Model Based Predictive Control of a Fully Parallel Robot

Oscar Andrès Vivas, Philippe Poignet

HAL Id: lirmm-00269468
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00269468v1

Submitted on 3 Apr 2008

MODEL BASED PREDICTIVE CONTROL OF A FULLY PARALLEL ROBOT

O. A. VIVAS, Ph. POIGNET

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), UMR CNRS 5506,
University Montpellier 2
161 Rue Ada, 34392 Montpellier cedex 5, France.
<vivas, poignet>@lirmm.fr

Abstract: This paper deals with an efficient application of a model based predictive control in parallel machines. A receding horizon 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. The model based predictive control and the commonly used computed torque control strategies are compared. The tracking performances and the robustness with respect to external disturbances or model / robot mismatch, are enlightened. Copyright © 2002 IFAC.

Keywords: Model based predictive control, parallel robots, performances, robustness.

1. INTRODUCTION

Parallel mechanisms were introduced by Gough (1957) and Steward (1965). Clavel (1989) proposed the Delta structure, a parallel robot dedicated to high-speed applications, that has intensively used in industry. This is due to the exceptional simplicity of the Delta 3-dof solution. For most pick-and-place applications, at least four dof are required (3 translations and 1 rotation to arrange the carried object in its final location). For the Delta robot, this is achieved thanks to an additional link between the base and the gripper, but it seems not to be as efficient as a parallel arrangement. On the other hand, 6-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 from their limited tilting angle. As an intermediate solution to these drawbacks, a 4-dof parallel mechanism – the H4 robot - have been proposed (Company and Pierrot, 1999; Pierrot et al, 2001). Figure 1 shows a photography of the H4 parallel robot.

Fig. 1. H4 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 high performances. 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 performances of the robot. In the past decade model predictive control (MPC) has become an efficient control strategy for a large number of process (Clarke et al, 1987). Several works have shown that predictive control are of great interest when requiring good performances in term of rapidity, disturbances or errors cancellations (Clarke et al, 1987; Allgöwer et al, 1999).

In this paper, we focus on the implementation of the predictive functional control (PFC) developed by Richalet (Richalet, 1993; Richalet et al, 1997) on the H4 parallel robot. Basically the procedure will consist in two steps i) the process is first linearized by feedback ii) secondly the model predictive control scheme is computed from a linearized model composed of a set of double integrators firstly stabilized with an inner closed loop structure. Experimental results are compared with those obtained from the classical model based computed torque control (CTC) (Canudas de Wit et al, 1996).

The paper is organized as follows : Section 2 is dedicated to the geometric, kinematics and dynamic modelling of the H4 robot required to implement the control strategy. Section 3 details the model predictive functional control. Section 4 introduces the compared control scheme that is the model based predictive control and the commonly used computed torque control. Section 5 exhibits major experimental results in terms of tracking performances and robustness with respect to model / robot mismatch or external disturbance such as load variation. Finally, conclusions are given in section 6.

2. MODELLING

2.1 Geometric and kinematics modelling

The Jacobian matrix and the forward geometric model are required to compute the dynamic model (see section 2.2) (Khalil and Dombre, 2002). Therefore we briefly

present the way of computing the different relationship necessary to obtain these model and matrix. The design parameters of the robot are described on Figure 2 where the following parameters have been chosen:

$\alpha_1 = 0$; $\alpha_2 = \pi$; $\alpha_3 = 3\pi/2$; $\alpha_4 = 3\pi/2$
$\boldsymbol{u_1} = \boldsymbol{u_y}$; $\boldsymbol{u_2} = -\boldsymbol{u_y}$; $\boldsymbol{u_3} = \boldsymbol{u_x}$; $\boldsymbol{u_4} = \boldsymbol{u_x}$

