Revisiting the Major Dynamic Control Strategies of Parallel Robots

Flavien Paccot[1]   Nicolas Andreff[1,2]   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 $170m.s^2$ in pick-and-place operations [2] and $100\mu m$ 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

[1]LASMEA - UMR 6602 CNRS, Blaise Pascal University, Clermont-Ferrand, France

[2]LAMI, IFMA, Aubiere, France

[3]ISRC, Sungkyunkwan University, Suwon, South Korea

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
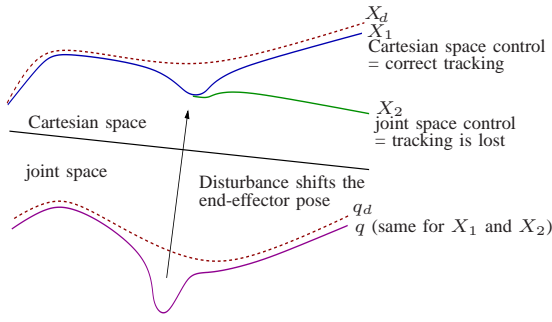
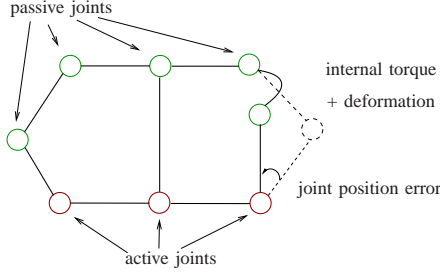Fig. 1. *Comparison between Cartesian and joint space control*

Fig. 2. *Internal torques created by joint errors*

one [1]. Thus, the numerical estimations limit the control performances in term of stability, accuracy and speed.

Secondly, a parallel robot can have several end-effector poses for one joint configuration [11]. Therefore, a disturbance on the end-effector trajectory can shift its pose without changing the joint configuration, especially in the neighborhood of singularities or naturally in the cuspidal robots case [12]. By servoing the joint positions, the return to the end-effector pose reference is not achieved while the joint reference tracking is still ensured. On the opposite, a 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.

Current classical measurement means, such as a laser tracker, are quite expensive and integration in a control scheme is not straightforward [14]. On the opposite, com-

puter vision seems to be perfectly relevant [15], [16]. 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:

$$\Gamma(q, \dot{q}, \ddot{q}) = A(q)\ddot{q} + H(q, \dot{q}) + G(q) + \Gamma_f(\dot{q}) + \Gamma_{ext} \quad (1)$$

where q are the active joint variables, Γ 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, Γ_f is the friction influence and Γ_{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 robots depends on the end-effector pose and time derivatives [24], [25]. It is hence more relevant to express the model above as [25]:

$$\Gamma(X, \dot{X}, \ddot{X}) = A(X)\ddot{X} + H(X, \dot{X}) + G(X) + \Gamma_f(\dot{X}) + \Gamma_{ext} \quad (2)$$

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
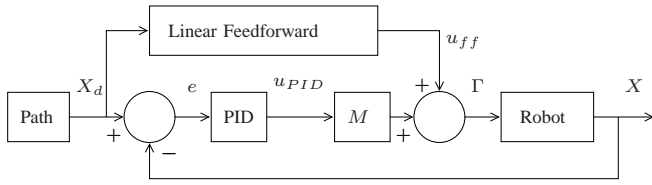
Fig. 3. *Linear control in the Cartesian space with PID controller and feedforward*

Fig. 4. *Nonlinear feedforward in the Cartesian space*

performances deteriorate quickly when the trajectory speed increases.

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

$$\Gamma(X, \dot{X}, \ddot{X}) = M\ddot{X} + F_v\dot{X} + F_s + \tilde{\Gamma}_{PID} \qquad (3)$$

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 $\tilde{\Gamma}_{PID}$ is the resulting error between Equation 2 and Equation 3. It can be expressed as:

$$\begin{aligned} \tilde{\Gamma}_{PID} &= (A(X) - M)\ddot{X} + H(X, \dot{X}) + G(X) \\ &\quad + \Gamma_f(\dot{X}) - F_v\dot{X} - F_s + \Gamma_{ext} \end{aligned} \qquad (4)$$

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

$$\Gamma(X_d, \dot{X}_d, \ddot{X}_d) = M\ddot{X}_d + F_v\dot{X} + F_s + Mu_{PID} \qquad (5)$$

