is known, thus the system of equations (1)-(3) is a nonlinear system with q_2, q_7 and q_9 as unknown. The Newton-Raphson (N-R) numerical method is chosen to solve the nonlinear system. The method converges rather quickly (quadratic convergence) when the initial guess is close to the desired solution (García de Jalón and Bayo, 1994).

The location of the moving platform is defined using a local coordinate system attached to it. The coordinates of the spherical joints of the moving platform are obtained after having found the generalized coordinates of each leg of the robot. These three joints share the plane of the platform, so a local axis X_p is defined as a unit vector $\vec{u}$ with the direction given by p_1 p_2. The axis Z_p is defined by a unit vector $\vec{v}$ perpendicular to the plane defined by pointsp_1, p_2 and p_3. Finally, the axis Y_p(axis $\vec{w}$) is determined by the cross product $\vec{u} \times \vec{v}$. The rotation matrix of the moving platform is given by,

$$^{O}R_p = \left\lfloor \vec{u}^T \quad \vec{v}^T \quad \vec{z}^T \right\rfloor \tag{4}$$

The remaining generalized coordinates (q_3, q_4 and q_5) are found from the rotation matrix.

On the other hand, the inverse kinematics consists of finding the actuators' linear motion given the roll (γ) and pitch (β) angle and the heave (z). Using an X-Y-Z fixed-angle system, the rotational matrix can be defined as,

$$^{O}R_p = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \tag{5}$$

In the above equation, c_* and s_* stand for *cos(*)* and *sin(*)*, respectively. Given γ and β the yaw angle (α) can be found as follows,

$$\alpha = \operatorname{atan2}\left(s_\beta s_\gamma, \left(c_\gamma + c_\beta\right)\right) \tag{6}$$

Having found the angle α, the remaining terms of the rotational matrix can be found. The actuator positions can be found by the following expressions (Tsai, 1999),

$$q_1 = p_z - \frac{1}{2}v_z h - \frac{1}{2}u_z h\sqrt{3} - \frac{1}{2}\Big(-2u_y h^2\sqrt{3}v_y - v_y^2 h^2 - 4p_x^2 + 4p_x u_x h\sqrt{3}u_y - 4u_y^2 h^2 - 4p_y^2 + 4p_y u_y h\sqrt{3} + 4p_y v_y h + 4l_r^2\Big)^{1/2} \tag{7}$$

$$q_6 = p_z + \frac{1}{2}u_z h\sqrt{3} - \frac{1}{2}v_z h - \frac{1}{2}\Big(-4p_x h\sqrt{3}u_x + 2u_x h^2\sqrt{3}u_y - 4p_y u_y h\sqrt{3} + 2u_y^2 h^2\sqrt{3} + 8p_x\sqrt{3}g + 12u_x hg + 4p_x u_y h - 4p_x^2 - 4u_y^2 h^2 - 4p_y^2 - v_y^2 h^2 - 3u_x^2 h^2 + 4l_r^2\Big)^{1/2} \tag{8}$$

$$q_8 = p_z + v_z h - \Big(-3v_y hg - p_x^2 - 2p_x u_y h + g\sqrt{3}p_x - u_y^2 h^2 + hg\sqrt{3}u_y - 3g^2 - p_y^2 - 2p_y v_y h + 3p_y g - v_y^2 h^2 + l_r^2\Big)^{1/2} \tag{9}$$

where $h = l_m/\sqrt{3}$, $g = l_b/\sqrt{3}$, p_x=-hu_y, p_y=-$h(u_x$-$v_y)$, p_z=z and l_b are the lengths between A_iA_j.

2.3. Dynamic model

One of the goals of this paper is to develop an open control architecture allowing the implementation and testing of dynamic control schemes. This kind of dynamic controllers requires describing the equation of motion as follows,

$$M\left(\vec{q},\vec{\Phi}\right)\cdot\ddot{\vec{q}} + \vec{C}\left(\vec{q},\dot{\vec{q}},\vec{\Phi}\right)\cdot\dot{\vec{q}} + \vec{G}\left(\vec{q},\vec{\Phi}\right) = \vec{\tau} \tag{10}$$

From equation (10) it can be seen that the system mass matrix ***M***, the vectors corresponding to the centrifugal and Coriolis forces ***C***, and the gravitational forces ***G*** depend on the dynamic parameters $\vec{\Phi}$ and the external generalized forces $\vec{\tau}$.

