APPLICATION OF PREDICTIVE CONTROL TECHNIQUES WITHIN PARALLEL ROBOT

Fabian Andres Lara-Molina[*]
`lara@fem.unicamp.br`

João Maurício Rosário[*]
`rosario@fem.unicamp.br`

Didier Dumur[†]
`didier.dumur@supelec.fr`

Philippe Wenger[‡]
`philippe.wenger@irccyn.ec-nantes.fr`

[*]UNICAMP - Mechanical Engineering School, Campinas - SP, Brazil

[†]SUPELEC Systems Sciences (E3S) - Automatic Control Department, Gif sur Yvette, France

[‡]IRCCyN - Institut de Recherche en Communications et Cybernétique de Nantes, Nantes, France

ABSTRACT

Aplicação de Técnicas de Controle Preditivo em Robô Paralelo

Este trabalho apresenta a aplicação de técnicas de controle preditivo para rastreamento de trajetórias de um robô paralelo. A estratégia de controle preditivo generalizado (GPC), que considera o modelo dinâmico linearizado, é usada para melhorar a precisão de rastreamento das trajetórias. O controlador preditivo generalizado é robustificado devido ao ruído de medição e dinâmicas não modeladas utilizando parametrização Youla. É realizada Uma simulação do robô Orthoglide considerando as incertezas dos parâmetros geométricos e dinâmicos, ruído nos sensores e atritos para duas trajetórias diferentes. Finalmente, o controlador GPC robustificado e a técnica de Controle de Torque Computado (CTC) são comparadas. Os resultados das simulações mostram que o controlador GPC robustificado apresenta um melhor desempenho para altas acelerações e também reduz o efeito do ruído no sinal de controle do robô paralelo.

KEYWORDS: Robô Paralelo, Controle Robótico, Controle Preditivo Generalizado

ABSTRACT

This paper addresses the position tracking control application of a parallel robot using predictive control techniques. A Generalized Predictive Control strategy (GPC), which considers the linear dynamic model, is used to enhance the tracking position accuracy. The robustification of GPC against measurement noise and neglected dynamics using Youla parameterization is performed. A simulation of the orthoglide robot considering uncertainties related to geometrical and dynamic parameters, sensors noise and frictions is performed on two different trajectories. Finally, it is compared the robustified GPC controller with the classical Computed Torque Control (CTC). The robustified GPC controller shows a better performance for high accelerations and it also reduces the effect of the noise in the control signal of the parallel robot.

KEYWORDS: Parallel robot, Robot control, Generalized predictive control

1 INTRODUCTION

Parallel robots are based on closed-loop chain mechanisms, thus parallel robots have a movable platform (tool or workpiece gripper) joined to a fixed basis by two or more kinematic chains. The kinematic chains are structures composed of passive articulations and links which allow the relative motion between the movable platform and the fixed basis.

Generally, the actuators are nearby or mounted on the fixed basis deliver the mechanical power to the mechanism.

Due to their mechanical structure, they have some conceptual advantages over serial robots, such as, higher stiffness, accuracy, payload-weight ratio and better dynamic performance. However, they have more kinematic and dynamic complexities than serial robots.

According to their characteristics, the parallel robots have been used widely in industrial applications such as "pick and place" robots (Briot and Bonev, 2010), machine tools (Weck and Staimer, 2002) and precision surgery robots (Wapler et al., 2003) among others.

Although theoretically parallel robots have some potential advantages, it is still difficult to take advantage of them (Abdellatif and Heimann, 2010). To reach a high performance in industrial applications, their dynamical potential advantages should be exploited completely. Consequently, it is essential to reduce the execution time and to increase the accuracy in order to improve the productivity and quality of manipulation and production processes that use parallel robots (Pietsch et al., 2005).

Two factors affect the accuracy of parallel robots. First, passive joints produce kinematic model errors due to clearances and assembly defects (Wang and Masory, 1993). Second, singularities within workspace volume produce a decreasing of the stiffness resulting in a lack of accuracy for a given task, this problem has been addressed through path planning (Dasgupta and Mruthyunjaya, 1998). Therefore, parallel robots still need improvements in design, modeling and control in order to reach their theoretical capabilities (Merlet, 2002). As seen, many works address modeling and design; nevertheless, there are few works related to parallel robot control.

Mainly two control approaches have been considered for parallel robots in literature: dynamic control, which is based on dynamic model of these robots (Paccot et al., 2009), (Wanga et al., 2009); and adaptive control which adjusts the parameters of the system or controller online (Pietsch et al., 2005). Additionally, dynamic control techniques as CTC does not deal very well with modeling errors. They create a perturbation on the error behaviors which may lead to a lack of stability and accuracy (Dombre and Khalil, 2010).

In the other hand, model based predictive control techniques have been applied to parallel robots; Belda et al. (Belda et al., 2003) designed a generalized predictive control law (GPC) for path control of redundant parallel robots; Poignet et al. (Vivas and Poignet, 2005) applied functional predictive control based on the simplified dynamic model of H4 parallel robot; Duchaine et al. (Duchaine et al., 2007) presented a predictive control law for position tracking and velocity control considering the dynamic of the robot. Nevertheless, it is necessary to have robust control laws towards model uncertainties such as measurement noise and parameter uncertainties; in this way, an acceptable behavior of control actions is ensured with an improvement of the dynamic performance of parallel robots.

In this paper, we use GPC to enhance the dynamic performance of a parallel robot in the position tracking control. Then we compare the GPC performance with the classical robot controller: Computed Torque Control (CTC). First, the dynamic equation of the robot is linearized in order to apply the linear control laws. After that, based on the linear model, we apply GPC and CTC control in each actuator of the parallel robot. We robustify the GPC controller toward model uncertainties via Youla parameterization. Finally, we perform the simulation considering uncertainties related to geometric and dynamic parameters, sensors noise and frictions of the complete model of the Orthoglide parallel robot; thus, the performance of CTC and robustified GPC controllers is evaluated in terms of tracking accuracy and control actions characteristics using two different workspace trajectories. We evaluate the behavior of tracking accuracy and disturbance rejection in order to compare the controllers performance.

This paper is organized in five sections. In section 2, CTC controller is presented. Section 3 sumarizes GPC design procedure and the robustification of GPC. Section 4 presents the kinematic and dynamic model of the Orthoglide parallel robot. In section 5, simulations are performed and results are presented. Finally, we present the conclusion and further work.