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

$$u_{pid} = K_d\dot{e} + K_pe + K_i \int edt \qquad (6)$$

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:

$$\ddot{e} + K_d\dot{e} + K_pe + K_i \int edt = M^+\tilde{\Gamma}_{PID} \qquad (7)$$

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

$$\begin{aligned} K_p &= 3\omega^2 \\ K_d &= 3\omega \\ K_i &= \omega^3 \end{aligned} \qquad (8)$$

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 $\tilde{\Gamma}_{PID}$, 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 [1], [24], [25]. Therefore, the first hypothesis is achieved only
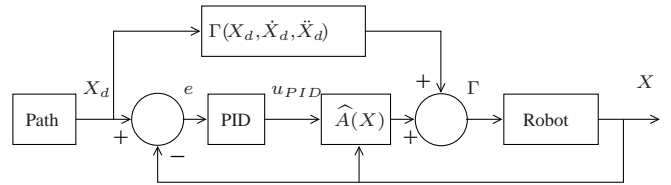
very locally and at low speed. In addition, the disturbances $\tilde{\Gamma}_{PID}$ are mainly generated by the dynamic coupling (see Equation 4) where the first two terms concern the dynamic coupling between legs). In this case, the second hypothesis is also only verified at low speed. In conclusion, 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:

$$\Gamma(X, \dot{X}, \ddot{X}) = \widehat{A}(X)\ddot{X} + \widehat{H}(X, \dot{X}) + \widehat{G}(X) + \widehat{\Gamma}_f(\dot{X}) + \tilde{\Gamma} \qquad (9)$$

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

$$\begin{aligned} \tilde{\Gamma} &= (A(X) - \widehat{A}(X))\ddot{X} + H(X, \dot{X}) - \widehat{H}(X, \dot{X}) \\ &\quad + G(X) - \widehat{G}(X) + \Gamma_f(\dot{X}) - \widehat{\Gamma}_f(\dot{X}) + \Gamma_{ext} \end{aligned} \qquad (10)$$

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:

$$\begin{aligned} \Gamma(X, \dot{X}, \ddot{X}) &= \widehat{A}(X)\ddot{X} + \widehat{H}(X, \dot{X}) + \widehat{G}(X) + \widehat{\Gamma}_f(\dot{X}) + \tilde{\Gamma} \\ &= \widehat{A}(X_d - e)\ddot{X}_d - \widehat{A}(X)\ddot{e} \\ &\quad + \widehat{H}(X_d - e, \dot{X}_d - \dot{e}) + \widehat{G}(X_d - e) \\ &\quad + \widehat{\Gamma}_f(\dot{X}_d - \dot{e}) + \tilde{\Gamma} \\ &= \widehat{A}(X_d)\ddot{X}_d - \widehat{A}(X)\ddot{e} + \widehat{H}(X_d, \dot{X}_d) \\ &\quad \widehat{G}(X_d) + \widehat{\Gamma}_f(\dot{X}_d) + \tilde{\Gamma} + o(e, \dot{e}) \end{aligned} \qquad (11)$$

where $o(e, \dot{e})$ are the higher order terms. Notice that the second term can be written as $\widehat{A}(X_d)\ddot{e}$ 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:

$$
\begin{aligned}
\Gamma(X, \dot{X}, \ddot{X}) =\ & \widehat{A}(X_d)\ddot{X}_d + \widehat{H}(X_d, \dot{X}_d) \\
& \widehat{G}(X_d) + \widehat{\Gamma}_f(\dot{X}_d) + \widehat{A}(X)u_{PID}
\end{aligned}
\tag{12}
$$

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

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

$$
\ddot{e} + K_d\dot{e} + K_p e + K_i \int e\,dt = \widehat{A}(X)^+ \left(\tilde{\Gamma} + o(e, \dot{e}) \right)
\tag{13}
$$

where $\widehat{A}^+$ is the pseudo inverse of $\widehat{A}$.

