Predictive Functional Control of a Parallel Robot

Oscar Andrès Vivas, Philippe Poignet

▶ **To cite this version:**

Oscar Andrès Vivas, Philippe Poignet. Predictive Functional Control of a Parallel Robot. Control Engineering Practice, 2005, Control Applications of Optimisation, 13 (7), pp.863-874. 10.1016/j.conengprac.2004.10.001 . lirmm-00105300

HAL Id: lirmm-00105300

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00105300

Submitted on 11 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Control Engineering Practice 13 (2005) 863–874
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Predictive functional control of a parallel robot

Andrés Vivas, Philippe Poignet*

LIRMM, UMR 5506, CNRS, University Montpellier 2, Robotique, 161 rue Ada, 34392 Montpellier cedex 5, France

Received 31 July 2003; accepted 4 October 2004
Available online 18 November 2004

Abstract

This paper deals with an efficient application of a model-based predictive control scheme in parallel mechanisms. A predictive functional control strategy based on a simplified dynamic model is implemented. Experimental results are shown for the H4 robot, a fully parallel structure providing 3 degrees of freedom (dof) in translation and 1 dof in rotation. Predictive functional control, computed torque control and PID control strategies are compared in complex machining tasks trajectories. Tracking performance and disturbance rejection are enlightened.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Model-based control; Predictive control; Robot control

1. Introduction

Parallel mechanisms were introduced by Gough (1957) and Stewart (1965). Clavel (1989) proposed the Delta structure, a parallel robot dedicated to high-speed applications, which has been intensively used in industry. Also the so-called "hexapod" robot with six U-P-S chains in parallel (U-P-S: Universal-Prismatic-Spherical) is another undeniable success for parallel robotics thanks to the enormous amount of research dedicated to these structures (Merlet, 1997; Thönshoff, 1998). Others structures like the Hexa robot (Pierrot, Dauchez, & Fournier, 1991) and the HexaM machine (Pierrot & Shibukawa, 1998) propose different solutions for machining tasks.

For most pick-and-place applications, at least four dof are required (3 translations and 1 rotation to put the carried object in its final location). For the Delta robot, it is achieved thanks to an additional link between the base and the gripper, but it does not seem to be as efficient as a parallel arrangement. Moreover, 6-degree of freedom (dof) fully parallel machines currently used in machining suffer from their complexity (they need at least 6 motors while the cutting process requires only 5 controlled axis plus the spindle rotation) and their limited tilting angle. An intermediate solution to these drawbacks, with the 4-dof parallel mechanism—the H4 robot—have been proposed (Company & Pierrot, 1999; Pierrot, Marquet, Company, & Gil, 2001). Fig. 1 shows a photograph of the H4 parallel robot.

This machine is based on 4 independent active chains between the base and the nacelle; each chain is actuated by a brushless direct-drive motor fixed on the base and equipped with an incremental position encoder. Thanks to its design, the mechanism is able to provide great performance. However, in order to achieve high speed and acceleration for pick-and-place applications or precise motion in machining tasks, advanced model-based robust controllers are often required to increase the performance of the robot.

In the past decade model predictive control (MPC) has become an efficient control strategy for a large number of processes (Clarke, Mothadi, & Tuffs, 1987). Different works have shown that predictive control schemes are of great interest when requiring good performance in terms of rapidity, disturbances or errors

*Corresponding author. Tel.: +33 467 418 561; fax: +33 467 418 500.
E-mail address: poignet@lirmm.fr (P. Poignet).

0967-0661/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2004.10.001

Fig. 1. H4 robot.

cancellations (Clarke et al., 1987; Allgöwer, Badgwell, Qin, Rawlings, & Wright, 1999). Stability has been the subject of several studies (Van den Boom, De Vries, Boumeester, & Verbruggen, 1994; Zheng, 1997).

In this paper, the focus is on the implementation of the predictive functional control (PFC) developed by Richalet (Richalet, 1993a,b; Richalet et al., 1997) on the H4 parallel robot. Recent contributions to this approach can be found in Rossiter (2002) as well as in (Abdelghani-Idrissi, Arbaoui, Estel, & Richalet, 2001), where industrial applications restricted to slow dynamic systems are presented.

Basically, the procedure will consist of two steps (i) the process is first linearized by feedback (ii) secondly, the MPC scheme is computed from a linearized model composed of a set of double integrators first stabilized with an inner closed-loop structure. Experimental results, enlightening performance on circular or angular path and robustness to load variation, are compared with those obtained from the model-based computed torque control (CTC) (Canudas de Wit, Siciliano, & Bastin, 1996) and the classical PID controller.

The paper is organized as follows: Section 2 is dedicated to the geometric, kinematics and dynamics modeling required to implement the control strategy. Section 3 details the model PFC. Section 4 introduces the compared control schemes: PFC, CTC and PID. Section 5 lists major experimental results in terms of (i) tracking performance in complex trajectories, that can be found in pick-and-place applications or machining tasks, and (ii) robustness. Finally, conclusions are given in Section 6.

2. Robot modeling

2.1. Geometric and kinematics modeling

The Jacobian matrix and the forward geometric model are required to compute the dynamic model (see Section 2.2) Khalil and Dombre, 2002. Therefore, a brief presentation detailing the computation of the different relationships required to obtain this model and matrix are presented. The design parameters of the robot are described in Fig. 2, where the following parameters have been chosen:

$$\alpha_1 = 0, \quad \alpha_2 = \pi, \quad \alpha_3 = 3\pi/2, \quad \alpha_4 = 3\pi/2,$$
$$\boldsymbol{u}_1 = \boldsymbol{u}_y, \quad \boldsymbol{u}_2 = -\boldsymbol{u}_y, \quad \boldsymbol{u}_3 = \boldsymbol{u}_x, \quad \boldsymbol{u}_4 = \boldsymbol{u}_x.$$

The angles α_i describe the position of the four motors, L is the length of arms, l is the length of the forearms, θ the nacelle's angle, and d and h are the half lengths of the "H" forming the nacelle. $\boldsymbol{O}$ is the origin of the base frame and D is the origin of the nacelle frame. R gives the motor's position. The $\boldsymbol{A}_i\boldsymbol{B}_i$ segments represent the arms of the robot and $\boldsymbol{P}_i\boldsymbol{B}_i$ the forearm segments. The joint positions are represented by q_i.

To obtain the inverse geometric model, it is necessary to express the different points of the mechanical system with respect to the origin $\boldsymbol{O}$. The origin is fixed in the middle of the nacelle with the coordinates (x, y, z). In the Cartesian space, the end effector position is given by (x, y, z, θ).

$$\boldsymbol{OD} = [x\ y\ z]^{\mathrm{T}} \tag{1}$$

The vector that joins the absolute origin $\boldsymbol{O}$ and all the forearms to the nacelle is

$$\boldsymbol{OA}_i = \boldsymbol{OD} + \boldsymbol{DA}_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \boldsymbol{DA}_i. \tag{2}$$

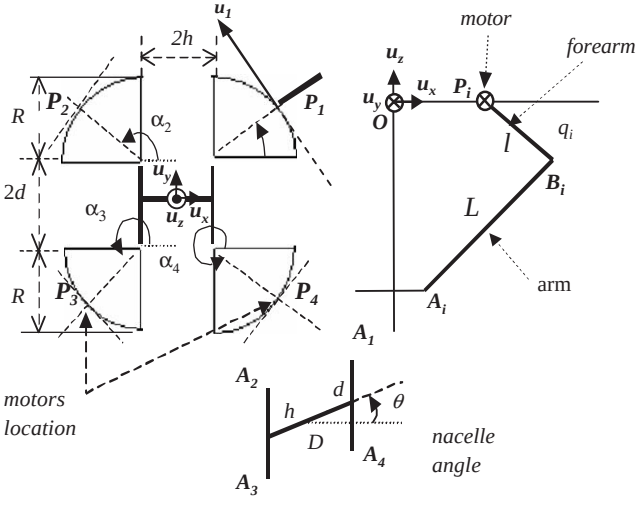

Fig. 2. Design parameters.