2 CONTROL OF PARALLEL KINEMATICS MACHINES IN RST FORM

In this section, we present the Computed Torque Control (CTC) of the parallel robots in the joint space. Finally, we introduce the representation of the CTC in the RST form.

CTC is composed of two independent loops: an inner-loop to linearize the non-linear dynamic of the robot and an outer loop to track a desired trajectory. Thus, the non-linear dynamic equation of the robot is considered as follows:

$$\mathbf{\Gamma} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \qquad (1)$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the joint space position, velocities and accelerations trajectories; $\mathbf{\Gamma}$ is the torque or force in the actuators; $\mathbf{A}(\mathbf{q})$ the inertia matrix; and $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ are the centripetal, Coriolis and gravitational vector.

The robot equations may be linearized and decoupled by non-linear feedback. $\hat{\mathbf{A}}(\mathbf{q})$ and $\hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$ are respectively the estimates of $\mathbf{A}(\mathbf{q})$ and $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$. Assuming that $\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{A}(\mathbf{q})$ and $\hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, the problem is reduced to a n linear and decoupled double-integrators system, where n is the number of degrees of freedom of the robot (Khalil and Dombre, 1999).

$$\ddot{\mathbf{q}} = \mathbf{w} \qquad (2)$$

With $\mathbf{w}$ being the new input control vector, the equation (2) corresponds to the inverse dynamic control scheme, where the dynamic of the robot is transformed into a double set of integrators (see Fig. 1). Thus, linear control techniques can then be used to design a tracking position controllers, such as the model-based predictive control (CARIMA model of section 3).

Let us assume that the desired trajectory for each actuator is specified with the desired position $\mathbf{q}^d$, velocity $\dot{\mathbf{q}}^d$ and acceleration $\ddot{\mathbf{q}}^d$. The outer-loop of the controller is:

$$\mathbf{w} = \ddot{\mathbf{q}}^d + \mathbf{K_P}(\mathbf{q}^d - \mathbf{q}) + \mathbf{K_D}\frac{d}{dt}(\mathbf{q}^d - \mathbf{q}) + \mathbf{K_I}\int_0^t (\mathbf{q}^d - \mathbf{q})d\tau \qquad (3)$$

where $\mathbf{K}_P = diag(k, \ldots, k)$, $\mathbf{K_D} = diag(kT_D, \ldots, kT_D)$, $\mathbf{K_I} = diag(k/T_I, \ldots, k/T_I)$.
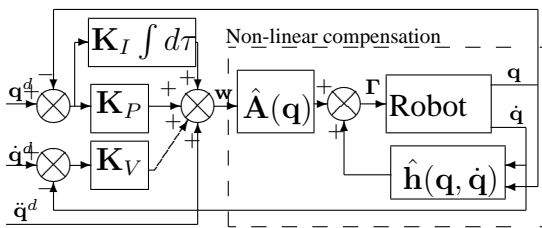

Figure 1: CTC controller, block diagram.

The controller gains are found in order to have in continuous-time domain the following closed-loop characteristic equation for each decoupled double-integrator (s is the Laplace variable)

$$(s + \omega_r)(s^2 + 2\xi\omega_r s + \omega_r^2) = 0 \qquad (4)$$

Thus, $k = (1 + 2\xi)\omega_r^2$, $kT_D = (1 + 2\xi)\omega_r$ and $k/T_I = \omega_r^3$.

The two degrees of freedom RST digital controller is a standard control form in industry (Landau, 1998). CTC controller is expressed in RST form using the Euler transform with sample period T_e and a low pass filter with NT_D time constant for the derivative action, being N a constant. Thus,

$$\begin{cases} S'(z^{-1}) = 1 \\ R'(z^{-1}) = k\dfrac{1 + T_e/T_{Id} + T_{Dd}/T_e}{\cdots} \\ \qquad \dfrac{+[-1 - \nu - T_e/T_{Id}(-1 + \nu) - 2T_{Dd}/T_e]z^{-1}}{1 - \nu z^{-1}} \\ \qquad \dfrac{+[\nu - T_e/T_{Id}\nu + TDd/T_e]z^{-2}}{\cdots} \\ T'(z^{-1}) = R(z^{-1}) \end{cases} \qquad (5)$$

where, z^{-1} is the backward shift operator, $T_{Id} = 2T_I$, $T_{Dd} = \frac{NT_e}{1 + NT_e/2T_D}$ and $\nu = \frac{1 - NT_e/2T_D}{1 + NT_e/2T_D}$. Fig. 2 shows the CTC controller in RST form.
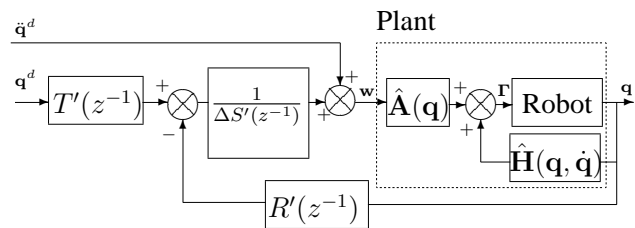

Figure 2: CTC controller in RST form.

3 PREDICTIVE CONTROL TECHNIQUES

This section has two parts. In the first part, the GPC procedure design is presented. In the second part is presented the GPC controller robustification using Youla parameterization.

3.1 GPC Design

This section presents the principles and briefly describes the formulation of GPC law to introduce the design procedure and implementation on the parallel robot. This control technique was developed by Clarke et al. (Clarke et al., 1987). Predictive control philosophy can be summarized as follows: 1) definition of a numerical model of the system to predict the future behavior, 2) minimization of a quadratic cost function over a finite future horizon, using future predicted errors, 3) elaboration of a sequence of future control values, applying only the first value both on the system and the model, 4) iteration of the whole procedure at the next sampling period according to the receding horizon strategy.

In linear GPC theory, the plant is modeled by input/output CARIMA (Controlled AutoRegressive Moving Average) form, which considers increments of input/output signals. As mentioned before, the robot equations have been linearized and decoupled by non-linear feedback, leading to equation (2). This linearized model was then discretized us-

ing the Euler transform and T_e sample period to find $A(z^{-1})$ and $B(z^{-1})$ polynomials of the model.