The gain tuning is the same as Equation 8. With the hypothesis of negligible disturbances $\tilde{\Gamma}$, the error has a 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 $\tilde{\Gamma}$ and small error signal. The disturbances $\tilde{\Gamma}$ 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 $\widehat{IKM}$ is the algebraic Inverse Kinematic Model, $F\check{K}M$ is a solution to the forward kinematic problem, $\check{D}$ is a solution to the forward instantaneous kinematic
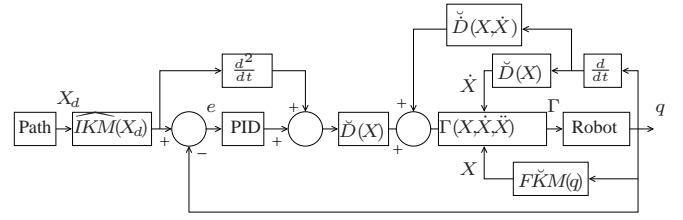

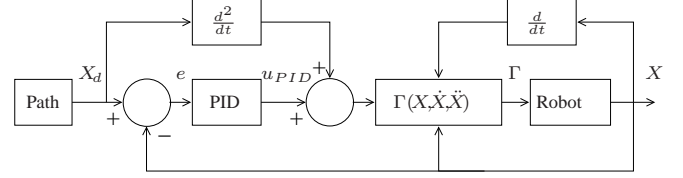Fig. 5. *Computed Torque Control in the joint space*

Fig. 6. *Computed Torque Control in the Cartesian space*

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:

$$
\Gamma(X, \dot{X}, \ddot{X}) = \widehat{A}(X)\left(\ddot{X}_d + u_{PID}\right) + \widehat{H}(X, \dot{X}) + G(X) + \widehat{\Gamma}_f(\dot{X})
\tag{15}
$$

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

$$
\ddot{e} + K_d\dot{e} + K_p e + K_i \int e\,dt = \widehat{A}(X)^+ \left(\tilde{\Gamma} \right)
\tag{16}
$$

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

The gain tuning is the same as Equation 8. With the hypothesis of negligible disturbances $\tilde{\Gamma}$, the error has a 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_∞ 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
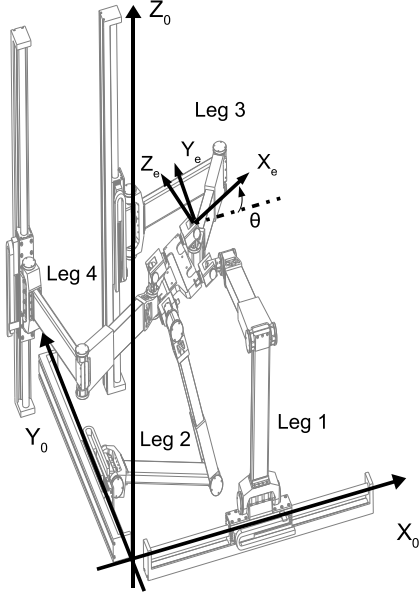
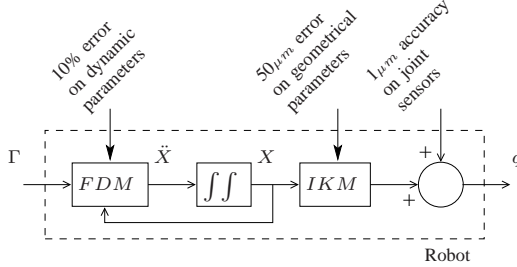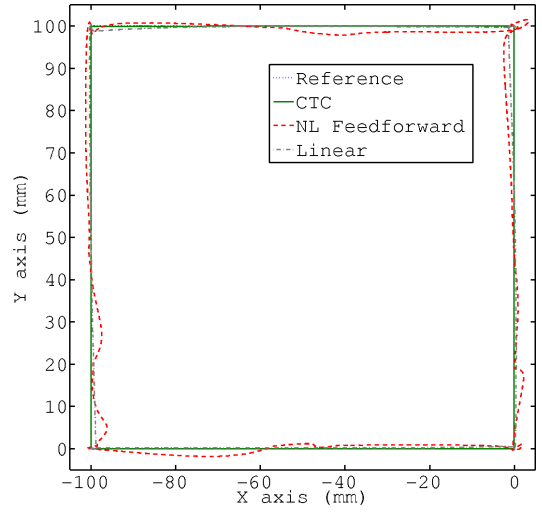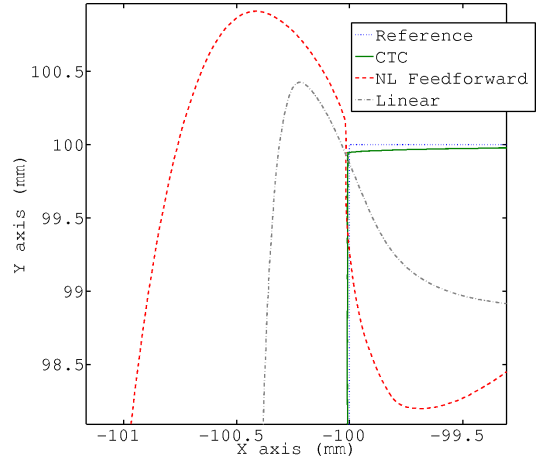Fig. 7. *Global view of the Isoglide-4 T3R1*

Fig. 8. *Dynamic model of the robot used in simulation*

(a) Global view

(b) Zoom

Fig. 9. *Comparison of the three control schemes on a 100mm square at $3m.s^{-2}$*

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 $20m.s^{-2}$ 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.