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 $\boldsymbol{A_iB_i}$ segments represent the arms of the robot and $\boldsymbol{P_iB_i}$ the forearm segments. The joint positions are represented by q_i.
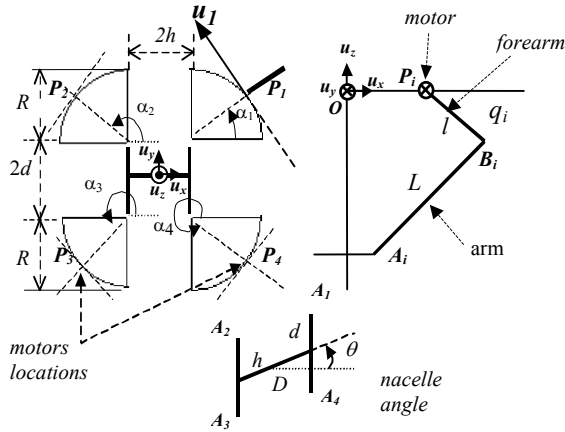

Fig. 2. Design parameters.

To obtain the geometric model, we need 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).

$$\boldsymbol{OD} = \begin{bmatrix} x & y & z \end{bmatrix}^T \qquad (1)$$

The vector that joins the absolute origin O and all of the forearms to the nacelle is:

$$\boldsymbol{OA_i} = \boldsymbol{OD} + \boldsymbol{DA_i} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \boldsymbol{DA_i} \qquad (2)$$

The $\boldsymbol{DA_i}$ segments can be expressed as:

$$\boldsymbol{DA_1} = \begin{bmatrix} h\cos\theta \\ h\sin\theta + d \\ 0 \end{bmatrix} ; \ \boldsymbol{DA_2} = \begin{bmatrix} -h\cos\theta \\ -h\sin\theta + d \\ 0 \end{bmatrix} \qquad (3)$$

$$\boldsymbol{DA_3} = \begin{bmatrix} -h\cos\theta \\ -h\sin\theta - d \\ 0 \end{bmatrix} ; \ \boldsymbol{DA_4} = \begin{bmatrix} h\cos\theta \\ h\sin\theta - d \\ 0 \end{bmatrix} \qquad (4)$$

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

$$\boldsymbol{OB_i} = \boldsymbol{OP_i} + \boldsymbol{P_iB_i} \qquad (5)$$

with:

$$\boldsymbol{P_iB_i} = \begin{bmatrix} l\cos q_i \cos\alpha_i \\ l\cos q_i \sin\alpha_i \\ -l\sin q_i \end{bmatrix} \qquad (6)$$

and actuator locations are:

$$\boldsymbol{OP_1} = \begin{bmatrix} h + R\cos\alpha_1 \\ d + R\sin\alpha_1 \\ 0 \end{bmatrix} ; \ \boldsymbol{OP_2} = \begin{bmatrix} -h + R\cos\alpha_2 \\ d + R\sin\alpha_2 \\ 0 \end{bmatrix}$$

$$\boldsymbol{OP_3} = \begin{bmatrix} -h + R\cos\alpha_3 \\ -d + R\sin\alpha_3 \\ 0 \end{bmatrix} ; \ \boldsymbol{OP_4} = \begin{bmatrix} h + R\cos\alpha_4 \\ -d + R\sin\alpha_4 \\ 0 \end{bmatrix} \qquad (7)$$

Finally, arms coordinates are given by:

$$\boldsymbol{A_iB_i} = \boldsymbol{A_iO} + \boldsymbol{OB_i} \qquad (8)$$

The analytical forward position relationship is difficult to compute. Up to now, the simplest model we've got is a 8th degree polynomial equation. The forward model is then computed iteratively using the classical formula:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{J(x_n, q_n)} [\boldsymbol{q} - \boldsymbol{q_n}] \qquad (9)$$

