Vision-based Computed Torque Control for Parallel Robots

Flavien Paccot[1] Nicolas Andreff[1,2] Philippe Martinet[1] Wisama Khalil[3]

[1]LASMEA - UMR CNRS 6602
24, Avenue des Landais 63177 Aubière Cedex, France

[2]LAMI
BP 265, 63175 Aubière Cedex, France

[3]IRCCYN - UMR CNRS 6597
Ecole Centrale de Nantes
BP 92 101, 44312 Nantes Cedex, France

Abstract— **This paper presents a novel approach to dynamic control of parallel robots based on the end-effector pose and velocity visual measurement. We show that Computed Torque Control in joint space is unsuitable for parallel robots and that Computed Torque Control in Cartesian space requires the end-effector pose and velocity estimation. Relying on a recent work showing ability of camera to measure both the pose and the velocity of a mobile object, we propose an original vision-based Computed Torque Control scheme. Improvements are discussed and simulation results show the expected performances.**

I. Introduction

Defined as an end-effector linked to a fixed base by at least two kinematics chains (also called legs), parallel robots offer more stiffness than serial robots, and higher speed as several actuators are used for one move [19]. Firstly used in tire testing [10] and flight simulation [22], parallel robots are also relevant for 'pick-and-place' (ABB Flexpicker with maximal acceleration of $100m.s^{-2}$ for a $2kg$ load, Figure 1) and High Speed Machining (Urane SX, $50m.s^{-2}$, $100m.min^{-1}$, Figure 2), where the expected stiffness and precision are missmatched [23]. Dynamic control of parallel robots is proposed to improve accuracy at high speed.

Fig. 1. *ABB Flexpicker*

Fig. 2. *Comau Automation Urane SX*

Merlet shows that parallel robots advantages (accuracy, stiffness and high-speed) are potential and can only be archieved if structures and control are well designed [20]. Indeed, dynamic phenomena, which occur with high acceleration (inertia, centrifugal and Coriolis forces), can not be fully compensated for with a simple linear controller, especially for parallel robots whose coupling between legs makes dynamic behaviour strongly non-linear. It results in a lack of accuracy. Computed Torque Control gives a solution by taking into account dynamic phenomena in the torque computation, via the Inverse Dynamic Model [12].

However, an important issue in parallel robots model-based control exists. Actually, there is a duality between parallel and serial robots (Figure 3). Whereas serial manipulators have a closed-form Forward Kinematic Model (FKM) depending on joint variables and numerical Inverse Kinematic Models (IKM), parallel robots have only algebraic Inverse Kinematic Models, function of the end-effector pose and numerical Forward Kinematic Model. Hence in a parallel robot model-based control such as Computed Torque Control, there is an ambiguity since the end-effector pose is needed and can not be obtained directly from only available joint variable measurement. Indeed, the end-effector pose is estimated via the Forward Kinematic Model which is often based on non-linear optimisation with many possible solutions [18], [11]. It decreases control speed and accuracy. A proposed solution consists in designing special mechanical structures to have simple Forward Kinematic Model, with closed-form expression such as Isoglide-4 T3R1 [9]. Other methods rely on metrological redundancy, which allows to make the end-effector pose computation easier and reduces the numbers of possible solutions of the Forward Kinematics problem, by adding proprio or exteroceptives measurements [3].

Nevertheless, these solutions are based on an estimation of the end-effector pose, limited by modeling and computation errors. Rather than estimating the end-effector pose, a measurement of it is preferable. In this way, computer vision allows an easy measure of the end-effector pose [6], [16]. Largely used for the control of serial robots [24], [7], visual servoing seems perfectly relevant for parallel robots [14]. However, in most cases, visual servoing consists of a kinematic control, thus it reduces dynamic performances. That is why Gangloff introduces a dynamic control of a serial robot using high-speed vision [8]. However, since robot dynamics are compensated for by an inner loop, this method is not relevant for parallel robot. Indeed dynamics compensation requires an estimation of the end-effector pose, which is not easy to obtain from joint measurement. Including an end-effector pose visual measure in a classical Computed Torque Control scheme seems to be more interesting in the parallel robot case, but the requirement for the end-effector velocity imposes numerical derivation, with the associated stability and noise problems. A recent work makes the end-effector velocity measurement from a visual information possible [2]. Therefore, a Computed Torque Control based on visual measurement of the end-effector pose and velocity is conceivable. Great improvement in accuracy, computation time and simplification of control schemes are expected.