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + \frac{\xi(t)}{\Delta(z^{-1})} \quad (6)$$

With $u(t)$, $y(t)$ the plant input and output and $\xi(t)$ a centered Gaussian white noise. The control signal is obtained by minimization of the quadratic cost function:

$$J = \sum_{j=N_1}^{N_2} [r(t+j) - \hat{y}(t+j)]^2 + \lambda \sum_{j=1}^{N_u} \Delta u(t+j-1)^2 \quad (7)$$

Where N_1 and N_2 define the output prediction horizons, and N_u defines the control horizon. λ is a control weighting factor, r the reference value, $\hat{y}$ the prediction output value, obtained solving diophantine equation, and u the control signal. The receding horizon principle assumes that only the first value of optimal control series resulting from the optimization of $\delta J/\delta u$ is applied, so that for the next step this procedure is repeated. Thus, the design has been performed adjusting N_1, N_2, N_u, λ to satisfy the required input/output behavour: fastest response consistent with stability requirements; with this control strategy a 2-DOF RST controller is obtained, the procedure is described in (Boucher and Dumur, 1995).

$$S'(z^{-1})\Delta(z^{-1})u(t) = -R'(z^{-1})y(t) + T'(z)r(t) \quad (8)$$
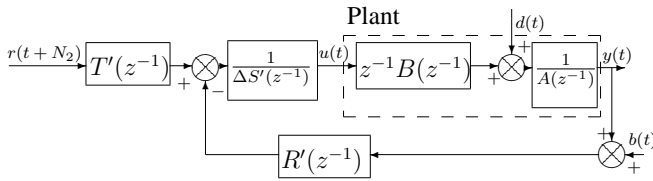
The resulting RST controller is showed in Fig. 3.

Figure 3: RST form of GPC controller.

Where, $b(t)$ and $d(t)$ are the disturbances.

3.2 GPC Robustification

GPC may significantly improve performances in terms of accuracy. Nevertheless, disturbances due to measurement noise or neglected dynamics within the model might affect the robot performance and hence control actions. In this way the robustification of the controllers against these uncertainties is important.

Initially, the procedure starts with GPC design, in our case in RST form. Then, the robustness of the GPC controller is

enhanced via Youla parameterization with regards to model uncertainties, while respecting time domain constraints, such as disturbance rejection. This parameterization allows formulating frequency and time-domain constraints as convex optimization. This optimization problem is approximated by a linear programing with inequality constraints, and the optimal parameters belonging to the research space set.

The set of all stabilising controllers of the system, shown in Fig. 4, which are given follows the Youla parametrization (Kouvaritakis et al., 1992) as follows:

$$\begin{cases} T(z^{-1}) = T'(z^{-1}) - A_o(z^{-1})Q_2(z^{-1}) \\ R(z^{-1}) = R'(z^{-1}) + \Delta(z^{-1})A(z^{-1})Q_1(z^{-1}) \quad (9) \\ S(z^{-1}) = S'(z^{-1}) - z^{-1}B(z^{-1})Q_1(z^{-1}) \end{cases}$$

where $Q_1(z^{-1})$ and $Q_2(z^{-1})$ are stable transfer functions, and $A_oA_c = \Delta AS' + z^{-1}BR'$ the characteristic polynomial of the closed loop of Fig. 3, this characteristic polynomial is split into a control polynomial A_c and an observer polynomial A_o.

Since $Q_2(z^{-1})$ modifies the input/output, to remain this transfer function unchanged, $Q_2(z^{-1}) = 0$. On the other hand, only $Q_1(z^{-1})$ is considered, since this parameter does not modify the input/output transfer function. This parameterization allows formulating frequency constraints (robustness to model uncertainties) and time domain constraints (disturbance rejection) as convex optimization.
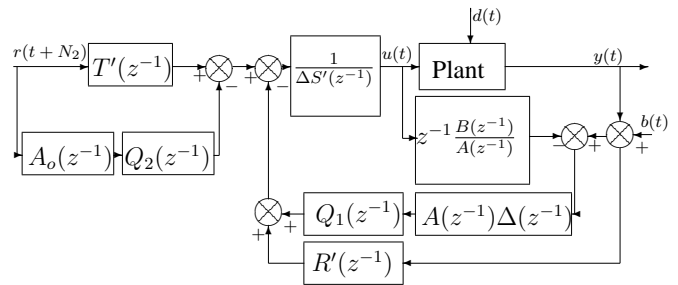

Figure 4: Robustified RST structure.

In this way, we consider the uncertainty block $\Delta_u(z^{-1})$ connected to the system in Fig. 5(a). $\Delta_u(z^{-1})$ represents an unstructured multiplicative direct uncertainty (M'Saad and Chabassier, 1996). The uncertainty block is connected to the $P(z^{-1}) = E(z^{-1})/V(z^{-1})$ system as shown in Fig. 5(b).

where $P(z^{-1})$ is the transfer function seen by the uncertainty block $\Delta_u(z^{-1})$, thus:
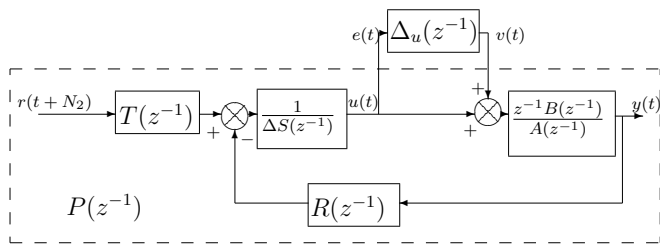
(a) Unstructured multiplicative direct uncertainty.
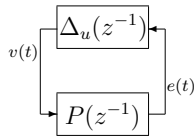
(b) $P(z^{-1})$ system connected to uncertainty block.

Figure 5: System connected to uncertainty.

$$P(z^{-1}) = -\frac{z^{-1}BR'}{A_oA_c} - \frac{z^{-1}B\Delta A}{A_oA_c}Q_1 \qquad (10)$$

The parameter $Q_1(z^{-1})$ results from optimization problem. It takes into account frequencies for which the model has more uncertainties and measurement noise influence than allowed by the $\Phi_{env1}(Q_1)$ in equation (11). Also, time time-domain specification, such as disturbance rejection dynamics, in terms of the transfer disturbance/output is denoted by $\Phi_{env2}(Q_1)$ in equation (11). The robustness is maximized where an H_∞ norm is minimized in the following way

$$\min_{\substack{Q_1 \in \Re H_\infty \\ \Phi_{env1}(Q_1)<0 \\ \Phi_{env2}(Q_1)<0}} \left\| P(z^{-1})W(z^{-1}) \right\|_\infty \qquad (11)$$

where $W(z^{-1})$ is a weighting transfer function which denotes the frequency ranges where model uncertainties are more important.