Speed $(m.s^{-2})$	Straightness (μm) and orientation error $(10^{-2}$ degree)	Linear	NL Feedforward	CTC
1	X	354.2	4034.0	11.2
	Y	458.2	4688.6	10.9
	Z	42.6	69853.1	10.2
	θ	0.25	2044.9	0.28
3	X	1646.3	5522.9	50.8
	Y	795.6	3036.0	37.5
	Z	161.1	45476.3	14.9
	θ	0.94	1641.9	0.49
5	X	2906.0	4153.8	109.8
	Y	1682.4	2831.1	81.7
	Z	285.7	47501.0	19.3
	θ	1.82	1833.0	0.72
10	X	8131.1	12954.7	384.1
	Y	5204.5	8738.3	203.6
	Z	609.2	23284.1	24.7
	θ	4.76	867.1	2.22
20	X	21331.1	25973.6	336.1
	Y	12759.1	15662.6	175.3
	Z	1191.6	30193.1	121.1
	θ	11.28	99.83	3.70

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

(a) Calibration pattern and interferometer mounted on the end-effector

(b) Camera and laser

Fig. 10. *Straightness measure with an high speed camera and a laser interferometer*
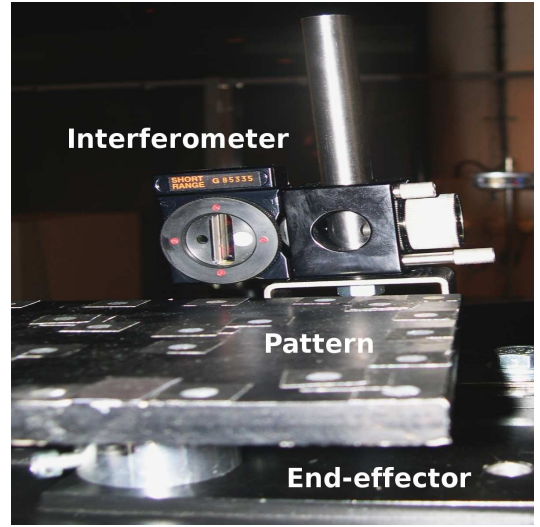
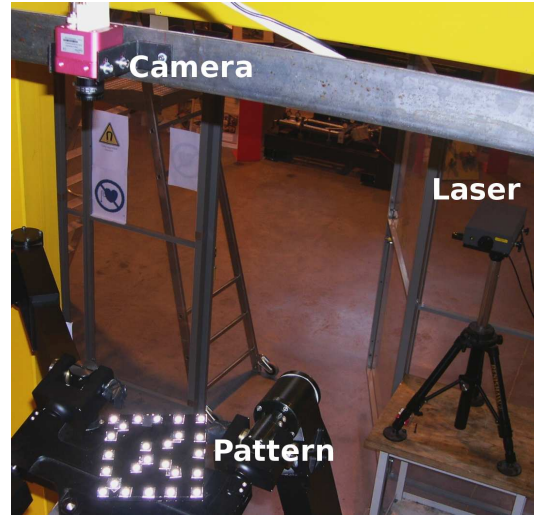## IV. RESULTS