The $\boldsymbol{DA}_i$ segments can be expressed as

$$\boldsymbol{DA}_1 = \begin{bmatrix} h\cos\theta \\ h\sin\theta + d \\ 0 \end{bmatrix}, \quad \boldsymbol{DA}_2 = \begin{bmatrix} -h\cos\theta \\ -h\sin\theta + d \\ 0 \end{bmatrix}. \quad (3)$$

$$\boldsymbol{DA}_3 = \begin{bmatrix} -h\cos\theta \\ -h\sin\theta - d \\ 0 \end{bmatrix}, \quad \boldsymbol{DA}_4 = \begin{bmatrix} h\cos\theta \\ h\sin\theta - d \\ 0 \end{bmatrix}. \quad (4)$$

Then, the vector that links the absolute origin and all of the arms to the forearms is

$$\boldsymbol{OB}_i = \boldsymbol{OP}_i + \boldsymbol{P}_i\boldsymbol{B}_i \quad (5)$$

with

$$\boldsymbol{P}_i\boldsymbol{B}_i = \begin{bmatrix} l\cos q_i\cos\alpha_i \\ l\cos q_i\sin\alpha_i \\ -l\sin q_i \end{bmatrix} \quad (6)$$

and actuator locations are:

$$\boldsymbol{OP}_1 = \begin{bmatrix} h + R\cos\alpha_1 \\ d + R\sin\alpha_1 \\ 0 \end{bmatrix}; \quad \boldsymbol{OP}_2 = \begin{bmatrix} -h + R\cos\alpha_2 \\ d + R\sin\alpha_2 \\ 0 \end{bmatrix}. \quad (7)$$

$$\boldsymbol{OP}_3 = \begin{bmatrix} -h + R\cos\alpha_3 \\ -d + R\sin\alpha_3 \\ 0 \end{bmatrix}; \quad \boldsymbol{OP}_4 = \begin{bmatrix} h + R\cos\alpha_4 \\ -d + R\sin\alpha_4 \\ 0 \end{bmatrix}. \quad (8)$$

Finally, arms coordinates are given by

$$\boldsymbol{A}_i\boldsymbol{B}_i = \boldsymbol{A}_i\boldsymbol{O} + \boldsymbol{OB}_i. \quad (9)$$

As usual for most parallel robots, the inverse geometric model is easy to be computed. The following equality can simply be written

$$\|\boldsymbol{A}_i\boldsymbol{B}_i\|^2 = L^2, \quad i = 1,\ldots,4. \quad (10)$$

This relationship leads to

$$M_i\cos q_i + N_i\sin q_i = G_i, \quad (11)$$

where

$$M_i = -2l(P_iB_{ix}\cos\alpha_i + P_iB_{iy}\sin\alpha_i),$$

$$N_i = 2lP_iB_{iz},$$

$$G_i = L^2 - l^2 - \boldsymbol{P}_i\boldsymbol{B}_i^2.$$

Resorting to the new variable $t_i = \tan(\theta_i/2)$, the result is

$$q_i = 2\tan^{-1}\left(\frac{-b_i^2 \pm \sqrt{b_i - 4a_ic_i}}{2a_i}\right), \quad (12)$$

where

$$a_i = G_i + M_i,$$

$$b_i = -2N_i,$$

$$c_i = G_i - M_i.$$

Eq. (11) thus solved, a mathematical singularity can occur when $a_i = 0$. It is possible to overcome this problem by introducing the following new variables:

$$\tan\chi_i = \frac{N_i}{M_i}, \quad \cos\beta_i = \frac{G_i}{M_i}. \quad (13)$$

This leads to another expression of the inverse kinematics:

$$q_i = \tan^{-1}\left(\frac{N_i}{M_i}\right) \pm \cos^{-1}\left(\frac{G_i}{\sqrt{M_i^2 + N_i^2}}\right). \quad (14)$$

Kinematics models have been used to determine the H4 workspace, depicted in Fig. 3 for $\theta = 0$. To have a good Jacobian matrix condition number, workspace should be limited. Eventually, this workspace is limited to a $(300 \times 300 \times 300)\,\mathrm{mm}^3$ centered cube: work will be done within this workspace. More details can be found in Pierrot et al. (2001) and Company, Marquet, and Pierrot (2003).

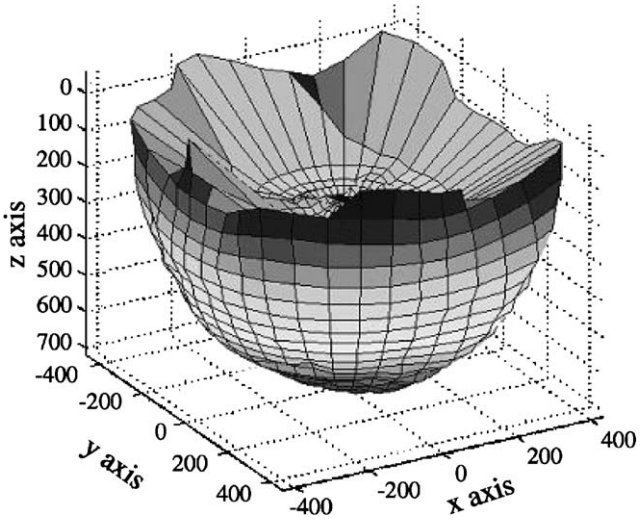

Fig. 3. H4 workspace for $\theta = 0$.

The analytical forward geometric model of a parallel robot is more difficult to compute. Up to now, the simplest model is an eighth degree polynomial equation. The forward model is then computed successively using the classical recurrent formula:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{J}(\boldsymbol{x}_n, \boldsymbol{q}_n)[\boldsymbol{q} - \boldsymbol{q}_n], \tag{15}$$

where $\boldsymbol{q}$ is the convergence point and $\boldsymbol{J}$ is the robot Jacobian matrix. If the mechanism is not in a singular configuration, this expression is derived as follows (Company & Pierrot, 1999; Pierrot et al., 2001):

$$\boldsymbol{J} = \boldsymbol{J}_x^{-1}\boldsymbol{J}_q, \tag{16}$$

where

$$\boldsymbol{J}_x = \begin{bmatrix} A_1B_{1_x} & A_1B_{1_y} & A_1B_{1_z} & (\boldsymbol{DC}_1 \times A_1B_1)_z \\ A_2B_{2_x} & A_2B_{2_y} & A_2B_{2_z} & (\boldsymbol{DC}_2 \times A_2B_2)_z \\ A_3B_{3_x} & A_3B_{3_y} & A_3B_{3_z} & (\boldsymbol{DC}_3 \times A_3B_3)_z \\ A_4B_{4_x} & A_4B_{4_y} & A_4B_{4_z} & (\boldsymbol{DC}_4 \times A_4B_4)_z \end{bmatrix}. \tag{17}$$

$$\boldsymbol{J}_q = \mathrm{diag}((\boldsymbol{PiBi} \times \boldsymbol{AiBi}) \bullet u_{mi}); \quad i = 1, \ldots, 4. \tag{18}$$

$\boldsymbol{DC}_i$ is the distance between the center of the nacelle and the center of the half-lengths of the "H" that forms the nacelle.