Last but not least, setting a Computed Torque Control implies to be able to determine a model which describes the robot dynamic behaviour. Numerous methods are proposed mainly based on Lagrange multipliers or virtual works [1], [15]. However, heavy computation needs make these methods unsuitable for control and implies pre-processing simplifications. Alternately, dynamic models can be deduced from Newton-Euler equations. Luh and al. propose an algorithm composed of two recurrences [17], [21] , making Newton-Euler method adapted for control. However, the parallel robots case requires an adaptation for this method, as proposed by Khalil [13].

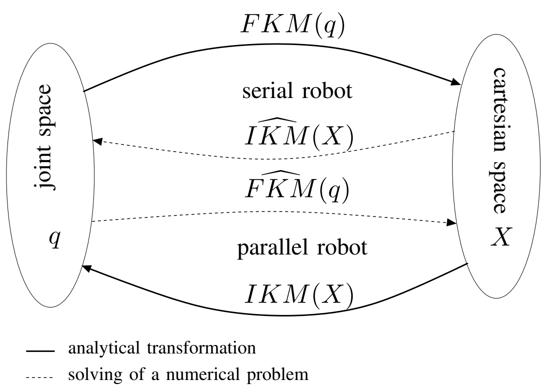

Fig. 3. *The kinematic duality between serial and parallel robot*

The main contribution of this work consists in a proposal of an original Computed Torque Control schemes using computer vision for the end-effector pose and velocity measurement. After a discussion on classical Computed Torque Control schemes, we show that a control in the Cartesian space is the best solution for a parallel robot. Then, after pointing out that vision is a relevant measurement mean, we propose a vision-based Computed Torque Control scheme. Advantages and expected improvements of this scheme over the previous ones are discussed and simulation results are provided. Finally, an application of recent dynamic modeling method to the Isoglide-4 T3R1 [9] is presented. Advantages of both structure and method are pointed out.

Section II deals with Computed Torque Control, whose variant techniques are presented and discussed, and a visual servoing solution is exposed. Section III is devoted to dynamic modeling. Section IV provides simulation results. Finally, a conclusion comes back on the contribution and research perspectives.

II. Computed Torque Control

A. Joint space control

Computed Torque Control schemes have been initially developped for serial robots to improve tracking performances at high speed [12]. The Inverse Dynamic Model (IDM) of a serial robot is a function of the joint variables:

$$IDM(q,\dot{q},\ddot{q}) = A(q)\ddot{q} + H(q,\dot{q}) \tag{1}$$

In this way, Computed Torque Control is generally designed in joint space (see Figure 4). Including the Inverse Dynamic Model allows a non-linear decoupling. Indeed, when dynamic modeling errors can be considered as negligible, replacing $\ddot{q}$ by an appropriate control input ω in the Lagrange formulation (Equation (1)) imposes a linear double integrator transfer function ($1/s^2$) between the control input and the joint variables.