The disturbances $d(t)$ and $b(t)$ affect the signals $u(t)$ and $y(t)$. In equation (12), the transfer functions are linearly parametrized by Youla parameter Q_1 (Kouvaritakis et al., 1992). These transfer functions must be considered for the time-domain constraint problem.

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} H_{ud} & H_{ub} \\ H_{yd} & H_{yb} \end{bmatrix} \begin{bmatrix} d \\ b \end{bmatrix} \qquad (12)$$

Assuming that $s(t)_{ij}$ is the response of H_{ij} transfer function to a specific input, the time-domain constraint deliver the limits in which $s(t)_{ij}$ must be restricted. The Q_1 parameters that satisfied this constraint are expressed in the following

way

$$C_{env} = \{Q_1/\forall t \geq 0; s_{min}(t) \leq s(t)_{ij} \leq s_{max}(t)\} \qquad (13)$$
$$= \{Q_1/\Phi_{env}(Q_1) \leq 0\}$$

with:

$$\Phi_{env}(Q_1) = \max\left(\max_{t\geq 0}(s(t)_{ij} - s(t)_{max}, s_{min}(t) - s(t)_{ij})\right) \qquad (14)$$

Full developments of the method are given in (Rodriguez and Dumur, 2005).

4 TEST-BED MODEL

4.1 Description

Orthoglide is a parallel robot with three translational degrees of freedom (Fig. 6). The Orthoglide mechanical structure has a movable platform, three prismatic actuators, and three identical kinematic chains PRP_aR (P prismatic, R rotational, P_a parallelogram, see Fig. 6(a)). The actuated joints are the three orthogonal prismatic ones at the fixed basis. The workspace volume is free of singularities and self collisions (see Fig. 6(b)), thus Orthoglide is useful for many tool paths.
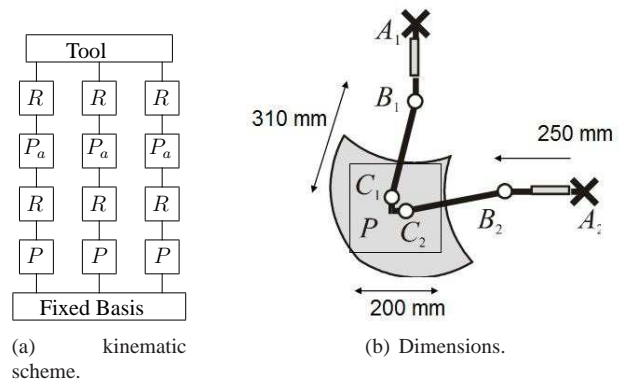

(a) kinematic scheme.

(b) Dimensions.

Figure 6: Orthoglide Robot, description.

Orthoglide robot was designed for high-speed machining, thus the machine reaches a tool velocity of $1.2\,m/s$ and an acceleration of $20\,m/s^2$ at the isotropic configuration, therefore the complete dynamics of the machine can not be neglected in control algorithms to obtain a high position accuracy in the tool (Wenger et al., 2002). Orthoglide robot has two helpful characteristics which facilitate the implementation of a new control law: an analytical expression for the direct kinematic model and no singularities within the workspace.

4.2 Geometric and kinematic modeling

In the model, the active joints are $\mathbf{q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \end{bmatrix}^T$ associated to the linear actuators. The passive joints are $\begin{bmatrix} q_{21} & q_{22} & q_{23} & q_{31} & q_{32} & q_{33} \end{bmatrix}^T$. The Inverse Geometric Model (IGM) delivers the actuator positions $\mathbf{q}$ as function of Cartesian position of end effector $^0\mathbf{p}_P = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T$, such as Fig. 7 presents, thus:

$$\begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix} = \begin{bmatrix} z_p \cos(q_{31})\cos(q_{21})d_{41} - d_{61} \\ x_p - x_{A2} - \cos(q_{32})\cos(q_{22})d_{42} - d_{62} \\ y_p - y_{A3} - \cos(q_{33})\cos(q_{32})d_{43} - d_{63} \end{bmatrix} \quad (15)$$

where:

$$q_{31} = \sin^{-1}\left(\frac{-y_p}{d_{41}}\right); \; q_{21} = -\left(\sin^{-1}\left(\frac{-x_p}{\cos(q_{31})d_{41}}\right) + \frac{\pi}{2}\right)$$

$$q_{32} = \sin^{-1}\left(\frac{-z_p+z_{A2}}{d_{42}}\right); \; q_{22} = -\left(\sin^{-1}\left(\frac{-y_p+y_{A2}}{\cos(q_{32})d_{42}}\right) + \frac{\pi}{2}\right)$$

$$q_{33} = \sin^{-1}\left(\frac{-x_p+x_{A3}}{d_{43}}\right); \; q_{32} = -\left(\sin^{-1}\left(\frac{-z_p+z_{A3}}{\cos(q_{33})d_{43}}\right) + \frac{\pi}{2}\right)$$

The direct geometric model is reduced to a simple equation of second order such as is presented at (Pashkevich et al., 2006):

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} q_{b1}/2 - t_1/q_{b1} \\ q_{b2}/2 - t_1/q_{b2} \\ q_{b3}/2 - t_1/q_{b3} \end{bmatrix} \quad (16)$$

where:

$q_{bi} = q_{bi} + d_{4i}$, for $i = 1, 2, 3$.

