Predictive Functional Control for a Parallel Robot

Oscar Andrès Vivas, Philippe Poignet, François Pierrot

PREDICTIVE FUNCTIONAL CONTROL FOR A PARALLEL ROBOT

Andrés Vivas Philippe Poignet François Pierrot
LIRMM, UMR 5506,CNRS Université de Montpellier II
161 rue Ada, 34392 Montpellier cedex 5, France.
<vivas, poignet, pierrot>@lirmm.fr

Abstract: This paper presents an efficient application of a model based predictive control in parallel mechanisms. A predictive functional control 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. The tracking performances are enlightened.

1. INTRODUCTION

Parallel mechanisms were introduced by Gough [1] and Steward [2]. Clavel [3] 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 [4], [5]. Fig. 1 shows a photography 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 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 [6]. Several works have shown that predictive control are of great interest when requiring good performances in term of rapidity, disturbances or errors cancellations [6], [7].

Fig. 1. H4 robot

In this paper, we focus on the implementation of the predictive functional control (PFC) developed by Richalet [8], [9], 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 model based computed torque control (CTC) [10] and the clasical PID controller.

The paper is organized as follows: Section 2 is dedicated to the geometric, kinematics and dynamics modelling required to implement the control strategy. Section 3 details the model predictive functional control. Section 4 introduces the compared control schemes: predictive functional control, computed torque control and PID. Section 5 exhibits major experimental results in terms of tracking performances in complex machining trajectories. Finally, conclusions are given in section 6.

2. MODELLING

A. Geometric and kinematics modelling

The Jacobian matrix and the forward geometric model are required to compute the dynamic model (see section 2.B) [11]. 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 Fig. 2 where the following parameters have been chosen:

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

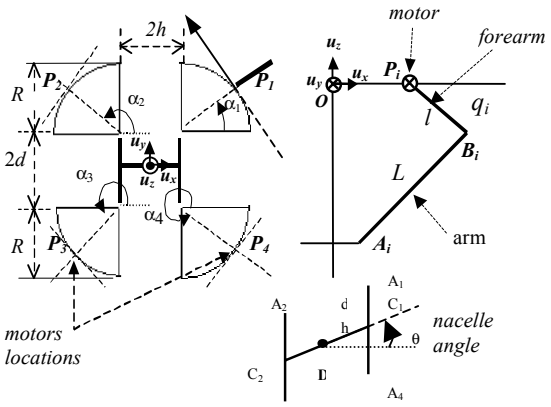

Fig. 2. Design parameters

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. 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.

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:

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

Where 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 [4], [5] :

$$\boldsymbol{J} = \boldsymbol{J}_x^{-1}\,\boldsymbol{J}_q \tag{2}$$

Where:

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

$$\boldsymbol{J}_q = diag((\boldsymbol{P_iB_i} \times \boldsymbol{A_iB_i}) \bullet \boldsymbol{u}_{mi}),\ \ i = 1,\ldots,4 \tag{4}$$

$\boldsymbol{DC_i}$ is the distance between the center of the nacelle and the center of the half lengths of the "H" that forms the nacelle and $\boldsymbol{u}_{mi}$ the unit vector for each direction.

B. 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 [4], [5].

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

$$\boldsymbol{\Gamma}_{mot} = \boldsymbol{I}_{mot}\ddot{\boldsymbol{q}} + \boldsymbol{J}^T\boldsymbol{M}(\ddot{\boldsymbol{x}} - \boldsymbol{G}) \tag{5}$$

where $\boldsymbol{I_{mot}}$ represents the motor's inertia matrix including the forearm's inertia, $\boldsymbol{M}$ is a matrix containing the mass of the nacelle and its inertia, $\ddot{\boldsymbol{x}}$ 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:

$$\boldsymbol{I}_{\text{mot}} = \text{diag}\,[I_{mot1}, I_{mot2}, I_{mot3}, I_{mot4}] \tag{6}$$

$$\boldsymbol{M} = \text{diag}\,[M_{nac}, M_{nac}, M_{nac}, I_{bc}] \tag{7}$$

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

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} \tag{8}$$

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

C. 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 [10], [12], [13].

TABLE I
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 [8], [9].

A. Internal Modeling

The model used is a linear one given by :

$$\boldsymbol{x}_M(n)=\boldsymbol{F}_M\,\boldsymbol{x}_M(n-1)+\boldsymbol{G}_M\,\boldsymbol{u}(n-1) \qquad (9)$$
$$\boldsymbol{y}_M(n)=\boldsymbol{C}_M{}^T\,\boldsymbol{x}_M(n)$$

where :
$\boldsymbol{x_M}$ is the state, $\boldsymbol{u}$ 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.