If ω is generated by the addition of a simple PID controller output and the acceleration reference, the tracking error behaviour is designed by the controller gains [12]:

$$\ddot{e} + K_d\dot{e} + K_p e + K_i \int e\,dt = 0 \tag{2}$$

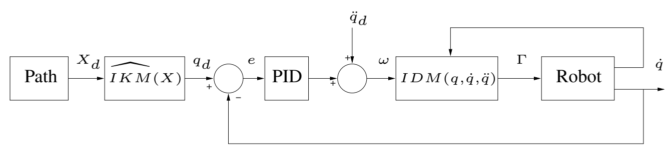

Fig. 4. *Computed Torque Control in the joint space for serial robots*

In the serial robots case, the Computed Torque Control in the joint space requires little computation when path generation, integrating a numerical Inverse Kinematic Model, is done off-line. Thus, it seems highly satisfactory with regard to its wide spreading in the industry. Therefore, this control scheme was transposed to parallel robots (Figure 5), nevertheless it seems to be unsuitable in this case. Indeed, since the Inverse Dynamic Model is expressed in the Cartesian space ($IDM(X,\dot{X},\ddot{X})$), transformations between joint and Cartesian spaces are needed:

$$X = \widehat{FKM}(q) \tag{3}$$

$$\dot{X} = \frac{dX}{dt} \text{ or } D_{inv}^{-1}(X)\dot{q} \tag{4}$$

$$\ddot{X} = D_{inv}^{-1}(X)\ddot{q} + \dot{D}_{inv}^{-1}(X,\dot{X})\dot{q} \tag{5}$$

where D_{inv} is the Inverse Instantaneous Kinematics matrix of the parallel robot.

However, these transformations, including the Forward Kinematics problem solving, should be computed in real-time. In many cases this is implicitely done in dynamic modeling. It may result in a decrease of control performance and lack of accuracy. In addition, refering to Figure 5, the control scheme is very complicated, contrary to serial robots case. Furthermore, since reference tracking errors are different for

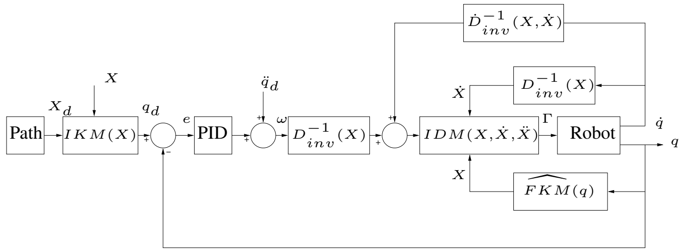

Fig. 5. *Computed Torque Control in the joint space for parallel robot*

each joint, parasite motions are created which are uncompatible with kinematics of the robot, thus generating internal torques. It leads to mechanical structure damage. This default was met for two-arm robots [5] but is generally forgotten for parallel robots. Another drawback is due to the fact that one joint configuration can lead to several end-effector poses [11]. Hence a perturbation can shift the end-effector pose and control will not care with it (Figure 6). These issues make Computed Torque Control in the joint space unsuitable for parallel robots.

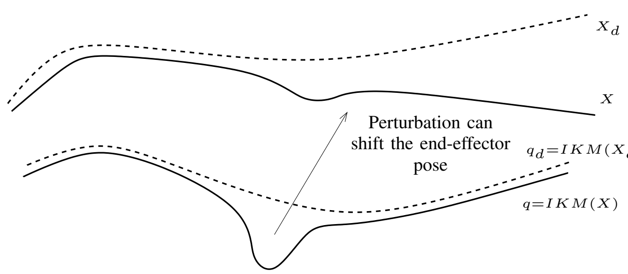

Fig. 6. *Joint space control can not care with effector pose shifting and does not ensure correct tracking in the task space*

B. Cartesian space control

A natural solution to previous problems is a Cartesian control where the regulated error is the difference between the Cartesian reference and the end-effector pose. This type of control exists for serial robots [12] (See Figure 7), where the control input ω replaces $\ddot{X}$ to ensure the pose error behaviour of equation (2). However more on-line computation are needed as a numerical transformation between cartesian and joint spaces is used:

$$\ddot{q} = D^{-1}(q)(\ddot{X} - \dot{D}^{-1}(q,\dot{q})\dot{q}) \tag{6}$$

where D is the Forward Instaneous Kinematics matrix of the serial robot.