In order to identify the dynamic parameters, the model in linear parameter form has to be build first as follows as in (Díaz-Rodriguez et al., 2010),

$$\mathrm{K}\left(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\right)\cdot\vec{\Phi} = \vec{\tau} \tag{11}$$

In equation (11), $\mathrm{K}\left(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}\right)$ is the observation matrix corresponding to the set of generalized coordinates, velocities and accelerations. For this parallel robot, a complete and reduced model can be obtained (Díaz-Rodriguez et al., 2012). The complete model contains all the rigid body dynamic parameters affecting the dynamic behavior of the robot has been obtained. This model consists of the Coulomb and viscous friction parameters presented in Table 1, the rotor and screw dynamics of the robot actuators (Table 2) and the 19 rigid body base parameters shown in Table 3.

However, not even those parameters could always be properly identified in this base parameters model. Thus, the reduced model contains only the relevant parameters obtained through a process which considers the robot's leg symmetries, the statistical significance of the identified parameters, and the physical feasibility of the parameters.

Tab. 1.Friction base parameter for the 3-PRS PM

$\vec{\Phi}_f$	**Base Parameter**	$\vec{\Phi}_f$	**Base Parameter**
1	F_{C_1}	2	F_{V_1}
3	F_{C2}	4	F_{V_2}
5	F_{C_3}	6	F_{V_3}

Tab. 2.Actuators base parameter for the 3-PRS PM

$\vec{\Phi}_a$	**Base Parameter**
1	J_1
3	J_2
5	J_3

Tab. 3.Friction base parameter for the 3-PRS PM

$\vec{\Phi}_f$	Base Parameter	$\vec{\Phi}_f$	Base Parameter
1	$I_{zz2} - l_r^2 \cdot \sum_{i=1}^{2} m_i$	11	$my_3 - \sin(2/3\pi) l_m \cdot \sum_{i=1}^{5} m_i$
2	$mx_2 + l_r \cdot \sum_{i=1}^{2} m_i$	12	mz_3
3	my_2	13	$I_{zz5} - lr^2 \cdot \sum_{i=4}^{5} m_i$
4	$I_{xx3} - (\sin(2/3\pi) l_m)^2 \cdot \sum_{i=1}^{5} m_i$	14	$mx_5 - lr \cdot \sum_{i=4}^{5} m_i$
5	$I_{xy3} + \cos(2/3\pi)\sin(2/3\pi) \cdot l_m{}^2 \cdot \sum_{i=1}^{5} m_i$	15	my_5
6	I_{xz3}	16	$I_{zz7} + lr^2 \cdot \sum_{i=1}^{5} m_i$
7	$I_{yy3} - (\cos(2/3\pi) l_m)^2 \sum_{i=1}^{3} m_i + \sin(2/3\pi) l_m \cdot \sum_{i=4}^{5} m_i$	17	$\sum_{i=1}^{7} m_i$
8	I_{yz3}	18	$mx_7 + lr \cdot \sum_{i=6}^{7} m_i$
9	$I_{zz3} - l_r^2 \cdot \sum_{i=1}^{3} m_i$	19	my_7
10	$mx_3 - \cos(2/3\pi) l_m \cdot \sum_{i=1}^{5} m_i + l_m \sum_{i=4}^{5} m_i$		