$t_1 = \frac{-B-\sqrt{B^2-4AC}}{2A}$, $A = q_{11}^2 q_{12}^2 + q_{11}^2 q_{13}^2 + q_{12}^2 q_{13}^2$, $B = q_{11}^2 + q_{12}^2 + q_{13}^2$ and $C = (q_{11}^2 + q_{12}^2 + q_{13}^2 - 4d_{41}^2)/(q_{11}^2 q_{12}^2 q_{13}^2)/4$

The Inverse Kinematic Model (IKM) delivers actuators velocities $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_{11} & \dot{q}_{12} & \dot{q}_{13} \end{bmatrix}^T$ as function of Cartesian velocity of the end effector $^0\mathbf{v}_P = \begin{bmatrix} \dot{x}_p & \dot{y}_p & \dot{z}_p \end{bmatrix}^T$:

$$\dot{\mathbf{q}} = {}^0\mathbf{J}_P^{-1} \, {}^0\mathbf{v}_P \quad (17)$$

where, $^0\mathbf{J}_P^{-1}$ is the inverse Jacobian matrix of the robot, thus:

$$^0\mathbf{J}_P^{-1} = \begin{bmatrix} -\frac{1}{\tan(q_{21})} & \frac{\tan(q31)}{\sin(q_{21})} & 1 \\ 1 & -\frac{1}{\tan(q_{22})} & \frac{\tan(q_{32})}{\sin(q_{22})} \\ \frac{\tan(q_{33})}{\sin(q_{23})} & 1 & -\frac{1}{\tan(q_{23})} \end{bmatrix} \quad (18)$$

The second order inverse kinematic model gives the actuators accelerations $\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{q}_{11} & \ddot{q}_{12} & \ddot{q}_{13} \end{bmatrix}^T$ as function of the Cartesian platform acceleration $^0\dot{\mathbf{v}}_P$, and actuator velocities $\dot{\mathbf{q}}$:

$$\ddot{\mathbf{q}} = {}^0\mathbf{J}_P^{-1}({}^0\dot{\mathbf{v}}_P - {}^0\dot{\mathbf{J}}_P\dot{\mathbf{q}}) \quad (19)$$
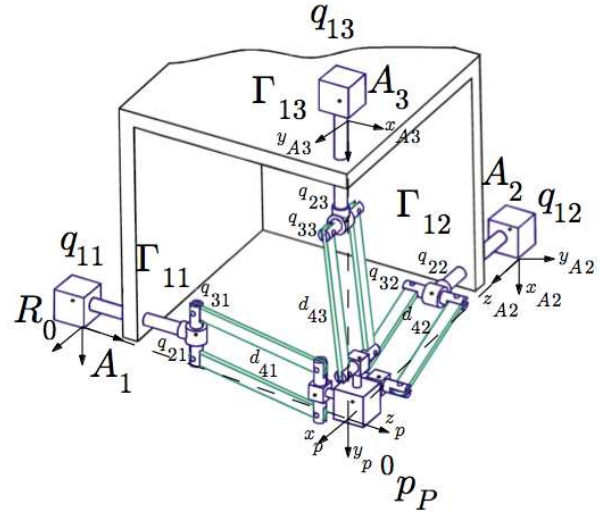

Figure 7: Orthoglide Robot, Geometric CAD.

4.3 Dynamic modeling

The Cartesian accelerations of the end effector $^0\dot{\mathbf{v}}_P$ can be calculated using the direct dynamic equation of the Orthoglide. It is written in the following form (Guegan et al., 2007):

$$^0\dot{\mathbf{v}}_P = \mathbf{A}(\mathbf{q})_{robot}^{-1}[{}^0\mathbf{J}_P^{-T}\mathbf{\Gamma} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})_{robot}] \quad (20)$$

where:

- $\mathbf{A}(\mathbf{q})_{robot} = \sum_{i=1}^3 [\mathbf{A}(\mathbf{q})_{ix}] + \mathbf{I}_3 m_p$ is the total inertia matrix (3x3) of the robot, the inertia of kinematic chains and movable platform.

- $\mathbf{\Gamma} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \end{bmatrix}^T$ is the actuators force vector.

- $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})_{robot} = \sum_{i=1}^3 [\mathbf{h}_{ix}(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{A}(\mathbf{q})_{ix} \, {}^0\dot{\mathbf{J}}_i\dot{\mathbf{q}}_i] - m_p\mathbf{g}$

- $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})_{ix}$ is the Coriolis, gravitation, and centrifuge force vector (3x1). $\mathbf{A}(\mathbf{q})_{ix}$ is the inertia matrix (3x3) of each kinematic chain.

- m_p is the mass of the movable platform, $\mathbf{g} = \begin{bmatrix} 0 & g & 0 \end{bmatrix}^T$ is the gravity vector.

Therefore, the numerical integration of the direct dynamic equation allows to simulate the dynamics of the robot. The dynamic parameters of the robot correspond to masses, inertias and frictions identified in (Guegan et al., 2003).

5 SIMULATION AND RESULTS

In order to establish a framework to compare the performance of the controllers, we simulated the dynamic response of the Orthoglide robot and we use the following controllers CTC and GPC, thus the control laws have been tested in simulation in Matlab / Simulink®.

5.1 Simulation

We simulated the dynamic response of the Orthoglide robot using the following controllers CTC and GPC in order to establish a performance comparison in terms of tracking accuracy and disturbance rejection. The controller parameters were set according to the procedure design presented earlier.

The robot behavior is simulated using the direct dynamic model of the parallel robot in equation (20). Uncertainties about dynamic parameters, errors in geometric parameters (due to assembly tolerances), fictions and Gaussian noise on sensors (variance 1. 10^{-9}) are included in the simulation (Fig. 8).
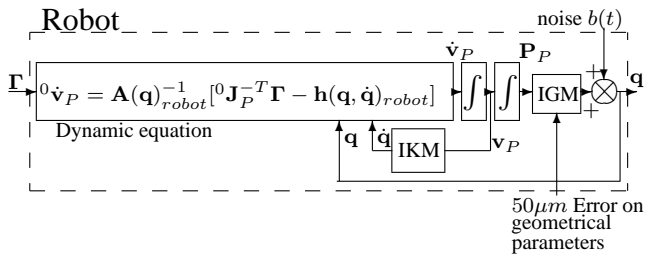

Figure 8: Orthoglide robot, simulation of complete model.

In order to provide a CTC/GPC comparison which makes sense, tuning of both controllers were performed as stated below, looking for the same input/output behavior. Furthermore, the tuning parameters were also adjusted in order to obtain similar frequency features for the open controlled loop (in terms of phase and gain margins in the bode diagram in particular).