2.2. Dynamic modeling

In a first approximation, the dynamic model is computed by considering the physical dynamics. Indeed, drive torques are mainly used to move the motor inertia, the forearms, arms and the nacelle that can be equipped with a machining tool. Because of the design, the forearm inertia can be considered as a part of the motor inertia and the arm (manufactured in carbon materials) effects are neglected (Company & Pierrot, 1999; Pierrot et al., 2001).

If $\boldsymbol{\Gamma}_{mot} (\in \mathbb{R}^{4\times4})$ is the actuator torque vector, the basic equation of dynamics can be written as

$$\boldsymbol{\Gamma}_{mot} = \boldsymbol{I}_{mot}\ddot{\boldsymbol{q}} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}(\ddot{\boldsymbol{x}} - \boldsymbol{G}), \tag{19}$$

where $\boldsymbol{I}_{mot}$ represents the motor's inertia diagonal matrix (Eq. (20)) including the forearm's inertia, $\boldsymbol{M}$ ($\in \mathbb{R}^{4\times4}$) is a diagonal matrix (Eq. (21)) containing the mass of the nacelle and its inertia (M_{nac} and I_{nac}, respectively), $\boldsymbol{J}$ is the Jacobian matrix given in Eq. (16), $\ddot{\boldsymbol{x}}$ is the vector of cartesian accelerations, and $\boldsymbol{G}$ the gravity vector. Thanks to the design, the forearm's inertia is taken into account in the motor's inertia,

with

$$\boldsymbol{I}_{mot} = \begin{bmatrix} I_{mot1} & 0 & 0 & 0 \\ 0 & I_{mot2} & 0 & 0 \\ 0 & 0 & I_{mot3} & 0 \\ 0 & 0 & 0 & I_{mot4} \end{bmatrix} \tag{20}$$

$$\boldsymbol{M} = \begin{bmatrix} M_{nac} & 0 & 0 & 0 \\ 0 & M_{nac} & 0 & 0 \\ 0 & 0 & M_{nac} & 0 \\ 0 & 0 & 0 & I_{nac} \end{bmatrix}. \tag{21}$$

The motor position $\boldsymbol{q} = [q_1\ q_2\ q_3\ q_4]^{\mathrm{T}}$ are directly measured, and the velocity $\dot{\boldsymbol{q}}$ and acceleration $\ddot{\boldsymbol{q}}$ are obtained by a first-order differentiation. As the acceleration measurement $\ddot{\boldsymbol{x}}$ is not available, $\ddot{\boldsymbol{x}}$ is computed with

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}, \tag{22}$$

where $\boldsymbol{J}$ depends on $\boldsymbol{x}$ and $\boldsymbol{q}$, and $\dot{\boldsymbol{J}}$ is computed using a first-order differentiation.

Then, the dynamic model can be written as

$$\boldsymbol{\Gamma}_{mot} = \boldsymbol{A}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{23}$$

with

$$\boldsymbol{A}(\boldsymbol{q}) = \boldsymbol{I}_{mot} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{J} \tag{24}$$

and

$$\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{G}. \tag{25}$$

2.3. Dynamic parameter estimation

The dynamic model can be linearly expressed with respect to the dynamic parameters by (Vivas, Poignet, Marquet, Pierrot, & Gautier, 2003):

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{\theta}, \tag{26}$$

where $\boldsymbol{Y}$ is the measurement vector of joint torques, $\boldsymbol{W}$ is the observation matrix and $\boldsymbol{\theta}$ is the parameters vector to be estimated. The parameters vector is then estimated using weighted least-squares techniques (Vivas et al., 2003).

Joint velocities and accelerations required in Eq. (23) are estimated by a band-pass filtering of the position. The band-pass filtering is obtained by the product of a low-pass filter in both the forward and the reverse direction and a derivative filter obtained by a central difference algorithm, without phase shift. A parallel filtering is implemented to reject the high-frequency ripples of the measured motor torques.

Exciting trajectories, composed of concatenated motions leading to a good condition number of the

Table 1
Estimated parameters (SI units)

Physical parameters	Estimated values	$\%\sigma_{\hat{x}r}$
I_{mot1}	0.0167	2.37
I_{mot2}	0.0164	2.36
I_{mot3}	0.0176	1.58
I_{mot4}	0.0234	1.16
M_{nac}	0.9840	0.47
I_{nac}	0.0029	3.73

observation matrix $\boldsymbol{W}$, are generated. The obtained estimated values, given in Table 1, will be considered as the nominal values for the feedback linearization (see Section 4.2) during the experiments. The relative standard deviation ($\%\sigma_{\hat{x}r}$) is also given. More details concerning the identification procedure may be found in (Vivas et al., 2003; Poignet & Gautier, 2001; Canudas de Wit et al., 1996).

3. Predictive functional control

This section is dedicated to briefly recall the main steps of the model PFC scheme used hereafter for the implementation. This predictive technique has been developed by Richalet and complete details of the computation may be found in Richalet (1993b); Richalet et al. (1997).

3.1. Internal modeling

The model used is a linear one given by

$$\boldsymbol{x}_M(n) = \boldsymbol{F}_M \boldsymbol{x}_M(n-1) + \boldsymbol{G}_M \boldsymbol{u}(n-1),$$

$$\boldsymbol{y}_M(n) = \boldsymbol{C}_M^{\mathrm{T}} \boldsymbol{x}_M(n), \tag{27}$$

where $\boldsymbol{x}_M$ is the state, u is the input, $\boldsymbol{y}_M$ is the measured model output, $\boldsymbol{F}_M$, $\boldsymbol{G}_M$ and $\boldsymbol{C}_M$ are, respectively, matrices or vectors of the right dimension.

If the system is unstable, the problem of robustness caused by the controller's cancellation of the poles is usually solved by a model decomposition (Richalet, 1993b).

3.2. Reference trajectory

The predictive control strategy of the MPC is summarized in Fig. 4. Given the set point trajectory over a receding horizon [0, h], the predicted process output $\hat{y}_P$ will reach the future set point following a reference trajectory y_R.

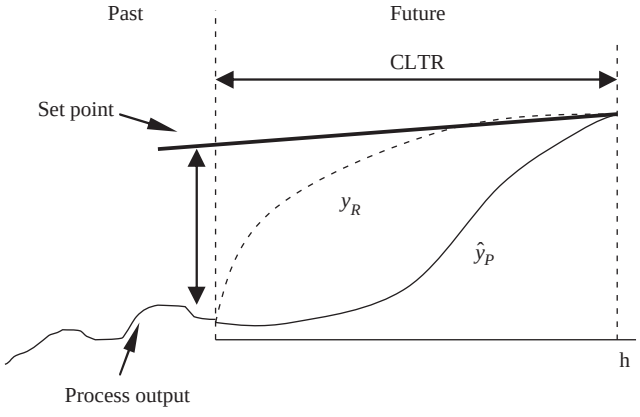

Fig. 4. Reference trajectory and predictive control strategy.

In Fig. 4, $\varepsilon(n) = c(n) - y_P(n)$ is the position tracking error at time n, c is the set point trajectory, y_P is the process output, and CLTR is the closed-loop time response.

Over the receding horizon, the reference trajectory y_R, which is the path towards the future set point, is given by

$$c(n+i) - y_R(n+i) = \alpha^i (c(n) - y_P(n)) \quad \text{for } 0 \leqslant i \leqslant h, \tag{28}$$

where $\alpha (0 < \alpha < 1)$ is a scalar which has to be chosen as a function of the desired closed-loop response time.

The predictive essence of the control strategy is completely included in Eq. (28). Indeed, the aim is to track the set point trajectory following the reference trajectory. This trajectory may be considered as the desired closed-loop behavior.