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

$$\boldsymbol{J} = \boldsymbol{J_x^{-1}} \boldsymbol{J_q} \qquad (10)$$

Where:

$$\boldsymbol{J_x} = \begin{bmatrix} A_1B_{1x} & A_1B_{1y} & A_1B_{1z} & (\boldsymbol{DC_1} \times A_1B_1)_z \\ A_2B_{2x} & A_2B_{2y} & A_2B_{2z} & (\boldsymbol{DC_2} \times A_2B_2)_z \\ A_3B_{3x} & A_3B_{3y} & A_3B_{3z} & (\boldsymbol{DC_3} \times A_3B_3)_z \\ A_4B_{4x} & A_4B_{4y} & A_4B_{4z} & (\boldsymbol{DC_4} \times A_4B_4)_z \end{bmatrix} \qquad (11)$$

$$\boldsymbol{J_q} = diag((\boldsymbol{P_iB_i} \times \boldsymbol{A_iB_i}).u_{mi}) , \ i = 1,...4 \qquad (12)$$

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 modelling

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

If $\boldsymbol{\Gamma_{mot}}$ is the (4x1) actuator torque vector, the basic equation of dynamics can be written as :

$$\mathbf{\Gamma}_{mot} = \mathbf{I}_{mot}\ddot{\mathbf{q}} + \mathbf{J}^T\mathbf{M}(\ddot{\mathbf{x}} - G) \tag{13}$$

where $\mathbf{I}_{mot}$ represents the motor's inertia matrix including the forearm's inertia, $\mathbf{M}$ is a matrix containing the mass of the nacelle and its inertia, $\ddot{\mathbf{x}}$ is the vector of cartesian accelerations, and G the gravity constant. Thanks to the design, the forearm's inertia is taken into account in the motor's inertia.

With:

$$\mathbf{I}_{mot} = \text{diag } [I_{mot1}, I_{mot2}, I_{mot3}, I_{mot4}] \tag{14}$$

$$\mathbf{M} = \text{diag } [M_{nac}, M_{nac}, M_{nac}, I_{bc}] \tag{15}$$

The motor position $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$ are directly measured, and the velocity $\dot{q}$ and acceleration $\ddot{q}$ are obtained by central derivation. As the acceleration measurement $\ddot{\mathbf{x}}$ is not available, $\ddot{\mathbf{x}}$ is computed with:

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{16}$$

where $\mathbf{J}$ depends on x and q, $\dot{\mathbf{J}}$ is computed using a central difference algorithm.

2.3 Identification

The dynamic parameters are estimated using weighted least square techniques. The estimated values, given in Table 1, will be considered as the nominal value during the experiments. More details concerning the identification procedure may be found in (Vivas et al, 2003; Poignet, Gautier, 2001; Canudas de Wit et al, 1996).

Table 1. Estimated parameters.

Physical parameters	Estimated values
I_{mot1}	0.0167 Nm^2
I_{mot2}	0.0164 Nm^2
I_{mot3}	0.0176 Nm^2
I_{mot4}	0.0234 Nm^2
M_{nac}	0.984 Kg
I_{bc}	0.0029 Nm^2

3. PREDICTIVE FUNCTIONAL CONTROL

This section is dedicated to briefly recall the main steps of the model predictive functional control 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, 1993; Richalet et al, 1997).

3.1 Internal Modeling

The model used is a linear one given by :

$$\begin{aligned} x_M(n) &= F_M x_M(n-1) + G_M u(n-1) \\ y_M(n) &= C_M^T x_M(n) \end{aligned} \tag{19}$$

where :

x_M is the state, u is the input, y_M is the measured model output, F_M, G_M and C_M are respectively matrices or vectors of the right dimension.

The problem of robustness because of the poles cancellation by the controller if the system is unstable is usually solved by a model decomposition (Richalet, 1993).

3.2 Reference trajectory

The predictive control strategy of the MPC is summarized on Figure 3. Given the set point trajectory on a receding horizon $[0,h]$, the predicted process output $\hat{y}_p$ will reach the future set point following a reference trajectory y_R.
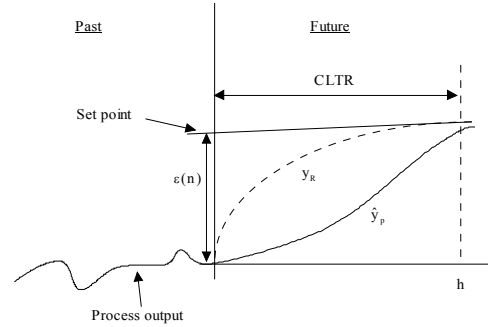

Fig. 3. Reference Trajectory and predictive control strategy.