The problem of robustness because of the poles cancellation by the controller if the system is unstable is usually solved by a model decomposition [9].

B. Reference trajectory

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

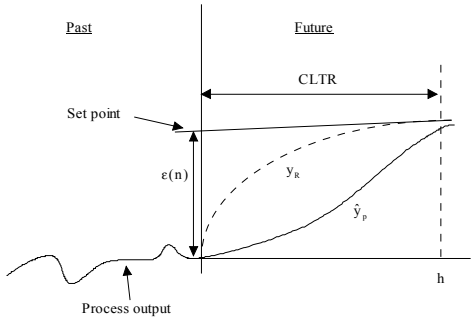

Fig. 3. Reference trajectory and predictive control strategy

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 :

$$c(n+i)-y_R(n+i)=\alpha^i(c(n)-y_P(n)) \quad \text{for } 0\le i\le h \qquad (10)$$

where α (0< α <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. 10. 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.

C. 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 :

$$D(n)=\sum_{j=1}^{n_h}\left\{\hat{y}_P(n+h_j)-y_R(n+h_j)\right\}^2 \qquad (11)$$

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 :

$$\hat{y}_P(n+i)=y_M(n+i)+\hat{e}(n+i) \qquad 1\le i\le h \qquad (12)$$

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 :

$$D(n)=\sum_{j=1}^{n_h}\left\{\hat{y}_P(n+h_j)-y_R(n+h_j)\right\}^2+\lambda\left\{u(n)-u(n-1)\right\}^2 \qquad (13)$$

where u is the control variable.

D. Control variable

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

$$u(u+i)=\sum_{k=1}^{n_B}\mu_k(n)u_{BK}(i) \qquad 0\le i\le h \qquad (14)$$

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 :

$$u_{BK}(i)=i^{k-1} \quad \forall k \qquad (15)$$

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

$$u(n)=\sum_{k=1}^{n_B}\mu_k(n)u_{BK}(0) \qquad (16)$$

The model output is composed in two parts :

$$y_M(n+i) = y_{UF}(n+i) + y_F(n+i) \quad 1 \le i \le h \tag{17}$$

where :
y_{UF} is the free output response (u = 0), y_F is the forced output response to the control variable given by Eq. 14.

Given Eq. 9 and Eq. 14, it follows :

$$y_{UF}(n+i) = \boldsymbol{C}_M{}^T \boldsymbol{F}_M{}^i \boldsymbol{x}_M(n) \quad 1 \le i \le h \tag{18}$$

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

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 :

$$\hat{e}(n+i) = e(n) + \sum_{m=1}^{de} e_m(n) i^m \quad for\ 1 \le i \le h \tag{19}$$

where :
d_e 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 :

$$u(n) = k_0 \{c(n) - y_P(n)\} + \sum_{m=1}^{\max(dc,de)} k_m \{c_m(n) - e_m(n)\} + V_X{}^T x_M(n) \tag{20}$$

where the gains 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

A. PID Controller

A typical PID controller for the H4 robot is shown in Fig 4. The gains tuning leads to K_p=500, K_i=5000, and K_d=6 in order to guarantee the best behavior in tracking situation.

B. Feedback linearization

In order to compute the PFC control strategy [14] 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 :

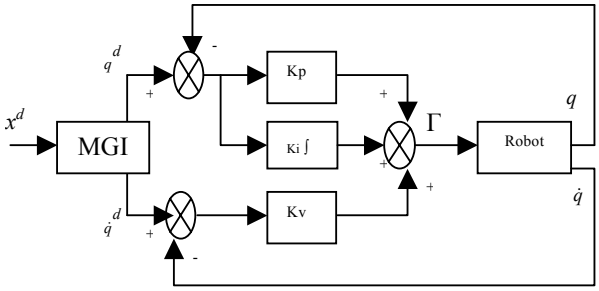

Fig. 4. PID controller

$$\boldsymbol{\Gamma} = \boldsymbol{A}(q)\ddot{q} + \boldsymbol{H}(q,\dot{q}) \tag{21}$$

It is well known that the rigid m-link robot equations may be linearized and decoupled by non linear feedback [15]. In fact, given the state space vector and the selected output :

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

The direct dynamic model can be written as follows :

$$\dot{\boldsymbol{x}} = \boldsymbol{A}x + \boldsymbol{B}\boldsymbol{\beta}^{-1}(x)\left[\boldsymbol{\Gamma} - \boldsymbol{\alpha}(x)\right] \tag{22}$$

where :

$$\boldsymbol{A} = \begin{bmatrix} 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix}; \boldsymbol{B} = \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix}; \boldsymbol{\beta}(x) = \mathrm{M}(x_1); \boldsymbol{\alpha}(x) = \boldsymbol{H}(x_1, x_2) \tag{23}$$

Considering a nonlinear feedback given by (Fig. 5):

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}(x) + \boldsymbol{\beta}(x)\boldsymbol{w} \tag{24}$$

The transfer between w and y is equivalent to :

$$\ddot{\boldsymbol{y}} = \boldsymbol{w} \tag{25}$$

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

Classical control techniques based in a linear model can now be used to design a tracking controller. Computed torque control and predictive functional control that use this linearization will be explained in the next sections.