3.3. Performance index

The performance index may be a quadratic sum of the errors between the predicted process output $\hat{y}_P$ and the reference trajectory y_R. It is defined as follows:

$$D(n) = \sum_{j=1}^{n_h} \left\{ \hat{y}_P(n+h_j) - y_R(n+h_j) \right\}^2, \tag{29}$$

where n_h is the number of coincidence time point, h_j are the coincidence time points over the prediction horizon. The predicted output $\hat{y}_P$ is usually defined as

$$\hat{y}_P(n+i) = y_M(n+i) + \hat{e}(n+i), \quad 1 \leqslant i \leqslant h, \tag{30}$$

where y_M is the model output, $\hat{e}$ is the predicted future output error.

It may be convenient to add a smoothing control term in the performance index. The index becomes

$$D(n) = \sum_{j=1}^{n_h} \{ \hat{y}_P(n+h_j) - y_R(n+h_j) \}^2 + \lambda \{ u(n) - u(n-1) \}^2, \tag{31}$$

where u is the control variable.

3.4. Control variable

The future control variable is assumed to be composed of a priori known functions

$$u(n+i) = \sum_{k=1}^{n_B} \mu_k(n) u_{BK}(i), \quad 0 \leqslant i \leqslant h, \tag{32}$$

where μ_k are the coefficients to be computed during the optimization of the performance index, u_{BK} are the base functions of the control sequence, n_B is the number of base functions.

The choice of the base functions depends on the nature of the set point and the process. Dynamical functions will be used hereafter

$$u_{BK}(i) = i^{k-1} \quad \forall k. \tag{33}$$

In fact, only the first term is effectively applied for the control, that is

$$u(n) = \sum_{k=1}^{n_B} \mu_k(n) u_{BK}(0). \tag{34}$$

The model output is composed of two parts:

$$y_M(n+i) = y_{UF}(n+i) + y_F(n+i), \quad 1 \leqslant i \leqslant h, \tag{35}$$

where y_{UF} is the free (unforced) output response ($u = 0$), y_F is the forced output response to the control variable given by Eq. (32).

Given Eq. (27) and Eq. (32), it follows

$$y_{UF}(n+i) = \boldsymbol{C}_M^T \boldsymbol{F}_M^i \boldsymbol{x}_M(n), \quad 1 \leqslant i \leqslant h,$$

$$y_F(n+i) = \sum_{k=1}^{n_B} \mu_k(n) y_{BK}(i), \quad 0 \leqslant i \leqslant h, \tag{36}$$

where y_{BK} is the model output response to u_{BK}. Assuming that the predicted future output error is approximated by a polynomial function, it follows

$$\hat{e}(n+i) = e(n) + \sum_{m=1}^{de} e_m(n) i^m, \quad \text{for } 1 \leqslant i \leqslant h, \tag{37}$$

where d_e is the degree of the polynomial approximation of the error, e_m are coefficients computed on-line knowing the past and present output error.

The minimization of the performance index without smoothing control term, in the case of the polynomial base functions, leads to the applied control variable:

$$u(n) = k_o\{c(n) - y_P(n)\} + \sum_{m=1}^{\max(dc,de)} k_m\{c_m(n) - e_m(n)\} + V_X^T x_M(n), \tag{38}$$

where d_c is the degree of the polynomial approximation of the set point and the gains k_o, k_m, V_X^T are computed off-line (see the Appendix).

Therefore, the control variable is composed of three terms: the first one is due to the tracking position error, the second one is placed especially for disturbance rejection and the last one corresponds to a model compensation.

4. Compared control strategies

Performance and robustness of PID, CTC and PFC controllers are compared on complex trajectories given in the task space such as a circle or angular path. The CTC and PFC controllers are based on non-linear compensation and decoupling through the computation of the inverse dynamic model, described in Section 4.2 (Vivas & Poignet, 2003). A stabilizing linear controller is then applied (Figs. 7 and 8).

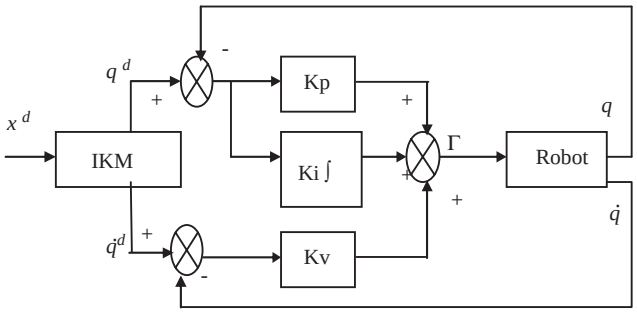

Fig. 5. PID controller in the task space.

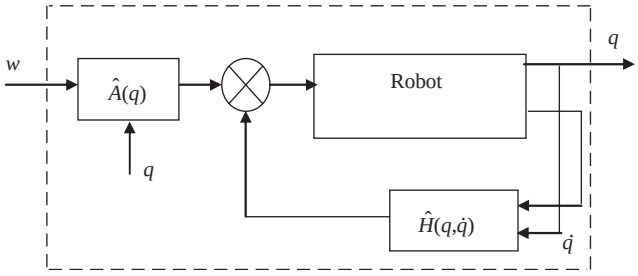

Fig. 6. Feedback linearized system.

4.1. PID controller in the task space

The PID controller applied to H4 robot in the task space is shown in Fig. 5. Frequency analysis yielded resonance frequency ($\omega_r = 50\,\text{rad/s}$). Using the tuning procedure proposed in (Khalil & Dombre (2002)), the gain parameters of the controller have been adjusted to $K_p = 500$, $K_i = 5000$, and $K_d = 6$, which guarantees a good bandwidth and good tracking performance.

4.2. Feedback linearization

In order to compute the PFC control strategy (Poignet & Gautier, 2000) as well as for the design of the CTC controller, it is basically required to linearize the non-linear dynamic model of the robot. Let us consider the non-linear dynamic equations for an m-link robot expressed as follows:

$$\boldsymbol{\Gamma} = \boldsymbol{A}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{39}$$

It is well known that the rigid m-link robot equations may be linearized and decoupled by non-linear feedback (Khalil, 1996; Khalil & Dombre, 2002). Let $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{H}}$, respectively, be the estimates of $\boldsymbol{A}$ and $\boldsymbol{H}$. Assuming that $\hat{\boldsymbol{A}} = \boldsymbol{A}$ and $\hat{\boldsymbol{H}} = \boldsymbol{H}$, the problem is reduced to that of the linear control on n decoupled double-integrators:

$$\ddot{\boldsymbol{q}} = \boldsymbol{w}, \tag{40}$$

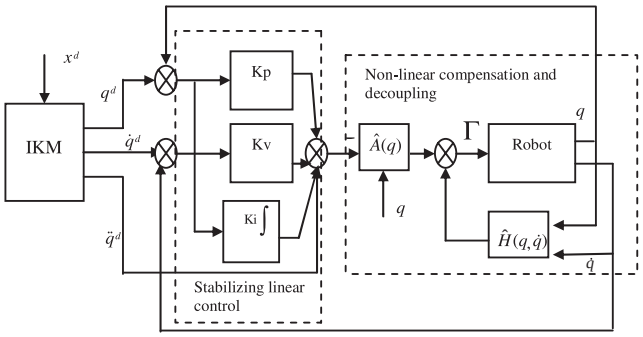

Fig. 7. CTC control.