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

On the prediction horizon, the reference trajectory y_R, which is the path towards the future set point, is given by :

$$c(n+i) - y_R(n+i) = \alpha^i(c(n) - y_p(n)) \text{ for } 0 \leq i \leq h \tag{20}$$

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

The predictive essence of the control strategy is completed included in Eq. 20. Indeed, the objective is to track the set point trajectory following the reference trajectory. This trajectory may be considered as the desired closed loop behaviour.

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 :

$$D(n) = \sum_{j=1}^{n_h} \left\{ \hat{y}_p(n+h_j) - y_R(n+h_j) \right\}^2 \tag{21}$$

where :
n_h is the number of coincident time point, h_j are the coincidence time point on the prediction horizon. The predicted output $\hat{y}_p$ is usually defined as :

$$\hat{y}_p(n+i) = y_M(n+i) + \hat{e}(n+i) \quad 1 \leq i \leq h \tag{22}$$

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 :

$$D(n) = \sum_{j=1}^{n_h}\left\{\hat{y}_p(n+h_j) - y_R(n+h_j)\right\}^2 + \lambda\left\{u(n) - u(n-1)\right\}^2 \quad (23)$$

where u is the control variable.

3.4 Control variable

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

$$u(n+i) = \sum_{k=1}^{n_B}\mu_k(n)u_{Bk}(i) \quad 0 \le i \le h \qquad (24)$$

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. Hereafter we will use :

$$u_{Bk}(i) = i^{k-1} \quad \forall k \qquad (25)$$

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

$$u(n) = \sum_{k=1}^{n_u}\mu_k(n)u_{Bk}(0) \qquad (26)$$

The model output is composed in two parts :

$$y_M(n+i) = y_{UF}(n+i) + y_F(n+i) \quad 1 \le i \le h \qquad (27)$$

where :

y_{UF} is the free output response $(u = 0)$, y_F is the forced output response to the control variable given by Eq. 24.

Given Eq. 19 and Eq.24, it follows :

$$y_{UF}(n+i) = C_M^T F_M^i x_M(n) \quad 1 \le i \le h$$
$$y_F(n+i) = \sum_{k=1}^{n_u}\mu_k(n)y_{Bk}(i) \quad 0 \le i \le h \qquad (28)$$

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

$$\hat{e}(n+i) = e(n) + \sum_{m=1}^{de}e_m(n)i^m \quad \text{for} \quad 1 \le i \le h \qquad (29)$$

where :

de is the degree of the polynomial approximation, e_m are some coefficients calculated 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 :

$$u(n) = k_0\left\{c(n) - y_p(n)\right\} - \sum_{m=1}^{de}k_m e_m(n) + v_x^T x_M(n) \qquad (30)$$

where the gains k_0, k_m, v_x^T are calculated off-line (see 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

4.1 Feedback linearization

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

$$\Gamma = M(q)\ddot{q} + H(q,\dot{q}) \qquad (31)$$

It is well known that the rigid m-link robot equations may be linearized and decoupled by non linear feedback (Khalil, 1996). In fact, given the state space vector and the selected output :

$$x_1 = q, \ x_2 = \dot{q}, \ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ and } y = x_1 ,$$

The direct dynamic model can be written as follows :

$$\dot{x} = Ax + B\beta^{-1}(x)[\Gamma - \alpha(x)] \qquad (32)$$

where :