CTC controller is tuned using the following parameters: ξ=1 to guarantee response without overshoot and ω_r=50 rad/s obtained experimentally from the parameter identification of the robot; leading to $k = 7500$, $kT_D = 150$, $k/T_I = 125000$ and a filter of the derivative action with $N = 30$. CTC con-

troller is implemented in RST in Fig. 2 using the Euler transform with sample period T_e=2.5 ms. In the same way, GPC was also tuned to obtain the same input/output behaviour with the same bandwith and damping ratio as for CTC, leading to the following set of parameters: N_1=1, N_2=10, N_u=1 and λ=1.10^{-9}. This was assessed by comparing the Bode diagram of the controlled loop. Furthermore robustification of GPC and filtering the derivative effect in the CTC controller leads to the same effect with respect to noise amplification. In that sense, both controllers were tuned in a similar way.
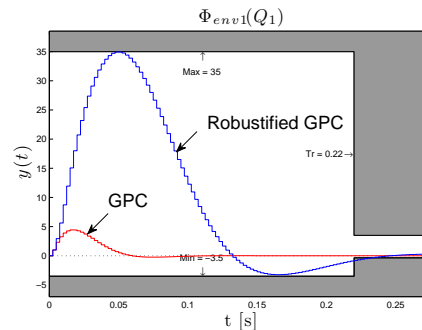
(a) Disturbance rejection.
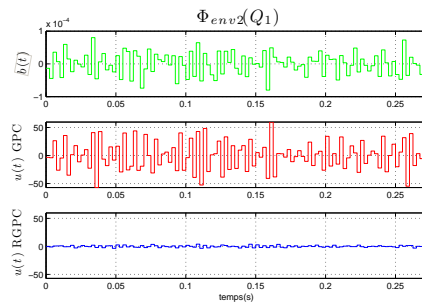
(b) Control and noise.

Figure 9:   Time-domain   constraints $\Phi_{env1}(Q_1)$ and $\Phi_{env2}(Q_1)$.

To robustify the GPC controller, the optimization problem of equation (11) has to be solved satisfying the two time domain constraints: $\Phi_{env1}(Q_1)$ and $\Phi_{env2}(Q_1)$. The constraint $\Phi_{env1}(Q_1)$ corresponds to a time domain template for disturbance rejection $y(t)/d(t)$, the template of this constraint is shown in Fig. 9(a). The constraint $\Phi_{env2}(Q_1)$ corresponds to a time domain constraint for measurement noise/control transfer function $u(t)/b(t)$, this specification restricts the noise effect on control signal, therefore the variation of $u(t)$ is limited to a range of $+/-1$ for this application. Fig 9(b) shows the fixed limits in control signal and the measurement noise. Finally, the selected weighting function is

$$W(z^{-1}) = \frac{1 - 0.6z^{-1}}{0.4} \qquad (21)$$

Another aspect for the comparison of the controllers is the computational complexity. The computational cost of the implementation of inverse dynamic model for the inner loop to linearize the robot dynamics is addressed in (Guegan et al., 2003). The outer loop for CTC and GPC controllers corresponds to the difference equations of each RST polynomial. The RST polynomials of CTC controller are presented in equation (5). The RST polynomials of GPC controller are obtained through an off-line optimization, the order of RST polynomials for this application are 3, 2 and 10 respectively. In the same way the GPC controller is robustified using the off-line optimization of equation (11), thus for this case the order of RST polynomials for this application are 8, 7 and 15 respectively. Consequently, the computational complexity is rather similar for both controllers, the advantage of the RST form being only the need for a few additions and multiplications, which goes very fast in real time.

5.2 Tracking position error

With the purpose of comparing the behavior of CTC and GPC controllers, two workspace trajectories p^d on the $x - y$ plane are used: 1) a triangular one (edge length $= 50\,mm$), with a fifth-degree polynomial interpolation; thus it has a smooth joint space trajectory, at the points where the direction of the trajectory changes the initial acceleration is $1\,m/s^2$ to test the behavior of the controllers (Fig. 10(a)); and 2) a circular one ($\varnothing=50\,mm$, Fig. 10(b)), the initial conditions velocity and acceleration are zero.

Fig 10 presents the respective workspace trajectories using CTC, GPC and RGPC. The highest acceleration on the end effector, for these workspace trajectories, is $5\,m/s^2$.

Initially, we can see that the workspace trajectories are closer to reference using GPC and RGPC controllers (Fig. 11(b)). GPC and RGPC controllers improve the tracking of the workspace trajectory, since with this controller the robot softly follows the abrupt changes in direction, due to the anticipative effect of the predictive control (Fig. 11(a)).

In order to establish the total tracking error of the three actuators of robot over a trajectory the Root Mean Square Error (RMSE) of the actuators is evaluated:
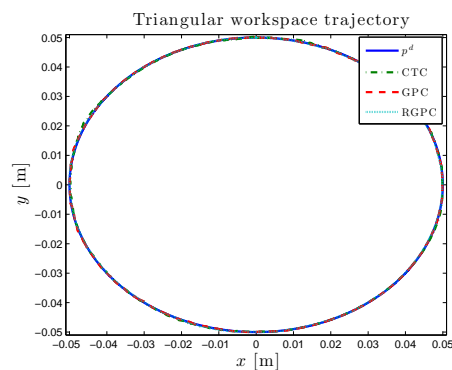
$$RMSE\,(\mathbf{e}) = \frac{1}{3m}\sum_{k=1}^{m}\sqrt{\mathbf{e}(k)^T\mathbf{e}(k)} \qquad (22)$$

where, $\mathbf{e}(k)$ is the error vector of the three actuators for each k instant.

In Fig. 12, the maximum acceleration of the end effector varies from $1\,m/s^2$ to $5\,m/s^2$ for triangular workspace trajectory (Fig. 12(a)) and circular workspace trajectory
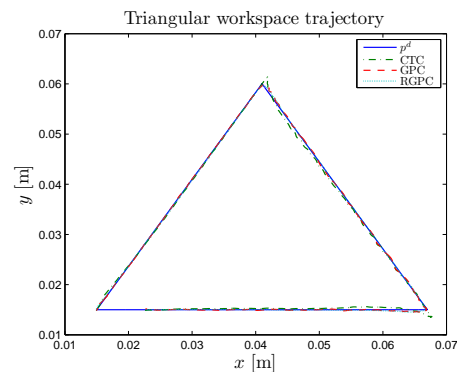
(a) Triangular.

(b) Circular.

Figure 10: Workspace trajectories.

