Revisiting the major dynamic control strategies of parallel robots
Flavien Paccot, Nicolas Andreff, Philippe Martinet

To cite this version:
Flavien Paccot, Nicolas Andreff, Philippe Martinet. Revisiting the major dynamic control strategies of parallel robots. ECC07 - European Control Conference, Jul 2007, Kos, Greece. pp.4377-4384, 10.23919/ECC.2007.7068791. hal-02468488

HAL Id: hal-02468488
https://inria.hal.science/hal-02468488
Submitted on 5 Feb 2020

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.
Revisiting the Major Dynamic Control Strategies of Parallel Robots

Flavien Paccot\(^1\) \quad Nicolas Andreff\(^{1,2}\) \quad Philippe Martinet\(^{1,3}\)

Abstract—In this paper, we propose a comparison of the three major dynamic control strategies for a parallel robot, namely the linear control, the nonlinear feedforward control and the Computed Torque Control. These control schemes are first transposed in the Cartesian space. This control space is shown to be perfectly relevant for a parallel robot. Then, the control equations are analysed to exhibit the ordinary differential equation of the controlled error. This allows for obtaining a characteristic of the disturbances on this error. Finally, simulation and experiment are provided to verify the theoretical approach.

I. INTRODUCTION

A parallel robot is a complex mechanism with an awkward behaviour [1]. Therefore, controlling such a system is not straightforward. Furthermore, required accuracies and speeds make this task all the more difficult. Concerning the two major applications of the parallel robots, the PAR4 manipulator reaches 170 m/s\(^2\) in pick-and-place operations [2] and 100 µm tolerances are generally required in machining. To reach these requirements, performant mechanical design, modeling and identification should be achieved [3]. Nevertheless, this is only useful when an adapted dynamic control method is used. The presented approach focuses on the three major ones.

The simplest and most widely used way to control a system is a linear PID controller without any a priori knowledge on the system. That is realised for most industrial parallel robot [1]. Nevertheless, this is not enough sufficient with regards to the dynamic behaviour of a parallel robot. Actually, a linear PID controller is efficient with a linear system, which is achieved very locally for the dynamics of a parallel robot. Consequently a simple control scheme only composed of a linear PID controller, without any manipulator structure considerations, can not ensure good tracking performances at high speed [4], [5]. The literature abounds of methods improving this simple control scheme either with a path generation taking into account the dynamics or with an adapted gain tuning [6], [7]. However, tracking performances are low since the still linear control is associated with a nonlinear system. Therefore, a nonlinear control including a model of the dynamic behaviour can be employed.

A first nonlinear method adapts the controller gains with regards to the end-effector pose and the mechanism dynamic behaviour [4]. Nevertheless, a simpler solution is to include an Inverse Dynamic Model in the control scheme, allowing for computing the torques required for the desired motion. This model can be inserted either in a feedforward loop [5] or between the controller and the process, in the case of the well-known Computed Torque Control [8].

To sum up, there are three major dynamic control strategies (linear control, nonlinear feedforward control and Computed Torque Control) on which more elaborated control can be built: gain tuning, performant path generation, adaptive control, robust control...

In this paper, we propose a comparison between the three major dynamic control strategies for parallel robots. This approach is based on a theoretical analysis of the control equations, where the dynamics of the controlled error are exhibited and discussed. The characteristics of the disturbance on these errors are thus obtained. The comparison is performed both in simulation and in experiments. Furthermore, we propose an improvement of the control schemes by employing the Cartesian space control. We show that this control space is more relevant for parallel robots than the traditionally used joint space.

This article is organised as follows: Section II is devoted to control, Section III presents the test-bed, Section IV concerns the simulations and experiments results.

II. MAJOR DYNAMIC CONTROL STRATEGIES REVISITED

This section focuses on three major dynamic control strategies met in the literature: simple linear control, nonlinear feedforward and Computed Torque Control. First of all, it is explained why a Cartesian space control is perfectly relevant for a parallel robot. Then, each control schemes are detailed and adapted in this control space.

A. Cartesian space control

Since the knowledge on parallel robots flows from the serial ones, control schemes used for the latter are generally reused without any adaptation. Therefore, in most cases, parallel robots are controlled in the joint space [5], [4], [7]. Nevertheless, next words show that this is not a relevant control space for such robots.