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \ \beta(x) = M(x_1) \qquad (33)$$

and

$$\alpha(x) = H(x_1, x_2) \qquad (34)$$

Considering a nonlinear feedback given by :

$$\Gamma = \alpha(x) + \beta(x)v \qquad (35)$$

The transfer between v and y is equivalent to :

$$\ddot{y} = v \qquad (36)$$

This is known as the feedback linearized system (Figure 4). It corresponds to the familiar inverse dynamics control scheme which transforms the direct dynamic model into a double set of integrator equations.
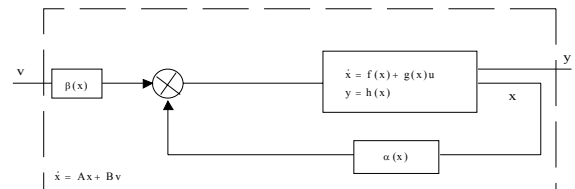

Fig. 4. Feedback linearized system.

Classical linear system control techniques can now be used to design a tracking controller. In the next sections, we will then compare the efficiency of the model predictive control described in section 3 and a classical computed torque control.

4.2 Computed torque control

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

$$v = K_p(q^d - q) + K_v(\dot{q}^d - \dot{q}) + \ddot{q}^d \qquad (37)$$
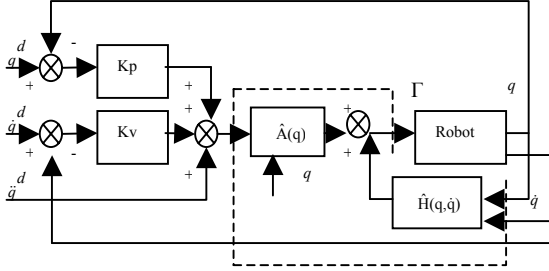
where K_p, K_v are the controller gains.

Fig. 5. Computed Torque Control.

Figure 5 illustrates the computed torque control scheme. The gains tuning leads to Kp=10000, Kv=90 in order to guarantee the torque and dynamic actuator constraints in tracking situation as well as in disturbance rejection.

4.3 Predictive functional control

The MPC is implemented with a second order internal model issued from the double set of integrators and an inner closed loop. Figure 6 shows the model predictive control scheme with the feedback linearization and the inner closed loop with the gains K_p and K_v that stabilize the set of double integrators. Three different base functions are used: step, ramp and parabola. The closed loop response time is fixed to CLTR = 18*$T_{sampling}$ in order to ensure the trade off between the tracking performances and robustness. Three coincidence time points on the prediction horizon are defined.
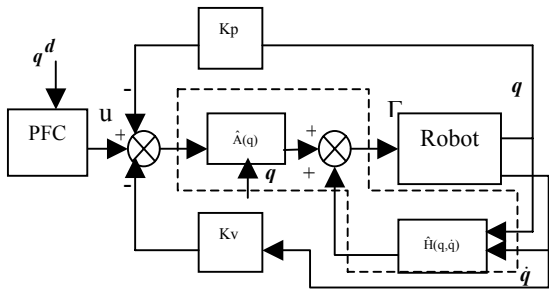

Fig. 6. Model predictive control.

The gains (K_p, K_v) have been tuned to 50 and 68 respectively.