C. 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 [10] computes the required arm torque. An integrator gain is added to this classical scheme for obtain a disminution of tracking error

due to the differences between the used dynamic model and the real system. The control law becomes:

$$\boldsymbol{w} = K_p(\boldsymbol{q}^d - \boldsymbol{q}) + K_v(\dot{\boldsymbol{q}}^d - \dot{\boldsymbol{q}}) + K_i \int (\boldsymbol{q}^d - \boldsymbol{q}) d\tau + \ddot{\boldsymbol{q}}^d \quad (26)$$

where K_p, K_v, K_i are the controller gains.

Fig. 5 illustrates the computed torque control scheme. The gains tuning leads to K_p=7000, K_v=60, Ki=60000 in order to guarantee the torque and dynamic actuator constraints in tracking situation.

D. Predictive functional control

The MPC is implemented with a second order internal model issued from a weighted least square identification technique [12]. The model identified ($G(s) = 2.7/s^2 - 52s + 54$) is instable. An inner closed loop in velocity is added, with a gain K_v=70, for stabilize the system wich will be the internal model for the predictive control.

5. EXPERIMENTAL RESULTS

These experiments are running within 1.5 ms sampling period. Complex machining tasks trajectories are used as operational set point: a 20 mm radius circle and a change in the direction of a line (55°); paths done in 3 and 6 seconds, velocities of 2 rad/s and 0.012 m/s respectively. These trajectoires show the different performances of the three controllers applied in this parallel mechanism. Other results can be found in [16].

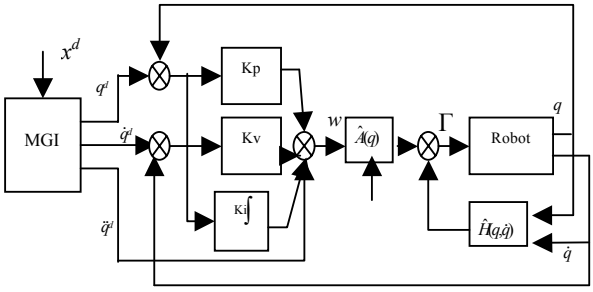

Fig. 5. Computed Torque Control

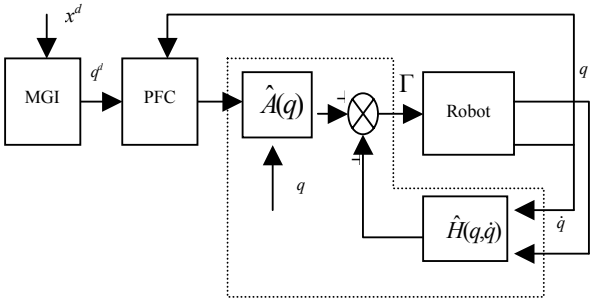

Fig 6. Model predictive control

Fig. 7 shows the circular path and Fig. 8 the tracking error performance. Fig. 9 shows the line path in the angle's change zone and Fig. 10 their tracking error performance.