Firstly, parallel and serial robots present a duality [9]. Whereas the state of serial robot is completely defined by its joint positions (algebraic Forward Kinematic Model), the behaviour of a parallel robot is only ruled by the end-effector pose, except in some rare cases where the Inverse Kinematic Model has no closed-form expression [10]. Consequently, a model-based control for a parallel robot, done in the joint space, requires forward transformations from joint to Cartesian space. In the parallel robots case, these transformations have not a closed-form expression contrary to the serial
The Cartesian space control brings back the end-effector pose to its reference (see Figure 1). Thirdly, considering the complex structure of a parallel robot, the joint motion has to be compatible with the end-effector one. Whatever a suitable joint trajectory can be planned with the Inverse Kinematic Model, the joint tracking errors are not necessarily compatible with the end-effector motion in the joint space control case. Consequently, internal torques can appear, like in the two-arm robots case (see Figure 2 and [13]). By servoing the end-effector pose, the active joint torques are filtered throughout the Forward Instantaneous Kinematics matrix. Consequently a Cartesian space control cancels the internal torques leading to better accuracy and structure damage preventing [13].

Last but not least, the Cartesian space is the user’s task space where trajectories are planned in Computer Aided Manufacturing. Consequently, to our mind, a parallel robot should be controlled in the Cartesian space instead of the joint space. Nevertheless, this assumes to have a fast measure of the end-effector pose, in a dynamic control context. However, speed and accuracy should be improved to meet required performances for a parallel robot. While waiting for a reliable, fast, accurate and flexible measurement mean, the forward kinematic model can be used to give an estimation of the end-effector pose. In special cases [17], [18], parallel robots have a closed-form forward kinematic model avoiding the numerical estimation issues listed above. In the other cases, adapted computation methods can be employed [19]. Furthermore, by adding extra sensors, the metrological redundancy decreases computation time and the number of solutions [20], [21]. Nevertheless, a model can not compete with a measure. Indeed, it can be biased with structure deformations and assembly errors, which have a great influence with regards to the numerous joints [22]. A kinematic identification can reduce this issue [23], [1].

In the further paragraphs, a measure of the end-effector pose is assumed to be available. Therefore, all presented control schemes are Cartesian space controls.

**B. Dynamic modeling**

To analyse the presented control schemes, the inverse dynamic modeling of a parallel robot is used. The Lagrangian formulation is generally employed to set up dynamic control laws. In the serial case, it is expressed as:

\[ \Gamma(q, \dot{q}, \ddot{q}) = A(q)\ddot{q} + H(q, \dot{q}) + G(q) + \Gamma_f(\dot{q}) + \Gamma_{ext} \]  

where \( q \) are the active joint variables, \( \Gamma \) are the torques applied in the active joints, \( A \) is the inertia matrix, \( H \) is a vector containing Coriolis and centrifugal forces, \( G \) is a vector containing the gravity forces, \( \Gamma_f \) is the friction influence and \( \Gamma_{ext} \) are the external forces applied on the robot.

This expression is generally directly reused for parallel robot. However, the dynamic behaviour of a parallel robot depends on the end-effector pose and time derivatives [24], [25]. It is hence more relevant to express the model above as [25]:

\[ \Gamma(X, \dot{X}, \ddot{X}) = A(X)\ddot{X} + H(X, \dot{X}) + G(X) + \Gamma_f(\dot{X}) + \Gamma_{ext} \]  

where \( X \) is the end-effector pose.

The Cartesian space formulation allows for computation saving [25] leading to faster, more accurate and more stable control. In the further paragraphs, this formulation is used.

**C. Linear control**

In most cases, industrial parallel robots are controlled with a simple linear control [1], [5]. This control scheme is well-known allowing for an easy maintenance. In addition, it can be used for a parallel robot as well as for a serial one. We propose an adaptation of this control scheme in the Cartesian space (see Figure 3). Nevertheless, such a control scheme does not allow for good dynamic accuracy [26], [5]. In fact, a desired position in the workspace can be reached with accuracy, with the prerequisite that a good kinematic identification is performed. However, the tracking
performances deteriorate quickly when the trajectory speed increases.