The rigid body parameters constituting the reduced model are 11, 17 and 18 in Tab. 3,

$$\Omega_1 = my_3 - \sin(2/3\pi) l_m \cdot \sum_{i=1}^{5} m_i \tag{12}$$

$$\Omega_2 = \sum_{i=1}^{7} m_i \tag{13}$$

$$\Omega_3 = mx_7 + lr \cdot \sum_{i=6}^{7} m_i \tag{14}$$

The equations of robot motion have several fundamental properties that can be exploited to facilitate dynamic controllers design. One of the useful properties is that there is a reparametrization of all unknown parameters into a parameter vector $\vec{\Phi} \in R^p$ that enters linearly in the system dynamics (11). Therefore, the following holds,

$$\begin{aligned} &M(\vec{q},\vec{\Phi}) \cdot \ddot{\vec{q}} + \vec{C}(\vec{q},\dot{\vec{q}},\vec{\Phi}) \cdot \dot{\vec{q}} + \vec{G}(\vec{q},\vec{\Phi}) \equiv \\ &\quad M_0(\vec{q}) \cdot \ddot{\vec{q}} + \vec{C}_0(\vec{q},\dot{\vec{q}}) \cdot \dot{\vec{q}} + \vec{G}_0(\vec{q}) + Y(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\vec{\Phi} \end{aligned} \tag{15}$$

where $M_0(.)$, $\vec{C}_0(.)$, $\vec{G}_0(.)$ represent the know part of system dynamics, and $Y(u,v,w,x)$ is a regressor matrix of dimension $[nxp]$ that contains nonlinear but known functions.

As a consequence of this property, the left hand side of (10) can be written as,

$$M_0(\vec{q}) \cdot \ddot{\vec{q}} + \vec{C}_0(\vec{q},\dot{\vec{q}}) \cdot \dot{\vec{q}} + \vec{G}_0(\vec{q}) + Y(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\vec{\Phi} = \vec{\tau} \tag{16}$$

Because the actual 3-PRS parallel robot reduced dynamic model has 12 parameters (3 of rigid body, 3 of the actuator dynamics and 6 of friction), it can be expressed as,

$$M(\vec{q}) \cdot \ddot{\vec{q}} + \vec{C}(\vec{q},\dot{\vec{q}}) \cdot \dot{\vec{q}} + \vec{G}(\vec{q}) = \vec{\tau} \tag{17}$$

where

$$M(\vec{q}) \cdot \ddot{\vec{q}} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \tag{18}$$

$$\begin{aligned} C(\vec{q},\dot{\vec{q}}) \cdot \dot{\vec{q}} = &\begin{bmatrix} F_{v_1}\dot{q}_1 + F_{c_1}\mathrm{sign}(\dot{q}_1) \\ F_{v_2}\dot{q}_2 + F_{c_2}\mathrm{sign}(\dot{q}_2) \\ F_{v_3}\dot{q}_3 + F_{c_3}\mathrm{sign}(\dot{q}_3) \end{bmatrix} + \\ &\begin{bmatrix} C_{11}(\vec{q},\dot{\vec{q}}) & C_{12}(\vec{q},\dot{\vec{q}}) & C_{13}(\vec{q},\dot{\vec{q}}) \\ C_{21}(\vec{q},\dot{\vec{q}}) & C_{22}(\vec{q},\dot{\vec{q}}) & C_{23}(\vec{q},\dot{\vec{q}}) \\ C_{31}(\vec{q},\dot{\vec{q}}) & C_{32}(\vec{q},\dot{\vec{q}}) & C_{33}(\vec{q},\dot{\vec{q}}) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \end{aligned} \tag{19}$$

$$G(\vec{q}) = g \begin{bmatrix} G_{11}(\vec{q}) & G_{12}(\vec{q}) & G_{13}(\vec{q}) \\ G_{21}(\vec{q}) & G_{22}(\vec{q}) & G_{23}(\vec{q}) \\ G_{31}(\vec{q}) & G_{32}(\vec{q}) & G_{33}(\vec{q}) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \tag{20}$$