Consequently, the control scheme is more complicated (see difference between Figures 4 and 7). Hence, a control in the joint space with a correct path planning is usually prefered. In parallel robots case, Computed Torque Control in the Cartesian space is more relevant because the robot dynamics depend on its end-effector pose:

$$IDM(X,\dot{X},\ddot{X}) = A(X)\ddot{X} + H(X,\dot{X}) \tag{7}$$

Thus, computation in equation (5) is useless, improving control performances and making control scheme simpler (see Figure 8). Then, internal torques are minimised since required torques are computed with regard to end-effector pose error, thus preventing from parasite motions. In addition, no pose shift can be expected when perturbations occur (Figure 9). However, the presence of the Forward Kinematics Model and Forward Instantaneous Kinematics still limit performances and accuracy.

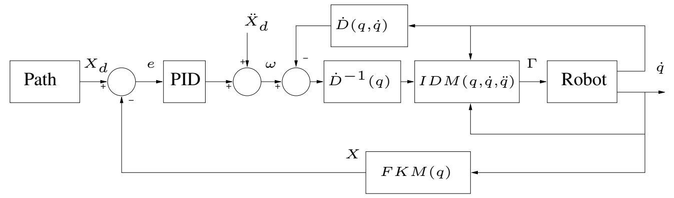

Fig. 7. *Computed Torque Control for serial robot in cartesian space*

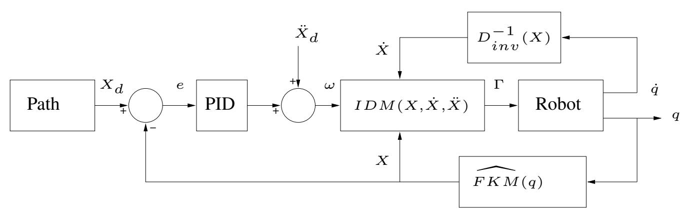

Fig. 8. *Computed Torque Control for parallel robot in cartesian space*

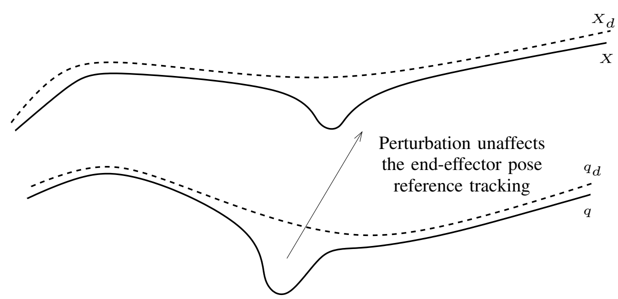

Fig. 9. *Cartesian space control ensures correct tracking in the task space*

C. Visual-based computed torque control

A solution to the above issues lies in measuring the end-effector pose and velocity with exteroceptive sensors. This operation is generally not easy and sensors able to do it are very expensive and stringent to use, such as a laser-tracker. By proposing a visual measure of the end-effector pose and velocity [2], Ait-Aider lets us imagine an original solution integrating this quite simple and cheap sensing method. Figure 10 shows a new approach for Computed Torque Control where Forward Kinematic Model and Forward Instaneous Kinematics are replaced by visual exteroceptive measurements. The control scheme is strongly simplified, and is identical to the control scheme in Figure 4. The kinematic duality between serial and parallel robots, also present in dynamics, can be realised in Computed Torque Control. There is only a difference since path generation is easier in the parallel robots case and can be done on-line.

Since control is still in the Cartesian space, advantages given above are maintened. In addition, visual measurement gives an unique pose with accuracy and reliability, contrary to Forward Kinematic Model, improvement in control accuracy is expected. Furthermore, since the end-effector pose is measured, geometrical errors and structure deformations can be compensated for. Hence, proposed control scheme seems to be perfectly relevant for parallel robot. Ideally, if a high sampling rate can be achieved ($1kHz$ is common for Computed Torque Control to ensure high performances [12]), joint sensors would be useless.