Indeed, by using a simple linear control, the hypothesis of a linear dynamic behaviour is made:

$$\Gamma(X, \dot{X}, \ddot{X}) = M\ddot{X} + F_v\dot{X} + F_s + \dot{\Gamma}_{PID}$$

(3)

where $M$ is the simplified inertia matrix, $F_v$ is the viscous friction matrix, these two matrices are supposed to be constant and diagonal, $F_s$ is the dry friction vector. The term $\dot{\Gamma}_{PID}$ is the resulting error between Equation 2 and Equation 3. It can be expressed as:

$$\dot{\Gamma}_{PID} = (A(X) - M)\ddot{X} + H(X, \dot{X}) + G(X) + \Gamma_f(\dot{X}) - F_v\dot{X} - F_s + \Gamma_{ext}$$

(4)

A linear feedforward term and a friction compensation can be used. The torque control signal is then expressed as:

$$\Gamma(X_d, \dot{X}_d, \ddot{X}_d) = M\ddot{X}_d + F_v\dot{X}_d + F_s + Mu_{PID}$$

(5)

where $X_d$ is desired trajectory, $u_{pid}$ is the signal generated by the linear controller:

$$u_{pid} = K_p\dot{e} + K_p\dot{e} + K_i \int \dot{e} dt$$

(6)

where $e = X_d - X$ is the error signal. Since dry friction depends on the speed sign, let us remark that this feedforward is not completely linear.

Considering Equations 3 and 5, the error signal has the following behaviour:

$$\dot{e} + K_p\dot{e} + K_p\dot{e} + K_i \int \dot{e} dt = M^+\dot{\Gamma}_{PID}$$

(7)

The gain tuning aims at obtaining a system with a triple negative pole, $(p + \omega)^{-3}$, with $\omega$ chosen smaller than the mechanical resonance frequency of the robot [8]:

$$K_p = 3\omega^2$$

(8)

$$K_v = 3\omega$$

$$K_i = \omega^3$$

where $K_p$ is the proportional gain matrix, $K_d$ the derivative gain matrix and $K_i$ the integral gain matrix. Consequently, under the hypothesis of negligible disturbances $\dot{\Gamma}_{PID}$, the error signal presents an exponential decay determined by the gain tuning.

With the linear behaviour and small disturbance hypotheses, a good dynamic accuracy can be expected. Nevertheless, the dynamics of a parallel robot are strongly nonlinear especially because of a dynamic coupling between legs [11], [24], [25]. Therefore, the first hypothesis is achieved only very locally and at low speed. In addition, the disturbances $\dot{\Gamma}_{PID}$ are mainly generated by the dynamic coupling (see Equation 4) where the first two terms concern the dynamic coupling between legs). In this case, the second hypothesis is also only verified at low speed. In conclusion, the hypotheses are too restrictive, grounding a linear control approach unable to meet high speed and high accuracy requirements. To improve the dynamic accuracy, nonlinear control taking into account the dynamic behaviour of the robot needs hence to be used.

D. Nonlinear feedforward control

In order to obtain good tracking accuracy at high speed, the control scheme should include a model of the robot dynamics. The aim is a linearisation of the dynamic behaviour of the robot by a compensation of the latter. Therefore, a simple linear controller on the linearized dynamics can thus ensure optimal performances. In the parallel robot case, the compensation of the robot dynamics in the feedback loop is difficult. Indeed, it depends on the end-effector pose, which is generally estimated with heavy computation. Consequently, Honegger proposed a joint space nonlinear feedforward control, compensating the robot dynamics with regards to the only reference Cartesian trajectory [5].

Here, an adaptation of this control scheme in the Cartesian space is proposed (see Figure 4) and an analysis of the control equations. Using the Lagrangian formulation, the estimated dynamic behaviour of the robot can expressed as:

$$\Gamma(X, \dot{X}, \ddot{X}) = \tilde{A}(X)\ddot{X} + \tilde{H}(X, \dot{X}) + \tilde{G}(X) + \tilde{\Gamma}_f(\dot{X}) + \tilde{\Gamma}_{ext}$$

(9)

where $\tilde{A}$ is the estimated inertia matrix, $\tilde{H}$ a vector containing the estimated Coriolis and centrifugal forces, $\tilde{G}$ the estimated gravity forces and $\tilde{\Gamma}_f$ the estimated friction. The term $\tilde{\Gamma}$ is the resulting error between Equation 2 and Equation 9. It can be expressed as:

$$\tilde{\Gamma} = (A(X) - \tilde{A}(X))\ddot{X} + \tilde{H}(X, \dot{X}) - \tilde{H}(X, \dot{X}) + G(X) - \tilde{G}(X) + \Gamma_f(\dot{X}) - \tilde{\Gamma}_f(\dot{X}) + \Gamma_{ext}$$

(10)

With the hypothesis of small errors $e = X_d - X$ and time derivative $\dot{e}$, the zero order approximation around $X_d$ of Equation 9 leads to:

$$\Gamma(X, \dot{X}, \ddot{X}) = \tilde{A}(X)\ddot{X} + \tilde{H}(X, \dot{X}) + \tilde{G}(X) + \tilde{\Gamma}_f(\dot{X}) + \tilde{\Gamma}$$

(11)
where \( o(e, \dot{e}) \) are the higher order terms. Notice that the second term can be written as \( \hat{A}(X_d)\ddot{e} \) for a joint space control where \( X \) is not accessible.