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 $50\mu m$ 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 $1\mu m$ accuracy on joint sensors (see Figure 8). For a fair comparison, gains are tuned with $\omega = 5Hz$ 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 ($\omega = 15Hz$). 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 ($100\mu m$) 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
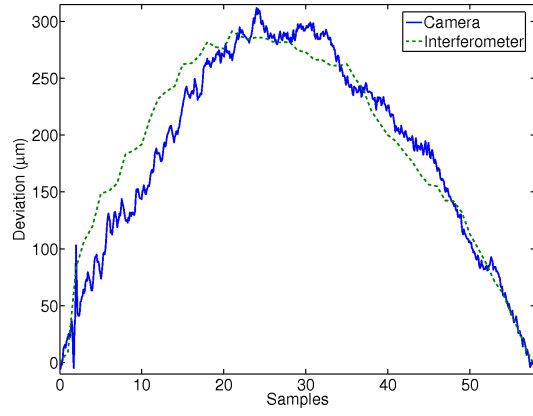
(a) Deviation on Y-axis
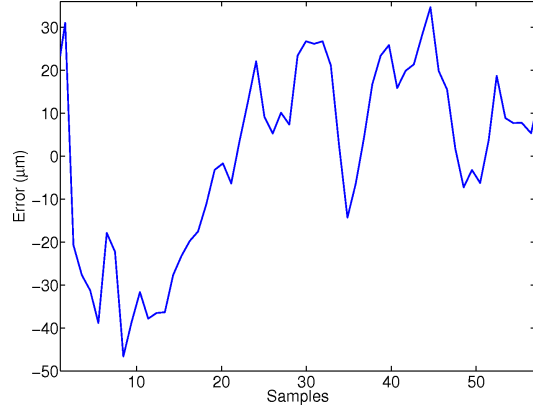
(b) Error between laser interferometer and camera measures

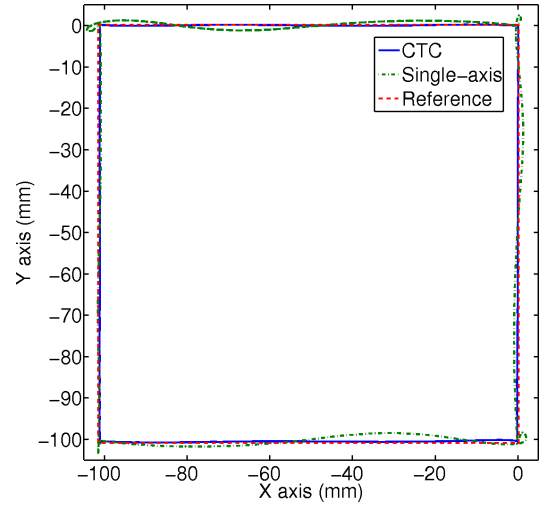Fig. 11.   *Comparison between laser interferometer and camera*

	Linear	CTC
Segment 1	733	154
Segment 2	2255	330
Segment 3	3318	443
Segment 4	3143	293

TABLE II

Measured straightness in μm on square segment with an high speed camera

(a) Global view

(b) Zoom

Fig. 12.   *Comparison between single-axis and CTC controller measured with an high speed camera on a 100mm XY square*

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 512×512 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 $3m.s^{-2}$ 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 $3m.s^{-2}$. 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.