5. EXPERIMENTAL RESULTS

5.1 Tracking performances

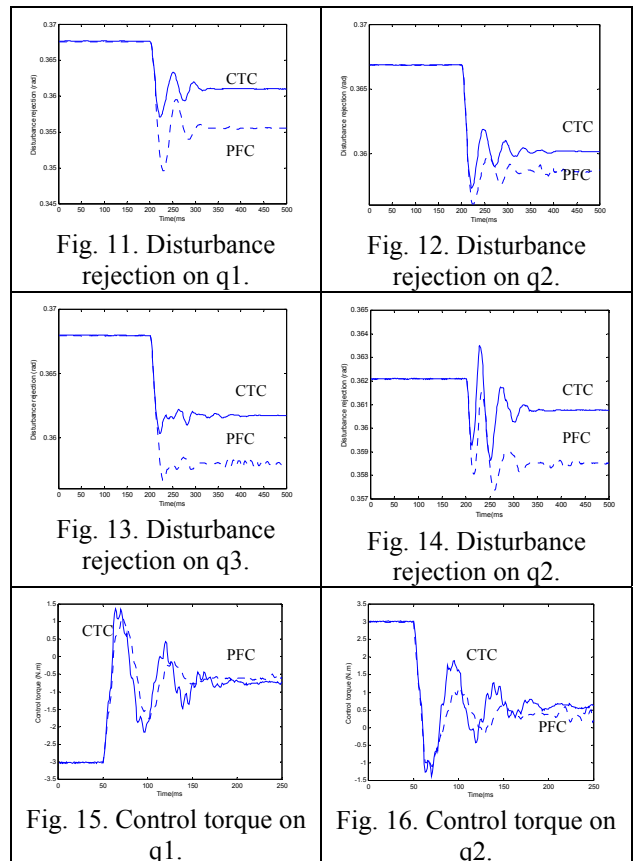These experiments are running within 1ms sampling period. A 5th degree polynomial function is used as joint set point trajectory. A motion of 0.1m is performed along the z-axis. This motion corresponds to a joint angular magnitude of 0.35rad (20°). Figures 7 and 9 show the tracking errors on each active joint. Figures 8 and 10 present the applied torques.

Fig. 7. Tracking error CTC.

Fig. 8. Control torque CTC.

Fig. 9. Tracking error PFC.

Fig. 10. Control torque PFC.
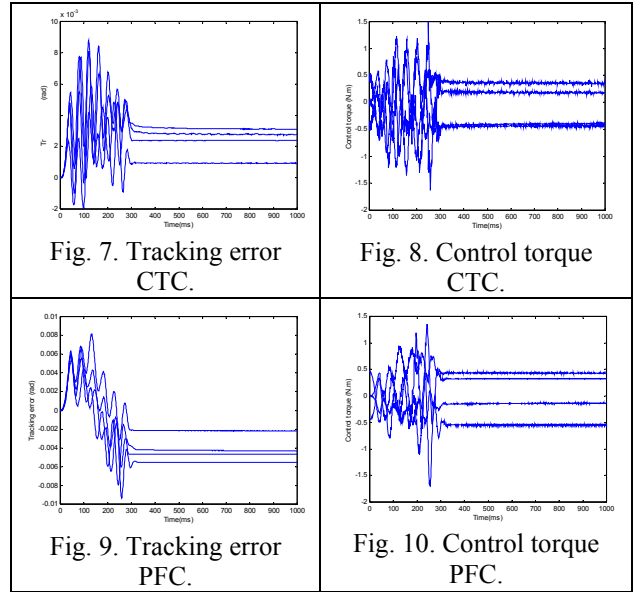
The tracking performances are very similar for both strategies. The torques respect the actuator limitations.

5.2 Disturbance rejection

The output space disturbance is a load variation of 4 kg on the nacelle. Figures 11, 12, 13 and 14 exhibit the influence of the disturbance on each joint. Figures 15, 16, 17 and 18 show the torques.

Fig. 11. Disturbance rejection on q1.

Fig. 12. Disturbance rejection on q2.

Fig. 13. Disturbance rejection on q3.

Fig. 14. Disturbance rejection on q2.

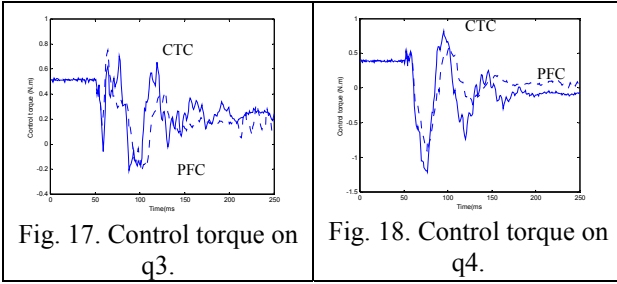Fig. 15. Control torque on q1.

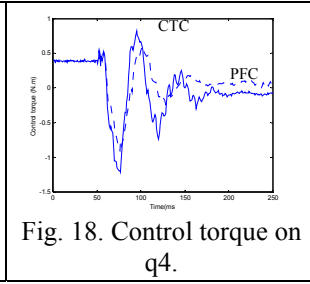Fig. 16. Control torque on q2.