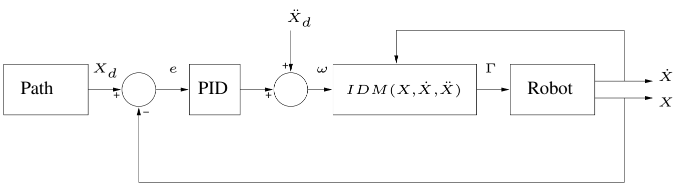

Fig. 10. *Proposed vision-based cartesian Computed Torque Control for parallel robot where X and $\dot{X}$ are obtained by vision*

III. Dynamic Modeling

Before setting up a Computed Torque Control, dynamic modeling has to be achieved. Generally, used methods are based on Lagrange multipliers, requiring heavy computation. It is more relevant to use a Newton-Euler based method in a control context. Hence, a general method for parallel robots dynamic modeling is presented.

A. Method outline

According to Newton-Euler equations, a robot dynamic model can be obtained by separating each body and expressing their velocities, accelerations and generated torques. An algorithm summarizes the method in two recurrences [17], [21], making the method adapted for control. A recent work proposes an adaptation for parallel robot taking into account the closed structure [13]. By dissociating legs and end-effector, the dynamic contribution in the active joint space of each parts can be expressed easily. On the one hand, Newton-Euler algorithm is used to compute the Inverse Dynamic Model of each leg, seen as a serial robot (often simple). Resulting torques in the link between a leg and the end-effector is expressed using the Inverse Instantaneous Kinematics matrix J_i of each leg. Then, this force is calculated in the end-effector frame with a Jacobian matrix linking terminal Cartesian variables of a leg to Cartesian variables of the end-effector. Finally, the robot Forward Instantaneous Kinematics matrix D_{inv}^{-1} projects the contribution of a leg into active joint space. On the other hand, the end-effector dynamics are obtained with Newton-Euler equations and are also expressed in the active joint space via D_{inv}^{-1}. To summarize, the Inverse Dynamic Model of a parallel robot can be expressed as:

$$\Gamma = \underbrace{D_{inv}^{-T} F_p}_{\substack{\text{Contribution of the}\\ \text{end-effector}}} + \sum_{i=1}^{n} \underbrace{D_{inv}^{-T} J_{pi}^{T} J_i^{T} H_i}_{\substack{\text{Contribution of}\\ \text{each leg}}} \tag{8}$$

where:

- D_{inv} is the Inverse Instantaneous Kinematics matrix of the robot
- F_P represents the dynamics of the end-effector
- n is the number of legs
- J_{pi} is the Jacobian matrix linking the Cartesian coordinates of the end-effector to the Cartesian coordinates of the terminal point of leg i
- J_i is the Inverse Instantaneous Kinematics matrix of the leg i
- H_i is the Inverse Dynamic Model of the leg i, considered as a stand alone subsystem, computed with Newton-Euler algorithms

B. Dynamic modeling of the Isoglide-4 T3R1

1) Presentation: The Isoglide-4 T3R1 is a fully-isotropic 4 DOF parallel robot with decoupled structure [9]. It is composed of four identical legs, actuated by a linear joint (see Figure 11). The first three actuators generate three independant translations, and a different height of the two vertical joints creates a rotation of the end-effector. This robot has a closed-form Forward Kinematic Model and Instantaneous Kinematics, contrary to most parallel robots.

Its main goal is High-Speed Machining with a desired acceleration of $20m.s^{-2}$. The required stiffnes imposes this robot a heavy structure ($31kg$ per leg and $14kg$ for the end-effector). Inertias of the leg are very important, contrary to others parallel robots with very moving elements (H4[4], Orthoglide[25]...). Consequently, the dynamics need be taken into account in control to deal with the important dynamic coupling between legs.