Therefore, different combinations can be considered according with the unknown robot parameters. For example, if the rigid body parameters constituting the reduced model are assumed to be unknown, then (16) can be written as,

$$\vec{\tau} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} F_{v_1}\dot{q}_1 + F_{c_1}\mathrm{sign}(\dot{q}_1) \\ F_{v_2}\dot{q}_2 + F_{c_2}\mathrm{sign}(\dot{q}_2) \\ F_{v_3}\dot{q}_3 + F_{c_3}\mathrm{sign}(\dot{q}_3) \end{bmatrix} + Y_1(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})\vec{\theta}_1 \tag{21}$$

where

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

$$\vec{\theta}_1 = [\Omega_1 \quad \Omega_2 \quad \Omega_3]^T \tag{23}$$

3. Adaptive model-based PM control

It is possible to find in the literature different adaptive control schemes that do not suffer from the parameter drift problem. For example, Bayard and Wen have developed in (Bayard and Wen, 1988) a class of adaptive robot motion controllers, but in this work the following one has been developed for the parallel robot:

$$\begin{aligned} \tau_c = &M_0(\vec{q})\ddot{\vec{q}}_d + \vec{C}_0(\vec{q},\dot{\vec{q}}_d)\dot{\vec{q}}_d + \vec{G}_0(\vec{q}) + \\ &Y(\vec{q},\dot{\vec{q}}_d,\dot{\vec{q}}_d,\ddot{\vec{q}}_d)\hat{\vec{\theta}} - K_d\dot{\vec{e}} - K_p\vec{e} \end{aligned} \tag{24}$$

$$\frac{d}{dt}\{\hat{\theta}(t)\} = -\Gamma_0 Y^T(\vec{q},\dot{\vec{q}}_d,\dot{\vec{q}}_d,\ddot{\vec{q}}_d)\vec{s}_1 \tag{25}$$

where $\vec{s}_1 = \dot{\vec{e}} + \Lambda_1\vec{e}$, with $\Lambda_1 = \lambda_1 I$ and $\lambda_1 > 0$.

The close-loop system (17)-(24)-(25) is convergent, that is the tracking error asymptotically converge to zero and all internal signals remain bounded, under a suitable conditions on the controller gains K_p and K_d.

To validate the correct operation of the adaptive control algorithm, several Matlab/Simulink schemes for the parallel robot simulation has been developed. Fig. 3 shows the scheme implemented for the adaptive controller. Simulink block *Y(q,dq,ddq)* implements the regressor matrix of equation (22). *Inertial Terms M0* and *Coriolis Terms C0* blocks implement the know part of the robot dynamics (equations (16) or (21)). Finally, *PD* block implements the proportional-derivative term.

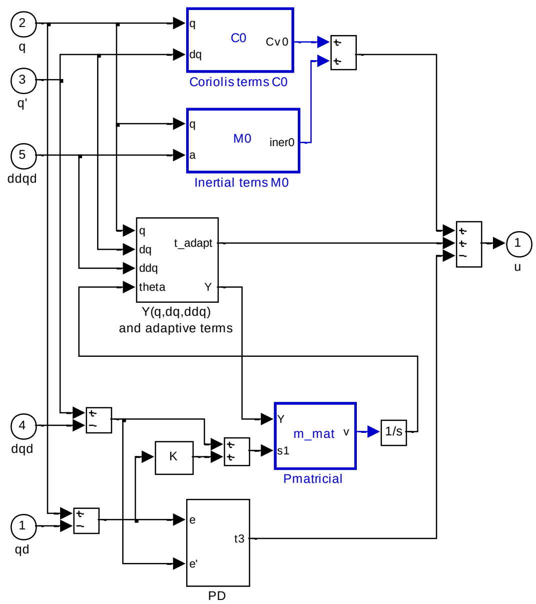


Fig. 3.Adaptive controller simulation scheme

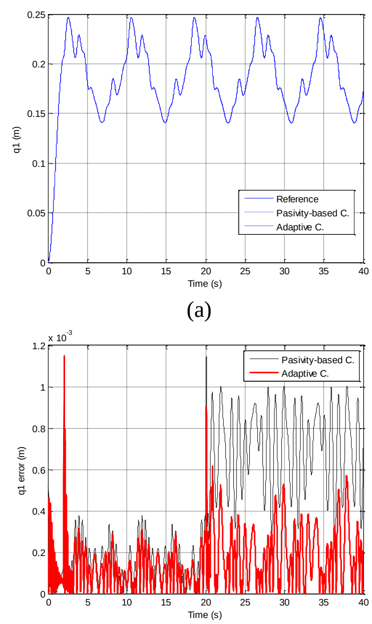
On the other hand, in order to verify the adaptive controller features, a passivity-based trajectory tracking controller has been also implemented. The control low considered is (Paden and Panja, 1988):

$$\tau_c = M(\vec{q})\ddot{\vec{q}}_d + C(\vec{q},\dot{\vec{q}})\dot{\vec{q}}_d + G(\vec{q}) - K_d\dot{\vec{e}} - K_p\vec{e} \qquad (26)$$