By analysing Equation 11, the Inverse Dynamic Model in Equation 9 applied to the reference \( X_d \) can be noticed. Therefore, by using these terms in a feedforward term, a suitable torque control signal is:

\[
\Gamma(X, \dot{X}, \ddot{X}) = \hat{A}(X_d)\ddot{X}_d + \hat{H}(X_d, \dot{X}_d) \hat{G}(X_d) + \hat{\Gamma}_f(X_d) + \hat{A}(X)u_{PID} \tag{12}
\]

where \( u_{PID} \) is still the control signal defined in Equation 6.

Considering Equations 11 and 12, the error signal has the following behaviour:

\[
\ddot{e} + K_d\dot{e} + K_p e + K_i \int \dot{e} dt = \hat{A}(X)^+ (\dot{\Gamma} + o(e, \dot{e})) \tag{13}
\]

where \( \hat{A}^+ \) is the pseudo inverse of \( \hat{A} \).

The gain tuning is the same as Equation 8. With the hypothesis of negligible disturbances \( \Gamma \), the error has an exponential decay determined by the gain tuning.

In this case, there is a double integrator between the controller output \( u_{PID} \) and the controller input \( e \):

\[
u_{PID} = \ddot{e} = \ddot{X} - \ddot{X}_d \tag{14}\]

Therefore, the control scheme allows for a tangent linearisation of the robot dynamics. Thus, a linear PID controller allows for optimal dynamic performances. Nevertheless, this is only achieved under the hypothesis of negligible disturbances \( \Gamma \) and small error signal. The disturbances \( \Gamma \) are modeling errors mainly due to assembly errors and unmodeled frictions. These errors can be reduced with an adapted modeling method [24] and a good dynamic identification [27], [28]. Let us remark that in a joint space control case, these disturbances are increased by the use of a numerical forward transformation expressing the joint errors in the Cartesian space. The most restricting hypothesis concerns the error signal. Indeed, high gains are needed to have small errors. However, the gain tuning is limited by the resonance frequency, rather low for industrial parallel robots. High gains are thus prohibited. To decrease errors with higher efficiency, robust techniques can be employed such as adaptive control [5]. In addition, the feedforward term can include higher order approximations of Equation 9. Nevertheless, these solutions require more complex models with more computation. Instead of using a nonlinear feedforward, the dynamics compensation can be also ensured in the feedback loop as detailed below.

E. Computed Torque Control

The so-called Computed Torque Control is widespread for serial robots [29], [8]. It allows for excellent tracking performances at high speed. Nevertheless, the performances are not as good as expected in the parallel case [30], [31]. Indeed, this control is generally done in the joint space (see Figure 5 where \( \overrightarrow{IKM} \) is the algebraic Inverse Kinematic Model, \( \overrightarrow{FKM} \) is a solution to the forward kinematic problem, \( \overrightarrow{D} \) is a solution to the forward instantaneous kinematic problem). The use of forward numerical transformations results in a lack of accuracy, speed and stability. To improve these issues, we propose a Cartesian space Computed Torque Control (see Figure 6), without any numerical estimation. We show that this control is perfectly relevant for a parallel robot and allows for better performances than the nonlinear feedforward control scheme.

By reporting result of Equation 14 directly in Equation 9, a suitable torque control law is expressed as:

\[
\Gamma(X, \dot{X}, \ddot{X}) = \hat{A}(X) (\dot{X}_d + u_{PID}) + \hat{H}(X, \dot{X}) + \hat{G}(X) + \hat{\Gamma}_f(X) \tag{15}
\]

Considering Equations 9 and 15, the error signal has the following behaviour:

\[
\ddot{e} + K_d\dot{e} + K_pe + K_i \int \dot{e} dt = \hat{A}(X)^+ (\dot{\Gamma}) \tag{16}
\]

where \( \dot{\Gamma} \) is expressed in Equation (10).

The gain tuning is the same as Equation 8. With the hypothesis of negligible disturbances \( \dot{\Gamma} \), the error has an exponential decay determined by the gain tuning.