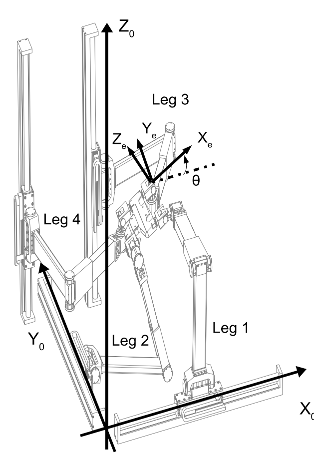

Fig. 11. *Global view of the Isoglide-4 T3R1*

2) Kinematics: The Isoglide-4 T3R1 has a very simple Inverse Kinematic Model, and algebraic Forward Kinematic Model [9]:

$$\left\{\begin{array}{ccc} X_e & = & q_1 + X_{01} - b_5 \\ Y_e & = & q_2 + Y_{02} + H + b_5 \\ Z_e & = & q_3 + Z_{03} + b_5 \\ \sin\theta & = & \frac{q_4 + Z_{04} - q_3 - Z_{03}}{L} \end{array}\right. \tag{9}$$

where :

- $[X_e\ Y_e\ Z_e\ \theta]^T$ are the Cartesian variables of the end-effector
- q_i is the active joint variable of leg i
- X_{01}, Y_{02}, Z_{03}, Z_{04} are the initial position of the actuators in the fixed frame (active joint offsets)
- H and L are dimensions of the effector
- b_5 is the length of the link between a leg and the end-effector

The Forward Instantaneous Kinematics matrix of the robot has also a closed-form expression, obtained by differentiating equation (9):

$$D_{inv}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{\cos\theta}{L} & \frac{\cos\theta}{L} \end{pmatrix} \tag{10}$$

3) Dynamics: The robot Inverse Dynamic Model is quite big, but have a closed-form expression depending on the end-effector pose. Therefore, each term can be interpretated and a posteriori simplifications can be done instead of a priori in the Lagrange approach. In each actuator, the main term is the inertia force of the leg and the end-effector. Then, the other influent terms concern centrifugal and Coriolis forces due to the rotation of the end-effector, arms and forarms.

IV. Results

Since the vision sensor is not yet integrated in the robot architecture, only simulation results can be provided. A comparison between proposed vision-based Computed Torque Control and classical Computed Torque Control in the cartesian space is set up for the Isoglide-4 T3R1. The dynamic behaviour of the robot is represented by the Forward Dynamic Model, obtained with the same method as used for the Inverse Dynamic Model. In the simulation, geometrical errors are fixed to 50μm according to manufacturing specifications and dynamic parameter errors to 10%, as it is generally achieved with classical identification. The sampling rate of the control is $1kHz$. The reference trajectory is a linear $100mm$ displacement with $20m.s^{-2}$ maximal acceleration and $1.2m.s^{-1}$ maximal velocity along the three axes X, Y and Z, with a 30° rotation around axe Y. A fifth degre interpolation is used to obtain a smooth trajectory.

On the one hand, the vision sensor has an incertitude of $100\mu m$ and 0.1° on the end-effector pose and a 1% error on velocity measurements. The camera is supposed to allow a $1kHz$ sampling rate acquisition. With these sensor performances, a maximal tracking error of 300μm and 0.08° is archieved (See Figures 12 and 13). On the other hand, a Computed Torque Control based on pose estimation, via the Forward Kinematic Model, allows a precision of 200μm and 0.11°, with a joint sensor accuracy fixed to $1\mu m$. But, there is an important bias of $5mm$ and 1.6° which has to be compensated for. Hence, a better calibration is needed contrary to vision-based control where geometrical errors are compensated for. Furthermore, this simulation does not take into account the structure defaults and deformations which decrease the accuracy of the Computed Torque Control with a Forward Kinematic Model. These defaults can be compensated for with the visual measurement. In addition, in the Isoglide-4 T3R1 case, the Forward Kinematic Model has a closed form expression, hence there are no computation problems. However, it can be interresting to apply this simulation for a robot with a numerical Forward Kinematic Model. Finally, it can be noticed that the tracking error is three times the measure error in the visual case, and two hundred times in the Forward Kinematic Model case.