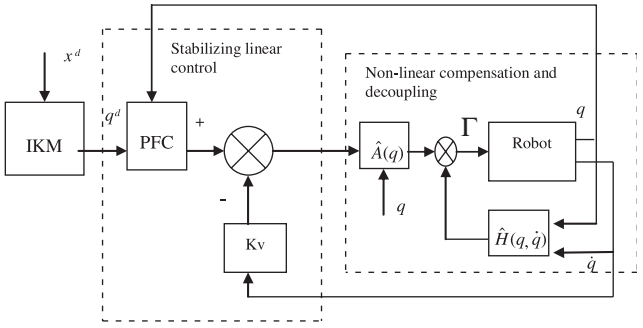

Fig. 8. PFC control.

where $\boldsymbol{w}$ is the new input control vector. This equation corresponds to the familiar inverse dynamics control scheme where the direct dynamic model characterizing the robot is transformed into a double set of integrators (Fig. 6).

Linear control techniques (Lewis, 1992; Khalil & Dombre, 2002) can then be used to design a tracking controller such as the model-based predictive control scheme described in Section 3 or a PID controller in case of the CTC described in the next section.

4.3. Computed torque control

Assuming that the motion is completely specified with the desired position $\boldsymbol{q}^d$, velocity $\dot{\boldsymbol{q}}^d$ and acceleration $\ddot{\boldsymbol{q}}^d$, the computed torque control (Canudas de Wit et al., 1996) computes the required input control vector as follows:

$$\boldsymbol{w} = \boldsymbol{K}_p(\boldsymbol{q}^d - \boldsymbol{q}) + \boldsymbol{K}_v(\dot{\boldsymbol{q}}^d - \dot{\boldsymbol{q}}) + \ddot{\boldsymbol{q}}^d, \tag{41}$$

where $\boldsymbol{K}_p$, $\boldsymbol{K}_v$ are the controller gains.

An integrator has been added to the classical scheme to decrease the static error due to the errors between the estimated inverse dynamic model and the real one (Fig. 7). Then, Eq. (41) becomes

$$\boldsymbol{w} = K_p(\boldsymbol{q}^{\boldsymbol{d}} - \boldsymbol{q}) + K_v(\dot{\boldsymbol{q}}^{\boldsymbol{d}} - \dot{\boldsymbol{q}}) + K_i \int (\boldsymbol{q}^d - \boldsymbol{q}) d\tau + \ddot{\boldsymbol{q}}^d. \tag{42}$$

The tuning of the CTC controller (Khalil & Dombre, 2002), leads to the gains $K_p = 5000$; $K_v = 65$ and $K_i = 60000$.

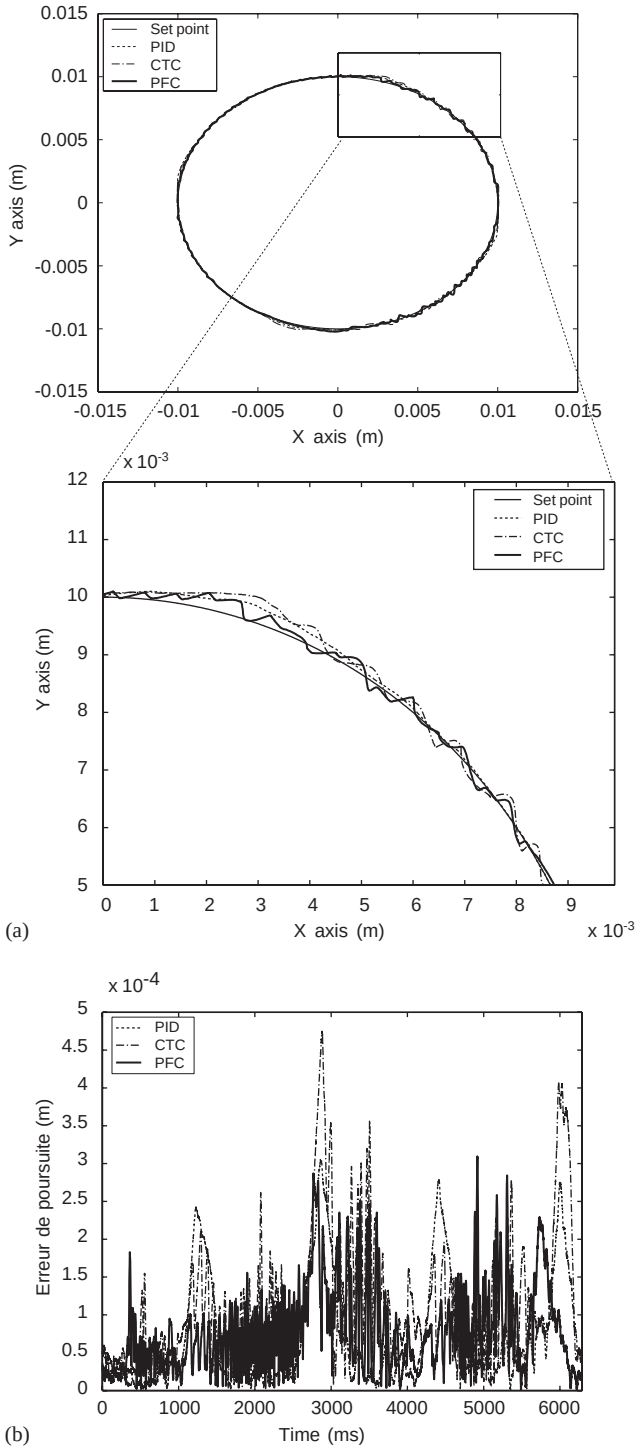

Fig. 9. Experimental results for a circle ($w = 1$ rad/s) (a) obtained and zoomed trajectories; (b) tracking errors.

4.4. Predictive functional control

The non-linear compensation does not indeed supply a double set of integrators due to estimation uncertainties. Both the process and its non-linear compensation

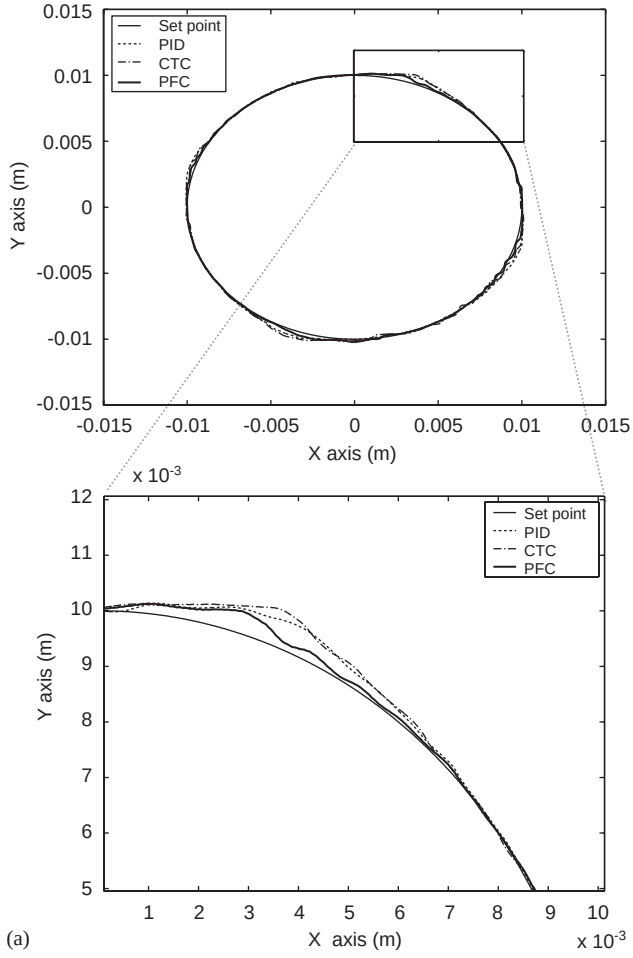

Fig. 10. Experimental results for a circle ($w = 2\,\mathrm{rad/s}$) (a) obtained and zoomed trajectories; (b) tracking errors.