Fig. 17. Control torque on q3.

Fig. 18. Control torque on q4.

With the same controller tuning as for the tracking experiments, these results show the good robustness with respect to load variation for both strategies. The rejection time is equivalent in both cases. However we notice static errors due to mechanism friction and backlash which are not yet considered in the dynamic model.

5.3 Model mismatch robustness

The model mismatch is obtained considering up to 50% errors from the nominal values given in Table 1.
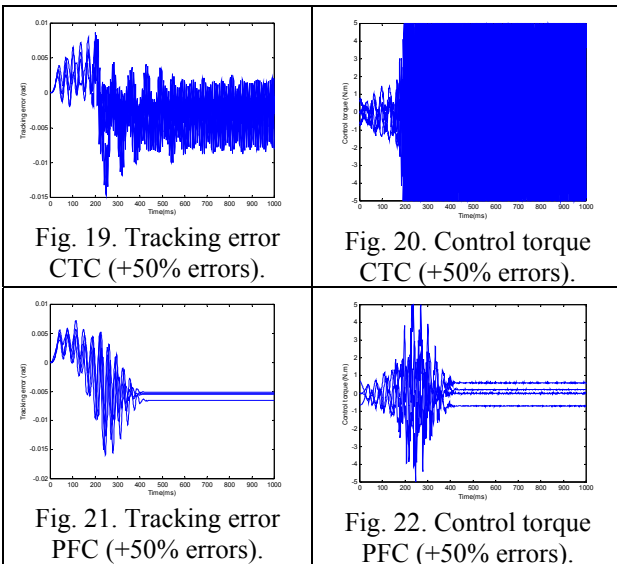

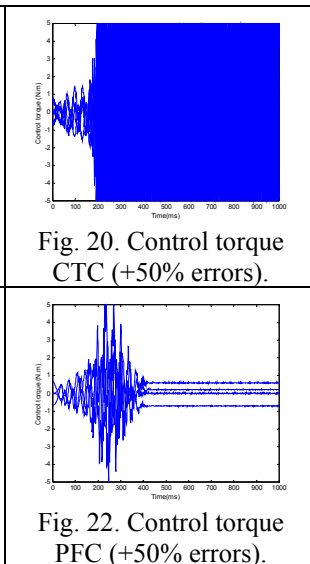Fig. 19. Tracking error CTC (+50% errors).

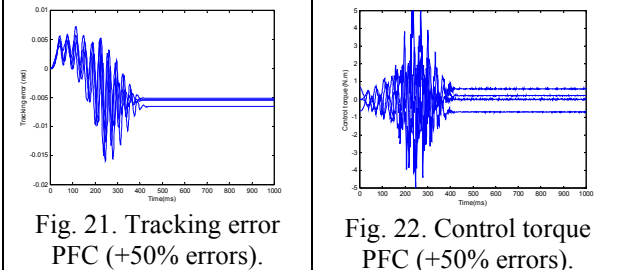Fig. 20. Control torque CTC (+50% errors).

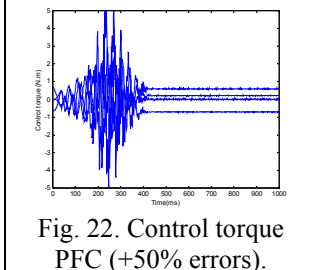

Fig. 21. Tracking error PFC (+50% errors).

Fig. 22. Control torque PFC (+50% errors).

The PFC controller results in case of model mismatch are very advantageous with respect to the CTC controller which saturates the torque inputs.

5. CONCLUSION

This paper exhibits relevant results of the application of model based control strategies. We compare a predictive scheme with the commonly used computed torque control in terms of tracking performances and robustness in case of disturbance rejection such as load variation and model / robot mismatch. The behaviour of both strategies are very similar for the tracking performances or robustness except in case of model mismatch where the CTC controller saturate the torque inputs. Further works will concern the validation of these results on more complex trajectories such as circle and implementation of these strategies in the context of machining tasks.