The interesting aspects of the nonlinear feedforward control, such as an error with an exponential decay and a linearisation of the dynamics, are still present. Let us remark that the linearisation is now exact instead of tangent. Nevertheless, the restrictive assumption of small errors is removed between Equation 13 and Equation 16. The remaining assumption concerns only the negligible modeling errors. With the prerequisite of a good model associated with a good dynamic identification, the Computed Torque Control can ensure the best performances. Of course, this control scheme can be improved in terms of robustness with regards to modeling errors with predictive or \( H_\infty \) methods [30], [31]. In addition, such methods should be easier to set up now than in the former control schemes, since there are fewer perturbations.

F. Summary

In conclusion of this theoretical approach, a discussion of each control scheme use with regards to the application
can be set up. Firstly, when the application requires either low speed with high accuracy or vice versa in a small part of the workspace, a linear control is enough. In addition, this a simple control with an easy set-up. Nevertheless, one should be careful with the path generation and the gain tuning (which can be done locally for better performances). Secondly, when speed and accuracy requirements are not met with linear control, nonlinear control should be used. The nonlinear feedforward control allows for an off-line dynamics compensation. In addition, it can be used in the joint space when the measure of the end-effector pose is not available. Nevertheless, high gains or robust techniques are required to make this control scheme useful. However, high gains are rarely conceivable, especially in high speed machining where robots are quite heavy. Furthermore, the robust techniques are not always relevant in an industrial context since they require great knowledge in automatic control. On the opposite, a Computed Torque Control does not require robust techniques to ensure good tracking with the prerequisite of a performant dynamic model and a good identification. Nevertheless, it requires more computation resources. Fortunately, actual computers allows for sufficient ones.

III. PRESENTATION OF THE ISOGLIDE-4 T3R1

To validate the above discussion, the proposed modeling and control scheme are applied to the Isoglide-4 T3R1. This parallel robot is a fully-isotropic one with decoupled motion (see Figure 7 and [18]). It is a four degrees of freedom robot with three translations and one rotation. It is designed for High Speed Machining. Hence, stiffness requirements impose an important weight: 31kg per leg and 14kg for the end-effector. A desired maximum acceleration of 20$m.s^{-2}$ associated with heavy weight creates an important dynamic coupling between the legs. This test-bed is well suited to the validation of the approach, since its weight prevents us from neglecting the dynamics. Moreover, its closed-form kinematic models remove from the control problem the troubles associated to kinematic nonlinear couplings [32]. They also compensate for the current technological lack of reliable and accurate high-speed sensor of the end-effector pose.
TABLE I  
MEASURE OF THE AVERAGE STRAIGHTNESS ALONG AXIS X, Y AND Z  
AND ORIENTATION ERROR ALONG AXIS θ FOR THE THREE CONTROL  
SCHEMES ON A 100mm SQUARE FOR SEVERAL ACCELERATIONS

<table>
<thead>
<tr>
<th>Speed (m.s⁻²)</th>
<th>Straightness and orientation error (10⁻² degree)</th>
<th>Linear</th>
<th>NL Feedforward</th>
<th>CTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X: 382.2</td>
<td>Y: 458.2</td>
<td>Z: 42.6</td>
<td>θ: 0.25</td>
</tr>
<tr>
<td>3</td>
<td>X: 1646.3</td>
<td>Y: 795.6</td>
<td>Z: 161.1</td>
<td>θ: 0.94</td>
</tr>
<tr>
<td>5</td>
<td>X: 2906.0</td>
<td>Y: 1682.4</td>
<td>Z: 285.7</td>
<td>θ: 1.82</td>
</tr>
<tr>
<td>10</td>
<td>X: 8137.1</td>
<td>Y: 5204.5</td>
<td>Z: 609.2</td>
<td>θ: 4.76</td>
</tr>
<tr>
<td>20</td>
<td>X: 21331.1</td>
<td>Y: 12759.1</td>
<td>Z: 1191.6</td>
<td>θ: 11.28</td>
</tr>
</tbody>
</table>

IV. Results

A. Simulation

A comparison between linear, nonlinear feedforward and Computed Torque controls is proposed in simulation. Each control is tested on a square trajectory in the XY plan. A fifth degree polynomial interpolation is used with several maximal accelerations, ranging from 1m.s⁻² to 20m.s⁻². The dynamic behaviour of the robot is simulated with the forward dynamic model (FDM). Realistic noises are included such as a 50µm error on the geometrical parameters (considering the required manufacturing tolerances for this robot), a 10% error on dynamic parameters (as generally achieved in dynamic identification) and a 1µm accuracy on joint sensors (see Figure 8). For a fair comparison, gains are tuned with $\omega = 5\ Hz$ for each control scheme (see Equation 8).