This passivity-based controller has been chosen because it has very good robust properties and because its expression is similar to the adaptive controller developed for this work, so it is easy to compare and analyze their characteristics.

As mentioned before, because the reduced robot model has 12 parameters, the adaptive scheme can be developed for different cases, depending on which parameters are considered unknown. In this work, the adaptive controller developed considers only rigid body parameters, so the robot model is expressed using equation (17)-(20).

In this way the following figures show the references and the positions obtained with an adaptive controller and a passivity-based dynamic controller, and the absolute position error. In the simulation it has been considered that at t=20sec a mass of 30 kg was placed in the mobile platform m_3, so it changed from 12kg to 42kg.



(a)

(b)

Fig. 4.Position (a) and absolute error (b) of the first actuated joint.

As it can readily be appreciated in the figure 4, the error that is discussed with both controllers before modifying m_3 is very similar. However, after modifying this mass, the adaptive controller response is much better since the passivity-based controller uses wrong values of some dynamic parameters. Note that very similar results has been obtained for second and third actuated joint.

In addition to the simulations schemes, in this work, the passivity-based and the adaptive controllers described before have been developed in a modular way using a real-time middleware.

In particular, the middleware used is Orocos (Open Robot COntrol Software), which provides the main features of a component-based middleware: creation of an abstraction layer between the operating system and the application layer and communication infrastructure component-based model. The Orocos project consists of a series of libraries and tools, being the most important the *OrocosToolchain*. This tool includes the two main libraries (RTT and OCL) to create the components and control schemes. Using this component-based middleware, we have developed a modular structure for the Paden-Panja control (Eq. 26) and adaptive control (Eq. 24).

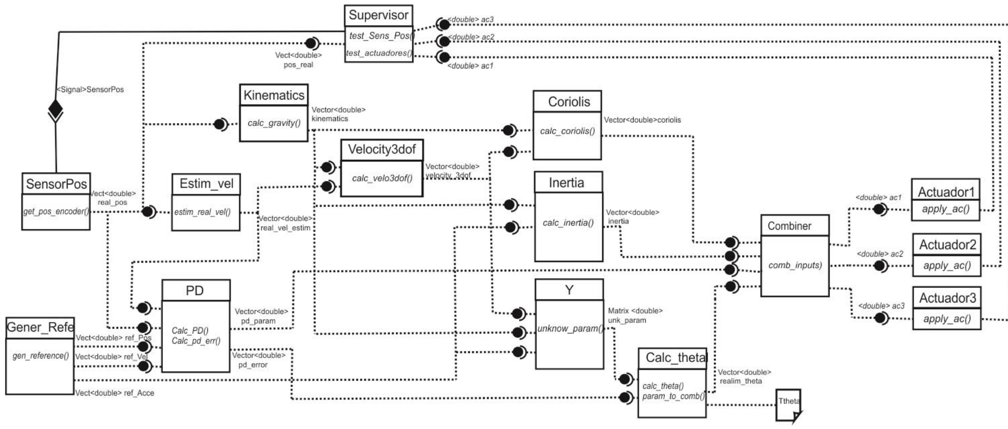


Fig. 5. Adaptive control flowchart implemented in Orocos

Furthermore, component-based software development (with Orocos) has a number of advantages such as:

- Easy flow tracking execution.
- Distributed execution, with each component in a different thread, reducing the execution time.
- Code-reusability. Note that the two controllers implemented (Eq. 24 and 26) have very similar structure. Thus, a single component has to be implemented only once and can be used on any other scheme many times.