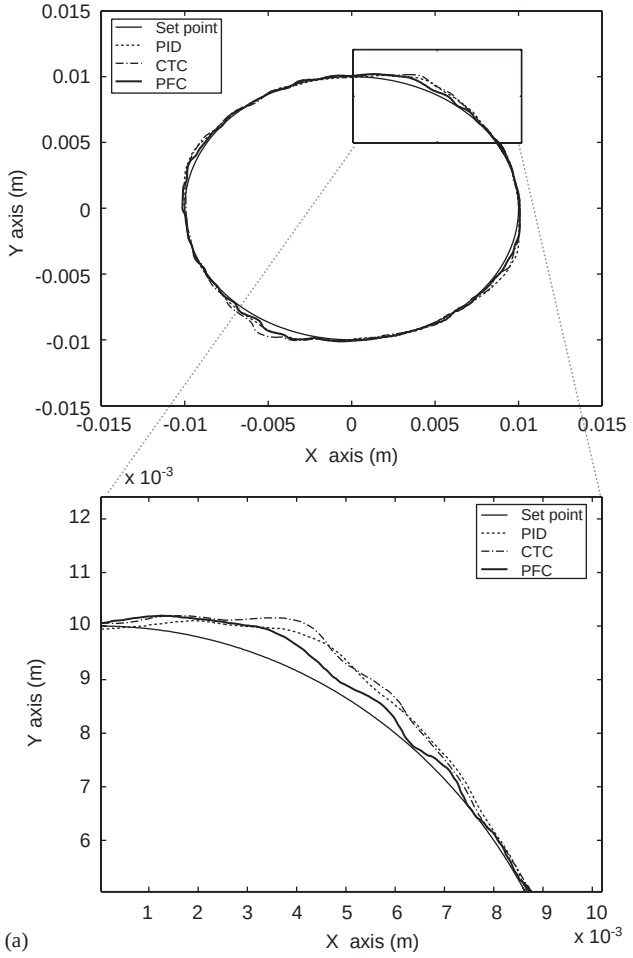

Fig. 11. Experimental results for a circle ($w = 4\,\mathrm{rad/s}$) (a) obtained and zoomed trajectories; (b) tracking errors.

are identified once again. The new linear model is then given by a second-order transfer function:

$$G(s) = \frac{2.7}{s^2 - 52.6s + 54.7}. \tag{43}$$

The process and its non-linear compensation are first stabilized with an inner velocity closed loop with a proportional gain K_v (equal to 70). The PFC is then implemented with the second-order model given by Eq. (43) and its stabilizing gain K_v (Fig. 8).

Three different base functions have been used: step, ramp and parabola. The closed-loop response time is fixed to $20^{*}T_{\text{sampling}}$ in order to ensure a trade-off between tracking performance and robustness. Three coincidence time points on the receding horizon are chosen.

5. Experimental results

The control system is implemented on a single PC (Pentium II, 200 MHz, 256 Mb) running under Windows NT and RTX (real time extension) is used as real time software to ensure a control task periodicity of 1.5 ms.

Different situations are considered in this section to illustrate the performance and robustness of each controller. First, complex trajectories are tracked at different velocities such as a circle and an angular path. Second, responses to external disturbances are shown.

5.1. Performance

The following paths, including inside the robot workspace, will be tracked by the robot:

(i) circular path with diameter $d = 20$ mm and angular velocity $\omega = 1$, 2 and 4 rad/s (that means a circle achieved in 6, 3, and 1.5 s, respectively).

(ii) linear path with a change of direction of 55° and linear velocity $v = 0.012$ m/s and 0.024 m/s (trajectories covered in 6 and 3 s, respectively).

Figs. 9–11 show the results for the circular path; results for the angular path are shown in Figs. 12 and 13. Table 2 points the average tracking errors.

5.2. Robustness

An external disturbance is introduced with a load variation of 4 kg on the nacelle, tested in regulation mode around a given steady state position. Figs. 14–16 show the disturbance rejection and Figs. 17–19 show the required torques in each case.

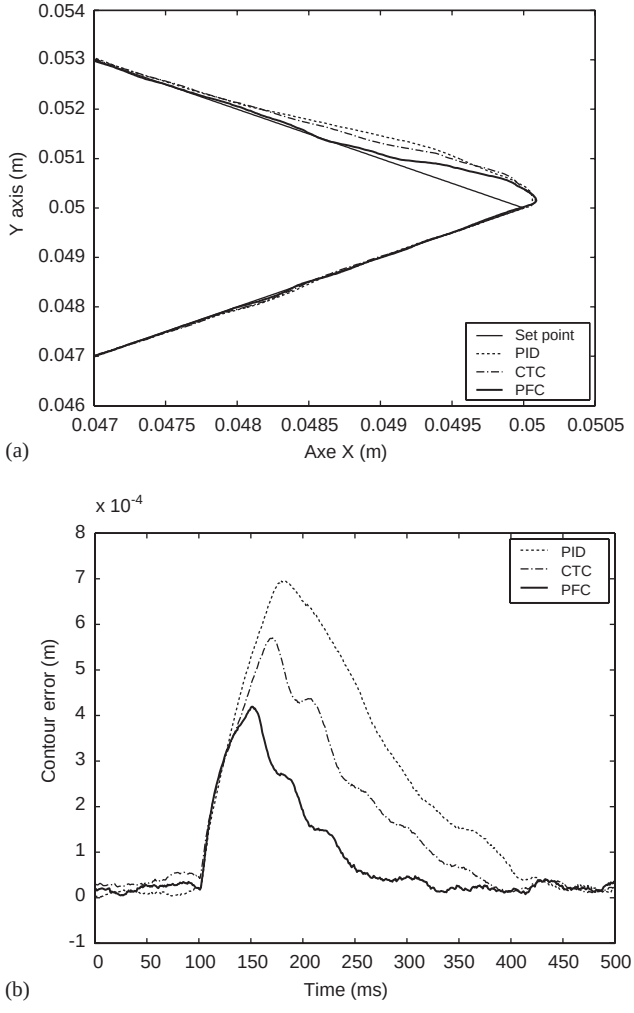

Fig. 12. Experimental results for an angular path ($v = 0.012$ m/s) (a) obtained trajectories; (b) tracking errors.

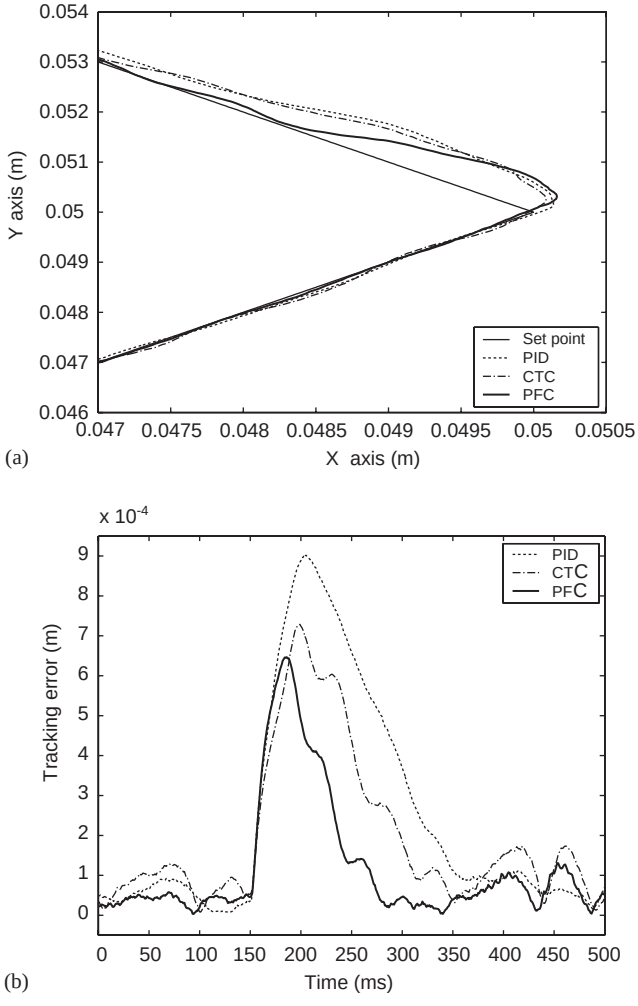

Fig. 13. Experimental results for an angular path ($v = 0.024$ m/s) (a) obtained trajectories; (b) tracking errors.