(Fig. 12(b)). For both trajectories, when acceleration increases, tracking accuracy decreases because RMSE increases. However, using the GPC and RGPC controllers, the increase of RMSE is smaller than using CTC controller; thus, the GPC and RGPC offer a better tracking accuracy. Hence, for parallel robots that operate at high accelerations (high dynamics), the GPC and RGPC controllers performance is better than CTC controllers according to the increase of acceleration.

The Fig. 13 shows measurement noise effect in the control signal on actuators for triangular and circular trajectories. The noise affects more the CTC and GPC control signals than the RGPC control signals. The noise in the control signal using CTC and GPC are very high since the nominal actuators force is $400\,N$.

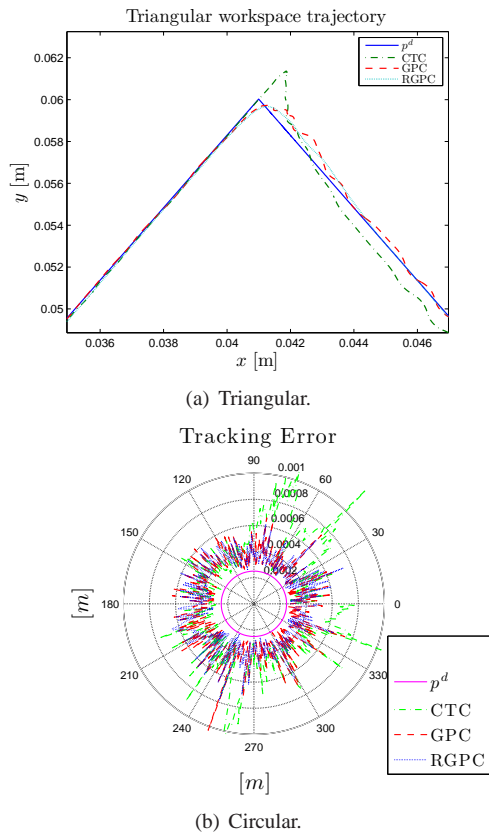The RGPC rejects the noise on the control signal and maintains the tracking position accuracy.

(a) Triangular.

(b) Circular.

Figure 11: Tracking workspace trajectory and errors.

6 DISTURBANCE REJECTION

Fig. 14 shows the joint space error when a step load of 1 Kg is applied in the end effector of the robot. Although the results in terms of tracking position accuracy of the GPG are better than the CTC, robustying GPC against noise decreases the dynamics of the closed-loop, thus transient of disturbance rejection is increased as shown Fig. 14(c). This is the main compromise to be fulfilled using the Youla parameter for robustification: trade-off between noise rejection and performances in terms of disturbance rejection. This trade-off can be adjusted by means of tuning parameters present in equation 11 (mainly envelops definition).

7 CONCLUSIONS

The simulation of Orthoglide parallel robot for a position tracking control using GPC, RGPC and CTC controllers is analyzed. In order to apply these control laws, the parallel robot was decoupled and linearized by feedback; after that, CTC and GPC controllers were applied to the linear equivalent model.
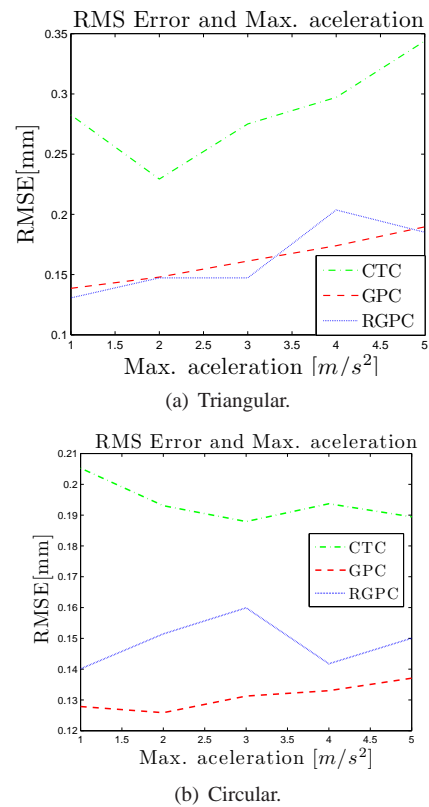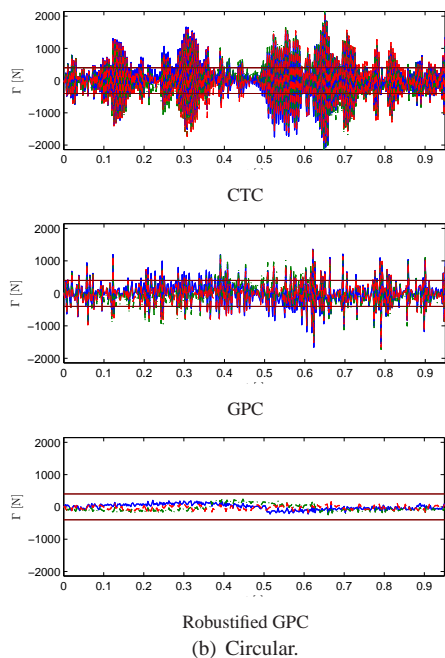

(a) Triangular.

(b) Circular.

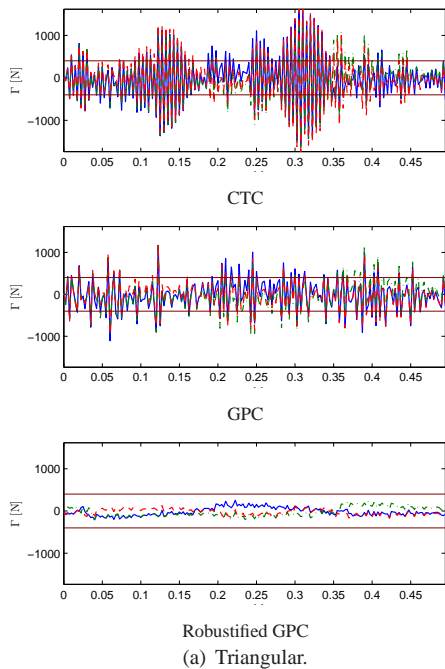Figure 12: RMSE and maximum acceleration.

(a) CTC.

(b) GPC.

(c) RGPC.