In order to implement the control architecture for the parallel robot, an industrial PC has been used. It is based on a high performance 4U Rackmount industrial system with 7 PCI slots and 7 ISA slots. It has a 3,06GHz Intel® Pentium® 4 processor and two GB DDR 400 SDRAM. The industrial PC is equipped with 2 Advantech™ data acquisition cards: a PCI-1720 and a PCL-833.

The PCI-1720 card has been used for supplying the control actions for each parallel robot actuator. It provides four 12-bit isolated digital-to-analog outputs for the Universal PCI 2.2 bus. It has multiple output ranges (0~5V, 0~10V, ±5V, ±10V), programmable software and an isolation protection of 2500 VDC between the outputs and the PCI bus. The PCL-833 card is a 4-axis quadrature encoder and counter add-on card for an ISA bus. The card includes four 32-bit quadruple AB phase encoder counters, an onboard 8-bit timer with a wide range time-based selector and it is optically isolated up to 2500V. Fig. 6 shows the control architecture based on an industrial PC developed for this study.

With this hardware and software control architecture different controllers and tests have been carried out. The real executions have shown that the robot response is very good and, if a payload is added to the moving platform, see Fig. 7, a direct change occurs in the estimation of the rigid body parameters, which is not the case for the viscous friction.

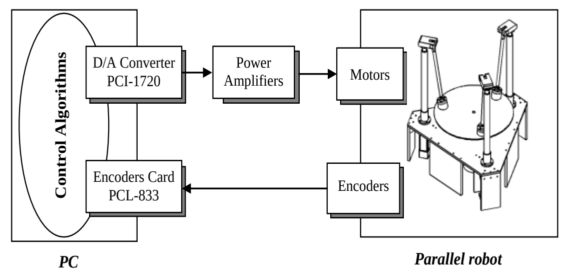


Fig. 6. Robot control architecture.

Using the parallel robot and its open hardware and software control system, different control algorithms have been developed and tested. It’s remarkable that all the schemes has been made in a modular way (using Orocos), with a cascade control and a frequency of 100Hz (t_{sample}=10 ms). For example, in Fig. 5 can be seen the adaptive control implemented in a modular way using Orocos. As commented above, the execution is in cascade, being the *SensorPos* component in charge of waking up the other components. Using this technique, a distributed execution is performed, decreasing execution time.

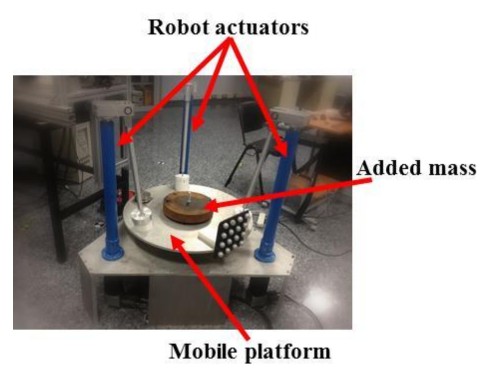


Fig. 7.Actual parallel robot with the 30kg load placed on the mobile platform

The following figures show the response obtained from the actual robot: Fig.8 shows the reference and the robot q_1 positions for the adaptive and the passivity-based controllers. Fig. 9 shows the absolute error values of passivity-based and adaptive controllers. Fig. 10 shows the control action (in volts.) provided by the adaptive controller. The motion references are very similar as the references 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 = 93seg). This time allows us to place a load of 30kg on the robot platform.

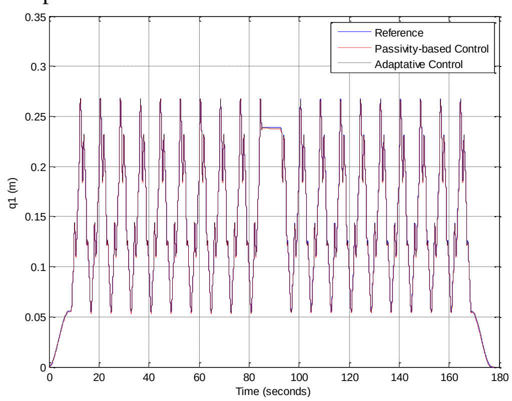


Fig. 8. Reference and robot positions (passivity-based and adaptive controllers) for the first actuated joint.