Table 2
Average tracking errors

	PID (m)	CTC (m)	PFC (m)
Circle ($w = 1$ rad/s)	7.6116e−5	1.1498e−4	8.2559e−5
Circle ($w = 2$ rad/s)	1.4882e−4	1.6047e−4	8.0214e−5
Circle ($w = 4$ rad/s)	2.7094e−4	2.8236e−4	1.7974e−4
Angle ($v = 0.012$ m/s)	1.0743e−4	9.2060e−5	5.0723e−5
Angle ($v = 0.024$ m/s)	8.2959e−5	8.9723e−5	5.8633e−5

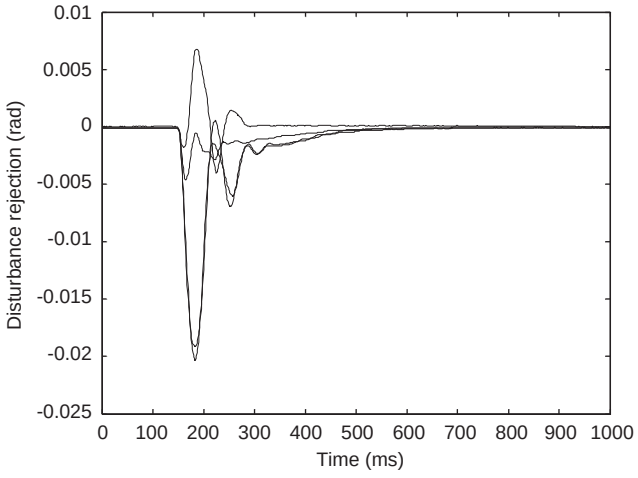

Fig. 14. Output disturbance rejection for the PID.

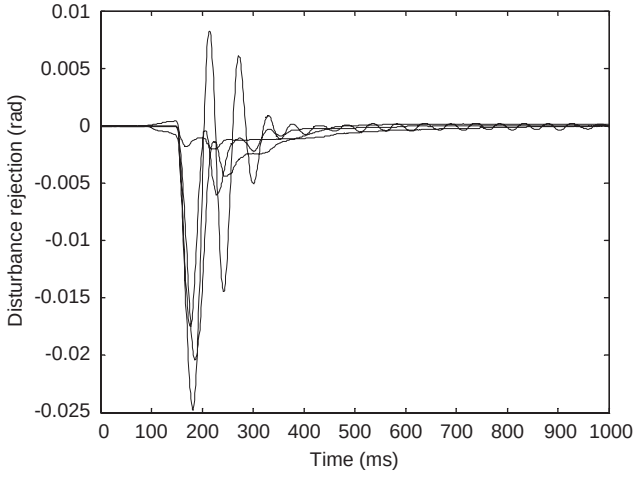

Fig. 15. Output disturbance rejection for CTC.

In all cases, PFC controller yields the smallest contour error, indicating good performance due to its anticipation capacity. The plots in Figs. 10b, 11b, 12b, 13b, show a good response of the PFC controller when the speed becomes higher. The tracking errors are significantly decreased (see also Table 2 giving the average tracking error) and the time response are better.

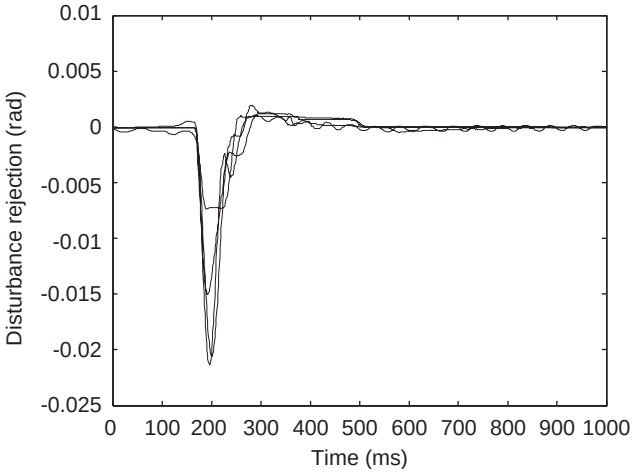

Fig. 16. Output disturbance rejection for PFC.

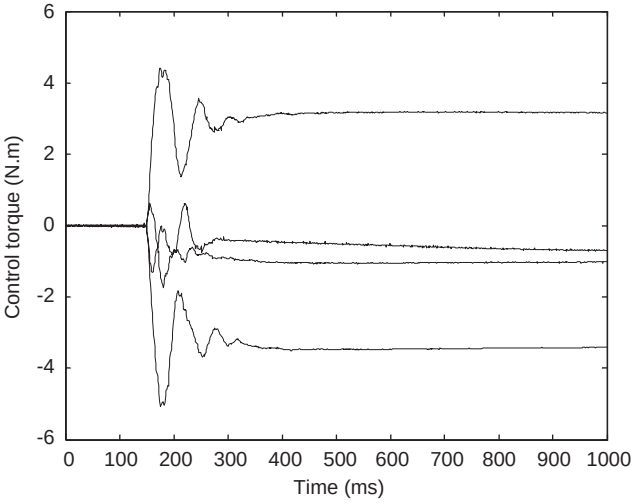

Fig. 17. Torques for output disturbance in PID.

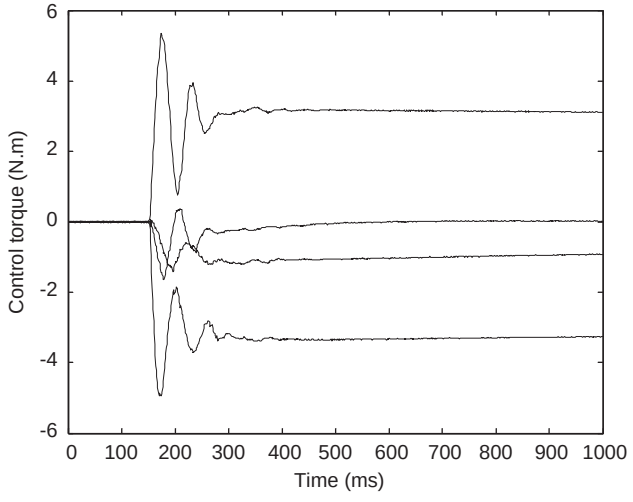

Fig. 18. Torques for output disturbance in CTC.