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

[1] J.P. Merlet. *Parallel Robots*. Kluwer Academic Publishers, 2000.
[2] V. Nabat, M. O Rodriguez, O. Company, S. Krut, and F. Pierrot. PAR4: very high speed parallel robot for pick-and-place. In *IEEE International Conference on Intelligent Robots and Systems (IROS'05)*, pages 1202–1207, Edmonto, Canada, 2005.
[3] J.P. Merlet. Still a long way to go on the road for parallel mechanisms. In *ASME 27th Biennial Mechanisms and Robotics Conference*, 2002.
[4] P.R. Ouyang, W.J. Zhang, and F.X. Xu. Non linear PD control for trajectory tracking with consideration of the design for control methodology. In *IEEE International Conference on Robotics and Automation (ICRA'02)*, pages 4126–4131, Washington, USA, May 2002.
[5] M. Honegger, R. Brega, and G. Schweitzer. Application of a non-linear adaptative controller to a 6 dof parallel manipulator. In *IEEE International Conference on Robotics and Automation (ICRA'00)*, pages 1930–1935, San Francisco, USA, April 2000.
[6] H. Abdellatif and B. Heimann. Adapted time-optimal trajectory planning for parallel manipulators with full dynamic modelling. In *IEEE International Conference on Robotics and Automation (ICRA'05)*, pages 411–416, Barcelone, Spain, April 2005.
[7] Y. Zhiyong and T. Huang. A new method for tuning PID parameters of a 3-DOF reconfigurable parallel kinematic machine. In *IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 2249–2254, New Orleans, USA, April 2004.
[8] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. London, 2002.
[9] K.J. Waldron and K.H. Hunt. Series-parallel Dualities in Actively Coordinated Mechanisms. *The International Journal of Robotics Research*, 10(2):pp. 473–480, 1991.
[10] T. Dallej, N. Andreff, Y. Mezouar, and P. Martinet. 3d pose visual servoing relieves parallel robot control from joint sensing. In *International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, October 2006.
[11] M.L. Husty. An algorithm for solving the direct kinematic of the Gough-Stewart platforms. Research report tr-cims-94-7, McGill University, Montréal, Canada, 1994.
[12] D. Chablat and P. Wenger. Working modes and aspects in fully parallel manipulators. In *IEEE International Conference on Robotics and Automation (ICRA'98)*, pages 1964–1969, Leuven, Belgium, May 1998.
[13] P. Dauchez, X. Delebarre, and R. Jourdan. Hybrid control of a two-arm robot handling firmly a single rigid object. In *2nd International Workshop on Sensorial Integration for Industrial Robots (SIFIR'89)*, pages 67–62, Zaragosa, Spain, November 1989.
[14] M. Vincze, J.P. Prenninger, and H. Gander. A laser tracking system to measure position and orientation of robot end-effectors under motion. *The International Journal of Robotic Research*, 13(4):305–314, 1994.
[15] J. Gangloff. *Asservissements visuels rapides d'un robot manipulateur à six degrés de libertés, Suivi de cible et de profilé*. Phd, Université Louis Pasteur, Strasbourg, 1999.
[16] O. Ait Aider, N. Andreff, J.M. Lavest, and P. Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *European Conference on Computer Vision (ECCV)*, pages 56–68, Graz, Austria, May 2006.
[17] S. Krut, O. Company, M. Benoit, H. Ota, and F. Pierrot. I4: a new parallel mechanism for scara motions. In *IEEE International Conference on Robotics and Automation (ICRA'03)*, pages 1875–1881, Taipei, Taiwan 2003.
[18] G. Gogu. Fully-isotropic T3R1-type parallel manipulators. *On Advances In Robot Kinematics*, pages 265–272, 2004.
[19] J.P. Merlet. Solving the froward kinematic of a Gough-type parallel manipulator with interval analysis. *The International Journal of Robotic Research*, 23(3):221–235, 2004.
[20] F. Marquet, O. Company, S. Krut, and F. Pierrot. Enhancing parallel robot with redundant sensors. In *International Conference on Robotics and Automation (ICRA'02)*, pages 4114–4119, Washington, USA, May 2002.
[21] L. Baron and J. Angeles. The direct kinematics of parallel manipulators under joint-sensor redundancy. *IEEE Transactions on Robotics and Automation*, Vol. 16(1):pp. 1–8, 2000.
[22] J. Wang and O. Masory. On the accuracy of a Stewart platform - part I: The effect of manufacturing tolerances. In *IEEE International Conference on Robotics and Automation (ICRA'93)*, pages 114–120, Atlanta, USA, 1993.
[23] D. Daney. Self calibration of Gough platform using leg mobility constraints. In *World congress on the theory of machine and mechanisms*, pages 104–109, Oulu, Finland, 1999.
[24] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. In *IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 3665–3670, New Orleans, USA, April 2004.
[25] B. Dasgupta and P. Choudhury. A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 34:801–824, 1999.
[26] B. Denkena and C. Holz. Advanced position and force control concepts for the linear direct driven hexapod PaLiDA. In *The fifth Chemnitz Parallel Kinematics Seminar*, pages 359–378, Chemnitz, Germany, 2006.
[27] M. Gautier and P. Poignet. Extended Kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Pratice*, 9(12):1361–1372, 2001.
[28] M.M. Olsen and H.G. Peterson. A new method for estimating parameters of a dynamic robot model. *IEEE Transactions on Robotics and Automation*, 17(1):95–100, 2001.
[29] J.Y.S Luh, M.W. Walker, and R.C.P Paul. Resolved acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, AC-25(3):468–474, 1980.
[30] A. Vivas, P. Poignet, and F. Pierrot. Predictive functionnal control of a parallel robot. In *IEEE International Conference on Intelligent Robots and Systems (IROS'03)*, pages 2785–2790, Las Vegas, USA, October 2003.
[31] S.H. Lee, J.B. Song, W.C. Choi, and D. Hong. Control design for a Stewart platform using small workspace characteristics. In *IEEE International Conference on Intelligent Robots and Systems (IROS'01)*, pages 2184–2189, Hawaii, USA, October 2001.
[32] R. Rizk, N. Andreff, J.C Fauroux, J.M. Lavest, and G. Gogu. Precision study of a decoupled four degrees of freedom parallel robot including manufacturing and assembling errors. In *International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME'06)*, Grenoble, France, May 2006.