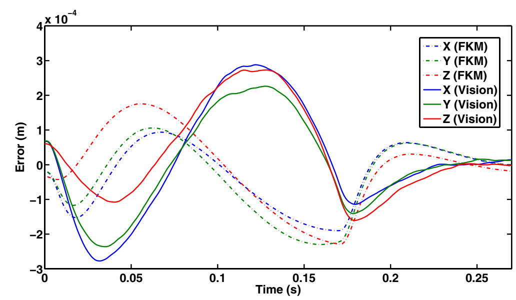

Fig. 12. *Cartesian position error for cartesian and $1kHz$ vision-based CTC*

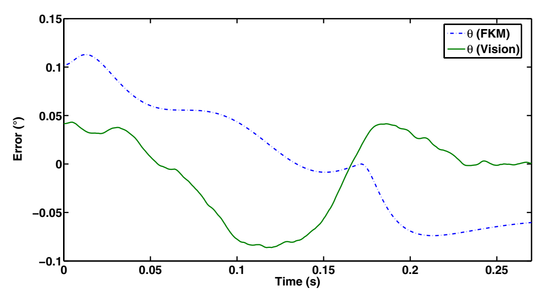

Fig. 13. *Orientation error for cartesian and $1kHz$ vision-based CTC*

Nevertheless, these good performances can only be archieved if vision sensor has a $1kHz$ sampling rate. At the moment, a $100Hz$ sampling rate is possible. Using a simple zero order hold between two images, the accuracy is reduced and reaches: 3.6mm and 1° (Figures 14 and 15). However, it can be noticed that controller gains have to be decreased. In this case, interpolation techniques to estimate the end-effector pose between two images can be used to improve accuracy.

V. Conclusion

In this article, we present a novel approach for the Computed Torque Control of parallel robots. We used a dynamic

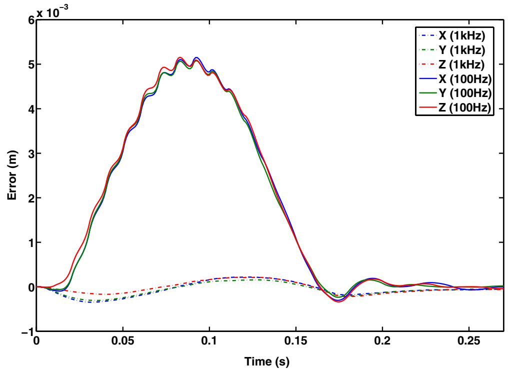

Fig. 14. *Tracking error for cartesian position with a 100Hz camera*

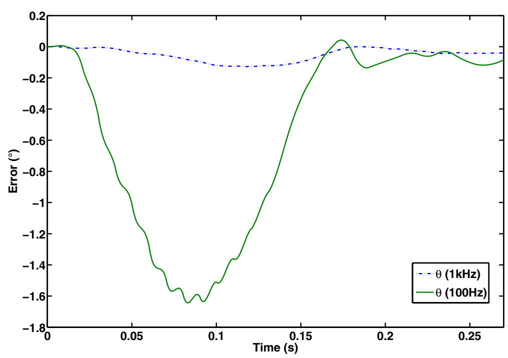

Fig. 15. *Tracking error for rotation θ with a 100Hz camera*

modeling method based on Newton-Euler equation allowing easy computation of the Inverse Dynamic Model which depends on the end-effector pose, velocity and acceleration. After discussing the limits of Computed Torque Control in the joint space and Computed Torque Control in the Cartesian space based on the Forward Kinematic Model, we propose a control scheme based on a visual measurement of the end-effector pose and velocity. This results in a very simple control scheme, where precise and reliable visual feedback for the end-effector pose and velocity increases control performance and accuracy. In addition, structure deformations and defaults can be easily compensated for. Simulation results also show that even if used vision sensor is not yet very competitive in term of precision and sampling rate, tracking errors are not that important. So with a performant visual sensor with great accuracy and high sampling rate, great performance can be expected without joint sensor.