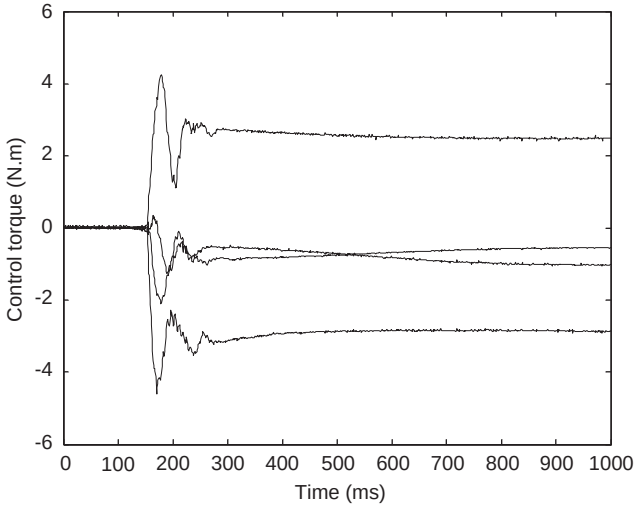

Fig. 19. Torques for output disturbance in PFC.

Figs. 14–16 show the disturbance rejection response. In this case, the PFC controller also shows a well-damped response as well as a faster one.

6. Conclusion

Advanced motion control of a parallel robot has been considered in this study. Experimental tracking performance on complex trajectories and disturbance rejection are compared for three types of controllers: a classical PID strategy and two model-based controllers, that is, the CTC and the PFC. In the model-based strategies, the process is first linearized by feedback. Due to the estimation uncertainties an integrator has been added in the CTC control scheme for reducing the static errors. In case of PFC control scheme, the identification processing has been performed once again after linearization in order to improve the quality of model used in the controller. The PFC controller is applied to the linearized process with a stabilized inner velocity closed loop.

Experimental tasks (circular path, linear path) that are widely used in industrial pick-and-place applications, have shown that predictive functional control has the best performance with respect to the two other control strategies. Further works will concern the implementation of additional sensors in order to directly control the device in the operational space.

Acknowledgements

We acknowledge the financial support extended by the University of Cauca, Popayán, Colombia.

Appendix

$$k_0 = v^{\mathrm{T}} \begin{pmatrix} 1-\alpha^{h_1} \\ 1-\alpha^{h_2} \\ \vdots \\ 1-\alpha^{h_{n_h}} \end{pmatrix}, \quad k_m = v^{\mathrm{T}} \begin{pmatrix} h_1^m \\ h_2^m \\ \vdots \\ h_{n_h}^m \end{pmatrix},$$

$$v_x = -\begin{bmatrix} C_M^{\mathrm{T}}(F_M^{h_1} - I) \\ C_M^{\mathrm{T}}(F_M^{h_2} - I) \\ \vdots \\ C_M^{\mathrm{T}}(F_M^{h_{n_h}} - I) \end{bmatrix}^{\mathrm{T}} v$$

and

$$v = R^{\mathrm{T}} \mathrm{u}_B(0),$$

where

$$R = \left\{ \sum_{j=1}^{n_h} y_B(h_j) y_B(h_j)^{\mathrm{T}} \right\}^{-1} [y_B(h_1)\ y_B(h_2)\ \cdots\ y_B(h_{n_h})],$$

$$y_B = (y_{B_1}\ y_{B_2}\ \cdots\ y_{B_{n_B}})^{\mathrm{T}},$$

$$u_B = (u_{B_1}\ u_{B_2}\ \cdots\ u_{B_{n_B}})^{\mathrm{T}}.$$

References

Abdelghani-Idrissi, M. A., Arbaoui, M. A., Estel, L., & Richalet, J. (2001). Predictive functional control of a counter-current heat exchanger using convexity property. *Chemical Engineering and Processing*, *40*, 449–457.

Allgöwer, F., Badgwell, T.A., Qin, J.S., Rawlings, J.B., & Wright, S. (1999). Non-linear predictive control and moving horizon estimation—An introductory overview. *European Control Conference*, vol. 99, Karlsruhe, Germany.

Canudas de Wit, C., Siciliano, B., & Bastin, G. (1996). *Theory of robot control*. New York: Springer.

Clarke, D. W., Mothadi, C., & Tuffs, P. S. (1987). Generalized predictive control. Part I: the basic algorithm. Part II: Extensions and interpretations. *Automatica*, *23*(2), 137–160.

Clavel, R. (1989). Une nouvelle structure de manipulateur parallèle pour la robotique légère. *APII-JESA*, *23*(6), 501–519.

Company, O., Marquet, F., & Pierrot, F. (2003). A new high-speed 4-DOF parallel robot synthesis and modeling issues. *IEEE Transactions on Robotics and Automation*, *19*(3), 411–420.

Company, O., & Pierrot, F. (1999). A new 3T-1R parallel robot. *ICAR*'99, Tokyo, Japan, (pp. 557–562).

Gough, V.E. (1956–1957). Contribution to discussion of papers on research in automotive stability, control and tyre performance. *Proceedings of the Auto Divisional Institute of Mechanical Engineers*.

Khalil, H. K. (1996). *Non linear systems* (2nd ed.). New York: Prentice Hall.

Khalil, W., & Dombre, E. (2002). *Modeling identification and control of robots*. London: Hermes Penton Science.

Lewis, F. L. (1992). *Applied optimal control & estimation*. Englewood Cliffs: Prentice-Hall.

Merlet, J. P. (1997). *Les robots parallèles* (2nd ed.). Paris: Editions Hermès.

Pierrot, F., Dauchez, P., & Fournier, A. (1991). Fast parallel robots. *Journal of robotics systems*, *8*(6), 829–840.

Pierrot, F., Marquet, F., Company, O., & Gil, T. (2001). H4 parallel robot: modeling, design and preliminary experiments. *Proceedings of the 2001 IEEE international conference on robotics & automation* Seoul, Korea, (pp. 3256–3261).

Pierrot, F., & Shibukawa, T. (1998). From hexa to hexaM. *Proceedings IPK'98: Internationales Parallelkinematik-Kolloquium* Zürich, Switzerland, (pp. 75–84).

Poignet, P., & Gautier, M. (2000). Nonlinear model predictive control of a robot manipulator. *Sixth international workshop on advanced motion control* Nagoya, Japan, (pp. 401–406).

Poignet, P., & Gautier, M. (2001). Extended kalman filtering and weighted least squares dynamic identification of robots. *Control Engineering Practice*, *9*(12), 1361–1372.

Richalet, J. (1993a). *Pratique de la commande prédictive*. Paris: Editions Hermès.

Richalet, J. (1993b). Industrial applications of model-based predictive control. *Automatica*, *29*(5), 1251–1274.

Richalet, J., Abu, E., Arber, C., Kuntze, H.B., Jacubasch, A., & Schill, W. (1997). Predictive functional control. Application to fast and accurate robot. *Tenth IFAC world congress*, Munich, Germany.

Rossiter, J. A. (2002). Predictive functional control: more than one way to prestabilise. *15th IFAC world congress*. Barcelone, Spain.

Stewart, D. (1965). A platform with 6 degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, vol. 180 (Part 1, 15), (pp. 371–386).

Van den Boom, T. J. J., De Vries, R. A. J., Boumeester, S. B., & Verbruggen, H. B. (1994). Constrained predictive control with guaranteed stability and convex optimisation. *Journal A*, *35*(3), 27–34.

Vivas, O.A., Poignet, P. (2003). Model based predictive control of a fully parallel robot. *Seventh international IFAC symposium on robot control—SYROCO 2003* Wroclaw, Poland, (pp. 277–282).

Vivas, O.A., Poignet, P., Marquet, F., Pierrot, F., & Gautier, M. (2003). Experimental dynamic identification of a fully parallel robot. *Proceedings of the 2003 IEEE international conference on robotics & automation* Taipei, Taiwan, (pp. 3278–3283).

Zheng, A. (1997). Stability of model predictive control with time-varying weights. *Computed Chemical Engineering*, *21*(12), 1389–1393.