The results confirm the assertions above. Figure 9 shows a comparison between the three control schemes at 3m.s⁻², which is a typical high-speed machining process acceleration. As expected, the Computed Torque Control ensures the best performances. The linear control can not compensate for the dynamic coupling between leg, even at such a low speed. Concerning the nonlinear feedforward control, the tracking performances are worse than expected. Indeed, the hypothesis of small errors is not verified here. To improve the latter performances, higher gains can be set up ($\omega = 15\ Hz$). However, the stability bounds are reached making this tuning unsuitable. Therefore, a robust controller should be used.

According to Figure 9(b), it can be noticed that no overshoot appears with the Computed Torque Control, contrary to the other controls. This is interesting in terms of path generation. In fact, corners are generally made smooth, with speed decreasing and circular connection, to avoid these overshoots. Consequently, with a Computed Torque Control, it does not seem necessary thus improving process duration.

As the Isoglide-4 T3R1 is designed for High-Speed Machining operations, a straightness measure on axis X,Y,Z and an orientation error measure are judicious. These measures are summarized in Table I for the three control schemes along the same trajectory. It can be noticed that the Computed Torque Control allows for reaching required tolerances in the XYZ plan (100µm) with machining typical accelerations (above 5m.s⁻²) contrary to the other control schemes.

B. Experiments

We propose now an experimental verification of the simulations realised above. Nevertheless, we aim at obtaining a control without robust techniques, which is industrially more interesting. According to the poor performances of the nonlinear feedforward control without robust controller, the
latter is thus not implemented. All the more as it requires as much implementation effort as the Computed Torque Control. Therefore, only Computed Torque and linear control are compared. The end-effector trajectory is measured with a 512 \times 512 camera running at 250Hz. This exteroceptive measure allows for focusing on the real motion of the end-effector contrary to an estimation with a forward kinematic model which is biased by structure defects and deformations. In order to prove the camera accuracy, a comparison between the latter and a laser interferometer is proposed (see Figure 10). The Figure 11 gives the measures from the interferometer and the camera on the same deviation. The camera is close to the interferometer with a 26\( \mu m \) average error between the two sensors. It validates the further results.

Figure 12 shows a comparison between linear and Computed Torque controls. The reference trajectory is again a simple 100 mm square in the XY frame. A fifth degree path generation with a 3\( m.s^{-2} \) maximal acceleration is used. The trajectory is executed segment by segment. According to Figure 12, it can be noticed that great improvements are realized. Indeed, due to heavy inertia, the dynamic coupling between legs is not negligible even at 3\( m.s^{-2} \). Using Computed Torque Control improves tracking since straightness errors are approximately divided by 7 for X-axis displacements and 10 for Y-axis displacements (see Table II where segment 1 is the left edge of square, segment 2 the right edge, segment 3 the bottom edge and segment 4 the top edge).

The simulation and experiments show similar behaviour. Contrary to the simulation, the experimental results take into account the structure defects and deformations. The performances are thus lower in experiments than in simulation.

### Table II

<table>
<thead>
<tr>
<th></th>
<th>Linear</th>
<th>CTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment 1</td>
<td>733</td>
<td>154</td>
</tr>
<tr>
<td>Segment 2</td>
<td>2255</td>
<td>330</td>
</tr>
<tr>
<td>Segment 3</td>
<td>3318</td>
<td>443</td>
</tr>
<tr>
<td>Segment 4</td>
<td>3143</td>
<td>293</td>
</tr>
</tbody>
</table>

*Measured straightness in \( \mu m \) on square segment with an high speed camera*
V. CONCLUSION AND PERSPECTIVES

Parallel robots have a different behaviour from serial ones. Therefore, the control schemes have to be adapted and not directly reused. In this paper, we revisited the three major dynamic control strategies of a parallel robot. First of all, we showed that the Cartesian space control is perfectly suitable for parallel robots. Next, each control scheme was transposed in this control space. Then, the three control schemes are compared through a theoretical analysis of control equations. Simulation and experiments on a particular test-bed were provided to prove our assertions. As a conclusion, a Cartesian space Computed Torque Control offers the best performances even at machining speed, although generally considered as quasi-static and not dynamic. From a theoretical point of view, this result can be generalised to every parallel robot with a closed-form Inverse Kinematic Model. Nevertheless, it assumes to have a competitive measure of the end-effector pose.

REFERENCES