Future work will focus on experimental validation with several parallel robots. In the current context, since vision sensor is not yet as competitive as we may wish it, special work could be done to care with downsampling issues. A solution based on non-linear observer could be envisaged to predict or estimate the end-effector between two images.

References

[1] M. Ait-Ahmed and M. Renaud. Dynamic modelling of closed-chain mechanism and its application for a 6 dof actuated manipulator. In *International Conference in Electronics and Automatic Control*, pages pp. 203–209, 1992.
[2] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet. Pose and velocity computing of fast moving object using a single view from rolling shutter camera. In *IEEE International Conference on Computer Vision Systems (to appear)*, 2006.
[3] 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.
[4] O. Company. *Machines-outils rapides à structure parallèle. Méthodologie de conception, applications et nouveaux concepts*. Thèse, Université Montpellier II, 2000.
[5] P. Dauchez, X. Delebarre, and R. Jourdan. Hybrid control of a two-arm robot handling firmly a single rigid object. In *2nd SIFIR*, pages pp 67–62, 1989.
[6] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2):pp. 123–141, 1995.
[7] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, Vol 8(3), 1992.
[8] J. Gangloff. *Asservissements visuels rapides d'un robot manipulateur à six degrés de libertés, Suivi de cible et de profilé*. Thèse, Université Louis Pasteur, Strasbourg, 1999.
[9] G. Gogu. Fully-isotropic T3R1-type parallel manipulators. *On Advances In Robot Kinematics*, pages 265–272, 2004.
[10] V.E. Gough. Contribution to discussion of papers on research in automobile stability, control and tire performance. *Proc. of Auto Div. Inst. Mech. Eng.*, 1956-57.
[11] M.L. Husty. An algorithm for solving the direct kinematic of the Gough-Stewart platforms. *Rapport de recherche TR-CIMS-94-7, Université de McGill, Montréal, Canada*, 1994.
[12] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. London, 2002.
[13] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. In *ICRA'04 : IEEE International Conference on Robotics and Automation*, pages 3665–3670, 2004.
[14] H. Kino, C.C. Cheah, S. Yabe, S. Kawamua, and S. Arimoto. A motion control scheme in task oriented coordinates and its robustness for parallel wire driven systems. In *Int. Conf. Advanced Robotics (ICAR'99)*, pages pp. 545–550, 1999.
[15] Y-W Li. Inverse dynamics and simulation of a 3-DOF spatial parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages pp. 4092–4097, 2003.
[16] D. G. Lowe. Fitting parameterized three-dimensional models to image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):pp. 441–450, 1991.
[17] J.Y.S Luh, M.W. Walker, and R.C.P Paul. On-line computational scheme for mechanical manipulators. *Transaction of ASME, J. of Dynamic Systems, Measurement, and Control*, 102(2):69–76, 1980.
[18] J.P. Merlet. An algorithm for the forward kinematics of general 6 d.o.f parallel manipulator. Rapport de recherche, INRIA, 1990.
[19] J.P. Merlet. *Parallel Robots*. Kluwer Academic Publishers, 2000.
[20] J.P. Merlet. Still a long way to go on the road for parallel mechanisms. In *ASME 27th Biennial Mechanisms and Robotics Conference*, 2002.
[21] R.C.P Paul. Robot manipulators : mathematics, programming and control. *MIT Press*, 1981.
[22] D. Stewart. A platform with 6 degrees of freedom. *Proc. of the Inst. of Mech. engineers*, pages 371–386, 1965.
[23] J. Tlusty, J. Ziegert, and S. Ridgeway. Fundamental comparison of the use of serial and parallel kinematics for machine tools. *Annals of the CIRP*, Vol 48(1):pp 351–356, 1999.
[24] L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, Vol. RA-3(5):pp. 404–417, 1987.
[25] P. Wenger and D. Chablat. Kinematic analysis of a new parallel machine tool : the Orthoglide. *Advances in Robot Kinematic*, pages 305–314, 2000.