REFERENCES

Allgöwer F., T. A. Badgwell, J. S. Qin, J. B. Rawlings and S. Wright (1999). Nonlinear predictive control and moving horizon estimation – An introductory overview. *ECC'99, pp. 391-449.*

Canudas de Wit C., B. Siciliano and G. Bastin (1996). *Theory of Robot Control,* Springer Verlag.

Clarke D. W., C. Mothadi and P. S. Tuffs (1987). Generalized predictive control. Part I: The basic algorithm . Part II: Extensions and interpretations. *Automatica, vol. 23, n° 2, pp. 137-160.*

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

Company O. and F. Pierrot (1999). A new 3T-1R parallel robot. *ICAR'99, Tokyo, Japan, October 25-27, pp. 557-562, 1999.*

Gough V. E. (1956-1957). Contribution to discussion of papers on research in automotive stability, control and tyre performance. *Proc. Auto Div., Inst. Mechanical Engineers.*

Khalil H. K. (1996). *Non Linear Systems*, Prentice Hall, 2nd Edition.

Khalil W. and E. Dombre (2002). *Modeling, Identification and Control of Robots.* Hermes Penton Science, London.

Pierrot F., F. Marquet, O. Company and T. Gil (2001). H4 Parallel Robot,: Modeling, Design and Preliminary Experiments. *Proceedings of the 2001 IEEE International Conference one Robotics & Automation, Seoul, Korea, May 21-26, 2001.*

Poignet Ph. and M. Gautier (2001). Extended kalman filtering and weighted least squares dynamic identification of robots. *Control Engineering Practice, vol. 9/12, pp. 1361-1372.*

Poignet Ph. and M. Gautier (2000). Nonlinear Model Predictive Control of a Robot Manipulator. *6th International Workshop on Advanced Motion Control, March 30-April 1, Nagoya, pp. 401-406.*

Richalet J., E. Abu, C. Arber, H. B. Kuntze, A. Jacubasch and W. Schill (1997). Predictive Functional Control. Application to Fast and Accurate Robot. *10th IFAC World Congress, Munich.*

Richalet J. (1993). *Pratique de la Commande Prédictive.* Hermès, Paris.

Stewart D. (1965). A plataform with 6 degrees of freedom. *Proc. of the Ins. of Mech. Engineers, 180 (Part 1, 15), pp. 371-386.*

Vivas O. A., Ph. Poignet, F. Marquet, F. Pierrot and M. Gautier (2003). Experimental dynamic identification of a fully parallel robot . *Submitted to ICRA 2003.*

APPENDIX

$$k_0 = v^T \begin{pmatrix} 1-\alpha^{h_1} \\ 1-\alpha^{h_2} \\ \vdots \\ 1-\alpha^{h_{n_h}} \end{pmatrix}, \quad k_m = v^T \begin{pmatrix} h_1^m \\ h_2^m \\ \vdots \\ h_{n_h}^m \end{pmatrix}, \quad v_x = -\begin{bmatrix} C_M^T(F_M^{h_1}-I) \\ C_M^T(F_M^{h_2}-I) \\ \vdots \\ C_M^T(F_M^{h_{n_h}}-I) \end{bmatrix}^T v$$

$v = R^T u_B(0)$ where :

$$R = \left\{ \sum_{j=1}^{n_h} y_B(h_j) y_B(h_j)^T \right\}^{-1} \begin{bmatrix} y_B(h_1) & y_B(h_2) & \cdots & y_B(h_{n_h}) \end{bmatrix}$$

$y_B = \begin{pmatrix} y_{B_1} & y_{B_2} & \cdots & y_{B_{n_B}} \end{pmatrix}^T$; $u_B = \begin{pmatrix} u_{B_1} & u_{B_2} & \cdots & u_{B_{n_B}} \end{pmatrix}^T$