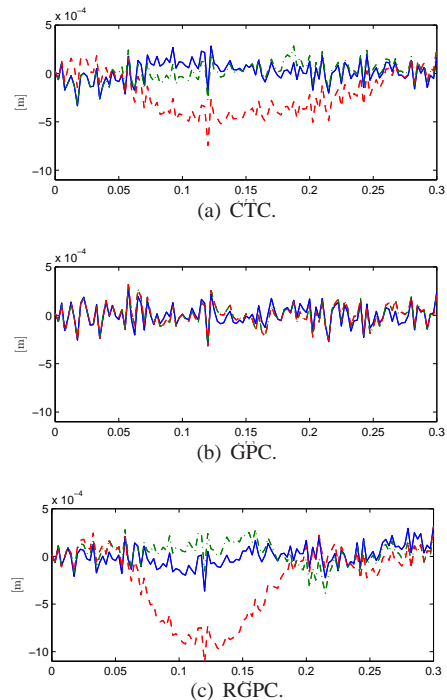Figure 14: Disturbance rejection, error on actuators.

(a) Triangular.

(b) Circular.

Figure 13: Control signal on actuators.

For trajectories typically used in machining, a better performance was obtained in terms of tracking accuracy with respect to parameter variation using GPC and RPGC controllers; thus, we show that the generalized predictive control improves the dynamic behavior of the parallel robot in terms of tracking error over a trajectory with high acceleration. The robustification of GPC significantly reduces noise in the signal control. In this way the robustification of GPC against these uncertainties is important in the parallel robots.

Further works will develop robust predictive control laws considering the complete model of parallel robots without linearization.

ACKNOWLEDGEMENTS

REFERENCES

Abdellatif, H. and Heimann, B. (2010). Advanced Model-Based Control of a 6-DOF Hexapod Robot: A Case Study, *IEEE/ASME Transactions on Mechatronics* **15**(2): 269–279.

Belda, K., Böhma, J. and Valášekb, M. (2003). State-space

generalized predictive control for redundant parallel robots, *Mechanics Based Design of Structures and Machines* **31**(3): 413–432.

Boucher, P. and Dumur, D. (1995). Predictive motion control, *Journal of Systems Engineering* pp. 148–162.

Briot, S. and Bonev, I. A. (2010). Pantopteron-4: A new 3T1R decoupled parallel manipulator for pick-and-place applications, *Mechanism and Machine Theory* **45**(5): 707–721.

Clarke, D., Mohtadi, C. and Tuffs, P. (1987). Generalized predictive control. part i. the basic algorithm, *Automatica* **23**(2): 137–148.

Dasgupta, B. and Mruthyunjaya, T. (1998). Singularity-free path planning for the stewart platform manipulator, *Mechanism and Machine Theory* **33**(6): 711–725.

Dombre, E. and Khalil, W. (2010). *Modeling, Performance Analysis and Control of Robot Manipulators*, Wiley.

Duchaine, V., Bouchard, S. and Gosselin, C. (2007). Computationally efficient predictive robot control, *IEEE/ASME Transactions on Mechatronics* **12**(5): 570–578.

Guegan, S., Khalil, W., Chablat, D. and Wenger, P. (2007). Modelisation dynamique d'un robot parallele a 3-DDL : l'Orthoglide, *0707.2185* . Conference Internationale Francophone d'Automatique (07/2002) 1-6.

Guegan, S., Khalil, W. and Lemoine, P. (2003). Identification of the dynamic parameters of the orthoglide, *IEEE International Conference on Conference on robotics and Automation*, Vol. 3, pp. 3272–3277.

Khalil, W. and Dombre, E. (1999). *Modélisation identification et commande des robots*, Hermes Sciences Publicat.

Kouvaritakis, B., Rossiter, J. A. and Chang, A. O. T. (1992). Stable generalized predictive control: an algorithm with guaranteed stability, *Control Theory and Applications, IEE Proceedings D* **139**(4): 349 – 362.

Landau, I. (1998). The R-S-T digital controller design and applications, *Control Engineering Practice* **6**(2): 155–165.

Merlet, J. P. (2002). Still a long way to go on the road for parallel mechanisms, *ASME 2002 DETC Conference.*

M'Saad, M. and Chabassier, J. (1996). *Commande Optimale. Conception Optimisée des systèmes*, Paris, France: Diderot.

Paccot, F., Andreff, N. and Martinet, P. (2009). A review on the dynamic control of parallel kinematic machines: Theory and experiments, *The International Journal of Robotics Research* **28**(3): 395–416.

Pashkevich, A., Chablat, D. and Wenger, P. (2006). Kinematics and workspace analysis of a three-axis parallel manipulator: The orthoglide, *Robotica* **24**(1): 39–49.

Pietsch, I., Krefft, M., Becker, O., Bier, C. and Hesselbach, J. (2005). How to Reach the Dynamic Limits of Parallel Robots? An Autonomous Control Approach, *IEEE Transactions on Automation Science and Engineering* **2**(4): 369–390.

Rodriguez, P. and Dumur, D. (2005). Generalized predictive control robustification under frequency and time-domain constraints, *IEEE Transactions on Automatic Control Technology* **13**(4): 577 – 587.

Vivas, A. and Poignet, P. (2005). Predictive functional control of a parallel robot, *Control Engineering Practice* **13**(7): 863–874.

Wang, J. and Masory, O. (1993). On the accuracy of a Stewart platform. i. The effect of manufacturing tolerances, *IEEE International Conference on Robotics and Automation.*, Vol. 1, pp. 114–120.

Wanga, J., Wu, J., Wanga, L. and You, Z. (2009). Dynamic feed-forward control of a parallel kinematic machine, *Mechatronics* **19**(3): 313–324.

Wapler, M., Urban, V., Weisener, T., Stallkamp, J., Durr, M. and Hiller, A. (2003). A Stewart platform for precision surgery, *Transactions of the Institute of Measurement and Control* **25**(4): 329–334.

Weck, M. and Staimer, D. (2002). Parallel kinematic machine tools – current state and future potentials, *CIRP Annals - Manufacturing Technology* **51**(2): 671–683.

Wenger, P., Chablat, D. and Majou, F. (2002). L'orthoglide: une machine-outil rapide d'architecture parallele isotrope, *2ème Assises Machines et Usinage à Grande Vitesse* .