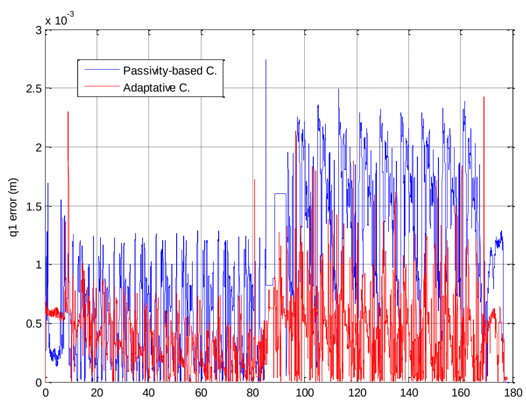


Fig. 9.Absolute error position for the first actuated joint

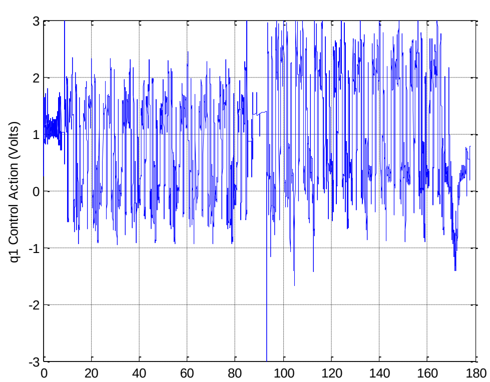


Fig. 10.Control action (adaptive controller) for the first actuated joint

The results obtained with the actual robot agree with those obtained in the simulation: because of the estimation the on-line dynamic parameters, the change of the load means that the robot response using an adaptive controller is significantly better than that obtained with the passivity-based controller.

The mean squared error of both controllers can be seen in Tab. 4. There, one can observe that during the first 90 seconds (without payload) the error with an adaptive and a passivity-based control is very similar. However, after placing a load of 30 kg on the robot platform (the next 90 seconds), the adaptive control works much better than the other one. This is because the adaptive control calculates the new dynamic parameters on-line (after placing the charge). However, since the passivity-based control doesn't calculate the dynamic parameters online, after putting the weight, the error increases significantly.

Tab. 4. Mean squared error (MSE) of adaptive and passivity-based controllers.

Errors	$\sqrt{\frac{\sum_{j=1}^{n}\sum_{i=1}^{DOF}\left(e_{i,j}\right)^2}{n \cdot DOF}}$	
	Adaptive Control	Passivity-Based Control
Without Payload	0.0004357412	0.0006278345
With Payload	0.0005678401	0.0014765707

4. Conclusions

In this paper, the adaptive control of a 3-DOF parallel manipulator was considered. The adaptive controller is based on a reduced robot dynamic model. This model contains only a set of relevant parameters obtained through a process which considers the robot's leg symmetries, the statistical significance of the identified parameters, and the physical feasibility of the parameters. The reduced dynamic model has uncertainties not only because of the unmodeled dynamics but also when unknown inertial parameters can appear, for instances, when a payload mass is grasped by the robot. This kind of uncertainty limits the applicability of model-based controllers. To relax the requirement of exact knowledge, an adaptive controller for the PM was implemented. The adaptive scheme can be rewritten depending on the robot parameters that are assumed to be unknown.

In order to analyze and validated the control algorithms, they have been tested on a virtual and an actual prototype of a parallel robot. The simulations of the virtual robot were developed in Matlab/Simulink. The actual prototype is a low-cost, parallel robot developed at Universitat Politècnica de Valencia.

The control of the actual robot has been implemented in Orocos middleware. Because it is a component-based middleware, Orocos provides several advantages like modular design and structure, reusable code, modules reconfiguration in real-time.

Using Orocos, an adaptive and a passivity-based controller have been developed. The results indicate that the adaptive controller perform better than the passivity-based controller if there are differences between the supposed and the real parameters used in the robot dynamic model.