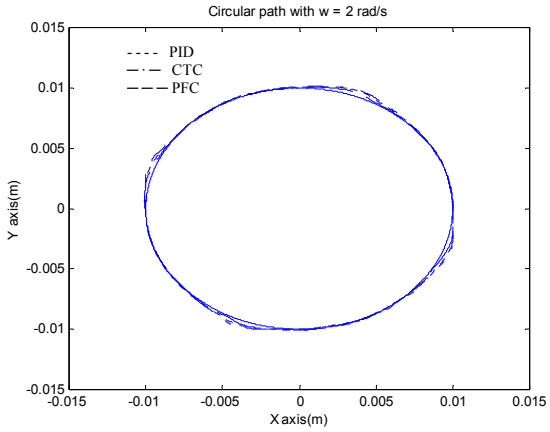

Fig. 7. Circle as a set point

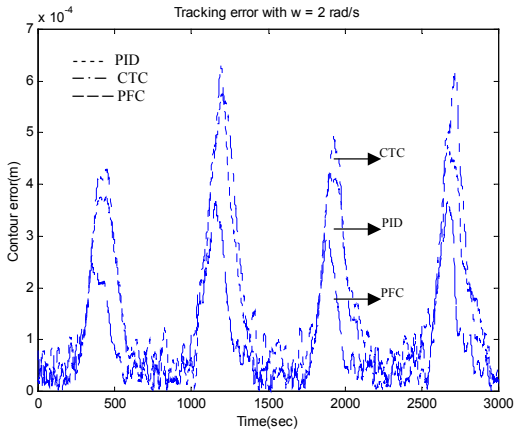

Fig. 8. Tracking performance for the circle

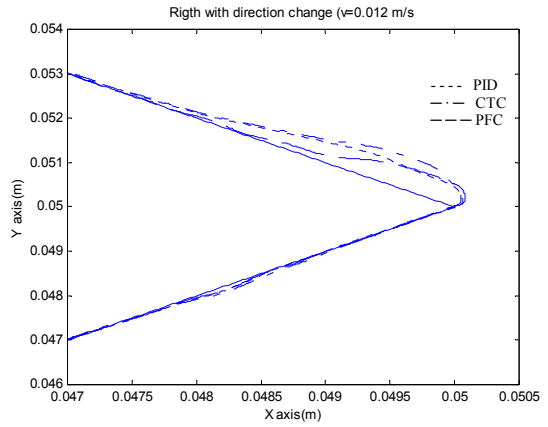

Fig. 9. Lines as a set point

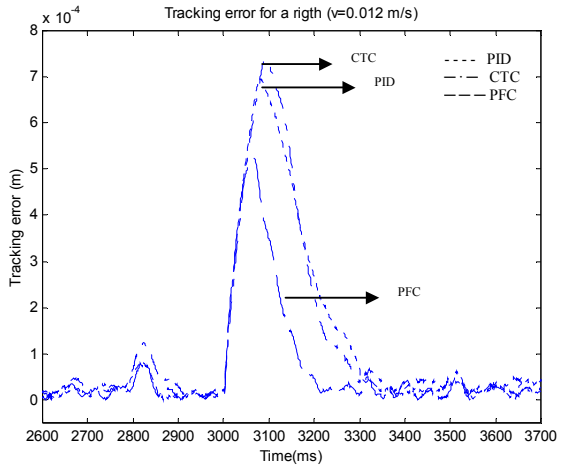

Fig. 10. Tracking performances for the lines

The results show the good performances of the PFC controller in comparision to the CTC and PID control. For both cases, PFC responses are most fast and the tracking error lower. Table II presents this, showing the average tracking errors.

TABLE II
AVERAGE TRACKING ERRORS

Trajectory	PID (m)	CTC (m)	PFC (m)
Circle (w=2 rads/s)	1.4882e-4	1.6735e-4	9.1518e-5
Rigth (v=0.012 m/s)	4.7068e-5	5.4752e-5	3.1372e-5

6. CONCLUSION

This paper exhibits relevant results of the application of model based control strategies. We compared a predictive scheme with the commonly used computed torque control and PID in terms of tracking performances in complex machining trajectories. The behavior of the PFC strategy is better than the CTC and PID controlers. Further works will concern also the analysis with internal and external disturbances as soon as their implementation in machining tasks.

7. REFERENCES

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

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

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

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

[5] F. Pierrot, F. Marquet, O. Company, T. Gil, "H4 Parallel Robot: Modeling, Design and Preliminary Experiments", *Proc. IEEE International Conference one Robotics & Automation*, Seoul, Korea, May 21-26, 2001.

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

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

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

[9] J. Richalet, *Pratique de la Commande Prédictive*, Hermès, 1993.

[10] C. Canudas de Wit, B. Siciliano, G. Bastin, *Theory of Robot Control*, Springer Verlag, 1996.

[11] W. Khali, E. Dombre, *Modeling, Identification and Control of Robots*, Hermes Penton Science, 2002.

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

[13] A. Vivas, Ph. Poignet, F. Marquet, F. Pierrot, M. Gautier, "Experimental dynamic identification of a fully parallel robot", *Proc. IEEE International Conference one Robotics & Automation,* Taiwan, 2003.

[14] Ph. Poignet, M. Gautier, "Nonlinear Model Predictive Control of a Robot Manipulator", *6th International Workshop on Advanced Motion Control*, March 30-April 1, Japan, 2000, pp. 401-406.

[15] H. K. Khalil, *Non Linear Systems*, Prentice Hall, 2nd Edition, 1996.

[16] A. Vivas, Ph. Poignet, "Model based predictive control of a fully parallel robot", *7th International IFAC Symposium on Robot Control*, Poland, 2003.

APPENDIX

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

$$v = R^T u_B(0) \text{ where :}$$

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

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