5. Acknowledgments

The authors want to express their gratitude to the Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (FEDER-CICYT) for the partial financing of this research under the projects DPI2010-20814-C02-(01, 02) and DPI2011-28507-C02-01. This research was also partially funded by the CDCHT-ULA Grant I-1286-11-02-B.

6. References

Abdellatif, H. and B. Heimann. 2010. Advanced Model-Based Control of a 6-DOF Hexapod Robot: A Case Study. IEEE/ASME Transactions on Mechatronics, Vol. **15**, pp.269-279.

Bayard, D. S. and J.T. Wen. 1988. New class of control laws for robotic manipulators Part 2. Adaptive case. International journal of control, Vol. **47** (5), pp.1387-1406.

Carretero J.A., R.O. Podhorodeski and M.A. Nahon. 2000. Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. Journal of Mechanical Design, Vol. **122**, pp.17-24.

Chablat D. and P. Wenger. 2003. Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the Orthoglide. IEEE Transactions on Robotics and Automation, Vol. **19**, pp.402-410.

Company O. and F. Pierrot. 2002. Modelling and design issues of a 3-axis parallel machine-tool. Mechanism and Machine Theory, Vol. **37**, pp.1325-1345.

Díaz-Rodriguez M., V. Mata, A. Valera and A. Page. 2010. A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. Mechanism and Machine Theory, Vol. **45**, pp.1337-1356.

Diaz-Rodriguez M., A. Valera, V. Mata V and M. Valles. 2012. Model-Based Control of a 3-DOF Parallel Robot Based on Identified Relevant Parameters. IEEE/ASME Trans on Mechatronics. DOI: 10.1109/TMECH.2012.2212716.

Fu K. and J.K. Mills. 2007. Robust control design for a planar parallel robot. International Journal of Robotics & Automation, Vol. **22**, pp.139-147.

García de Jalón J. and E. Bayo. 1994. Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge. Springer-Verlag, New-York.

Gou H.B., Y.G. Liu, F.R. Liu and H.R. Li. 2009. Cascade control of a hydraulically driven 6-dof parallel robot manipulator based on a sliding mode. Control Engineering Practice, Vol. **16**, pp.105-1068.

Li Y. and Q. Xu. 2007. Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation. IEEE/ASME Transaction on Mechatronics, Vol. **12**, pp.265–273.

Merlet, J.P. 2000. Parallel Robots. Kluwer, London, U.K.

Merlet, J.P. 2002. Optimal design for the micro parallel robot MIPS. In Proceedings of IEEE Int. Conf. on Robotics and Automation, pp.1149-1154.

Nabat V. and M. de la O Rodriguez, O. Company, S. Krut and F. Pierrot. 2005. PAR4: very high speed parallel robot for pick-and-place. In Proceedings of Int. Conf. on Intelligent Robots and Systems (IROS'05), Edmonton, Canada, pp.1202-1207.

Paden, B. and R. Panja. 1988. Globally Asymtotically Stable 'PD+' Controller for Robot Manipulators. International Journal on Control, Vol. **47**, pp.1697-1712.

Stan S.D., R.Balan, V. Maties and C. Rad. 2009. Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot. Mechanika, Vol. **75**, pp.62-66.

Stewart, D.A. 1965. A platform with 6 degree of freedom. Proceedings of the Institution of mechanical engineers Part 1, Vol. **15**, pp.371-386.

Tsai, L.W. 1999. Robot Analysis: The Mechanics of Serial and Parallel Manipulator. Wiley Interscience, Canada.

Valles M., M. Díaz-Rodríguez, A. Valera, V. Mata and A. Page. 2012. Mechatronic Development and Dynamic Control of a 3-DOF Parallel Manipulator. Mechanics Based Design of Structures and Machines, Vol. **40**, pp.434-452.

Zhan Y.X., S. Cong, W.W. Shang, Z.X. Li and S.L. Jiang. 2007. Modeling, Identification and Control of a Redundant Planar 2-DOF Parallel Manipulator. International Journal of Control, Automation and Systems, Vol. **5**, pp.559-569.