

Control of Parallel Robots: Towards Very High Accelerations

Ahmed Chemori, Guilherme Sartori Natal, François Pierrot

▶ **To cite this version:**

Ahmed Chemori, Guilherme Sartori Natal, François Pierrot. Control of Parallel Robots: Towards Very High Accelerations. SSD 2013 - 10th International Multi-Conference on Systems, Signals and Devices, Mar 2013, Hammamet, Tunisia. lirmm-00809514

HAL Id: lirmm-00809514

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00809514

Submitted on 9 Apr 2013

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.

2013 10th International Multi-Conference
on Systems, Signals & Devices (SSD)
Hammamet, Tunisia, March 18-21, 2013

SSD'13 1569725643

Control of Parallel Robots: Towards Very High Accelerations

Ahmed Chemori, Guilherme Sartori-Natal and Franois Pierrot
LIRMM, Université Montpellier Sud de France (UMSF) - CNRS
161 Rue Ada
34392, Montpellier, France
Email: Ahmed.Chemori@lirmm.fr

***Abstract*—This paper deals with control of parallel manipulators for very high acceleration tasks including two case studies. The first one is the case of non-redundant parallel manipulators, where we are interested in control of `PAR2` robot for 2D pick-and-place trajectories following. The proposed control scheme in this case is a nonlinear dual mode adaptive controller complied with a high gain state observer for the estimation of joint velocities. The second case study concerns redundantly actuated parallel manipulators, where we are interested in control of `R4` robot for tracking of 3D pick-and-place trajectories. The proposed solution in this case is a dual-space adpative controller. The proposed controllers have been implemented in real-time on the two prototypes, and their effectiveness has been demonstrated through the obtained experimental results.**

I. Introduction

During the last few decades, parallel robots have attracted a considerable attention from different research communities. It must be mentioned, however, that the first design patent of a parallel robot (a motion platform for the entertainment industry) was applied by J. Gwinnett in 1928, being issued in 1931 [2]. The first industrial parallel robot (for automated spray painting) to be built was patented by W. Pollard in 1942. A few years later (more precisely in 1947), the parallel robot that became the most popular in the industry of that time (the variable-length-strut octahedral hexapod for tire testing) was invented by E. Gough. During the 60's, it was K. Cappel who independently designed the very same hexapod, had it patented and licensed to the first flight simulator companies, making it the first commercial octahedral hexapod motion simulator. Yet, it was D. Stewart who, unintentionally, made Gough's concept popular and proposed, once again, this idea for flight simulators, machine tools and universal milling machines [2].

In the early 80's, a new type of parallel robots was invented by R. Clavel, namely the Delta robot [1]. Its basic design idea consists in the use of parallelograms, which allows an output link to remain at a fixed orientation with respect to an input link. Since then, several robots, based on the same concept, have been designed, with several applications ranging from packaging industrial tasks (pick-and-place/assembly tasks), to the medical domain (i.e. carry a 20 kg microscope for surgery purposes) and even to the entertainment domain (haptic game controllers).

The known advantages of parallel robots, in comparison to their serial counterparts, are their higher stiffness and lighter structures, which allow them to reach very high velocities and accelerations. In order to achieve such high accelerations while perform an accurate movement, a good controller must also be designed. By using classical linear single-axis controllers (such as a Proportional Derivative (PD) controllers), the tracking performance are usually limited, especially when the robot behaves with high nonlinear dynamics and when the operating parameters are time varying [7]. In these cases, a more advanced (nonlinear/adaptive) controllers are necessary.

In the literature, different nonlinear/adaptive controllers have already been proposed for the control of parallel robots. For instance, in [7], a nonlinear adaptive feedforward controller was proposed for the control of `Hexaglide` (a 6 dof parallel robot), in addiction to a PD feedback term. The main objective of this work was to show the convergence of the adaptive parameters in simulation. In [6], a control scheme similar to the so called computed torque controller [5] was also proposed to control the `Hexaglide` robot, but with real-time experimental results that show a good improvement in the trajectory tracking w.r.t. a PD controller, although no control signal has been presented.

In most control algorithms, proposed in the literature, it is assumed that the joint velocities are available, that is, they need to be either calculated or estimated. The easiest way to compute them consists in a straight numerical derivative of the measured articular positions. However, if the measurements are noisy, this technique will significantly amplify the noise/quantization effects. In order to overcome this problem, an estimation of the articular velocities by means of a state observer is better to be considered. The choice of the estimation mechanism is strongly influenced by the existence of uncertainties in the system model. Whereas model-based observers are usually restricted to cases where the model and its parameters are well known, filters can provide a model-independent means of estimating velocities [13].

In this paper, we are interested in control of parallel manipulators, where two case studies are considered. The first case concerns the control of the non-redundant parallel manipulator `PAR2` for 2D pick-and-place trajectory tracking. The proposed control scheme in this case is a nonlinear dual mode adaptive controller complied with a high gain state observer for the estimation of the joint velocities. The second case study deals with the control of the redundantly actuated parallel

978-1-4673-6457-7/13/$31.00 ©2013 IEEE

manipulator R4, where a dual-space adpative controller is proposed for the tracking of 3D pick-and-place trajectories.

This paper is organized as follows. In next section the experimental platforms of our demonstrators are described. Section III concerns the description and dynamic modeling of parallel manipulators PAR2 and R4. The proposed control solutions for both cases are detailed in section IV. Section V is devoted to the presentation and discussion of the experimental results. The paper ends with some concluding remarks.

II. Description of Experimental Platforms

In this section the experimental platforms of both parallel manipulators PAR2 and R4 are presented.

A. The non-redundant case: PAR2 robot

The real-time experimental testbed is displayed in Figure 1, where:

- the computer ① is used for the development of the control algorithms,
- the power supply and drivers are embedded in box ②,
- the item ③ represents the emergency stop,
- the item ④ represents Par2 parallel manipulator.



Figure 1. View of the experimental setup of Par2 parallel manipulator

A more detailed description of the PAR2 robot manipulator in terms of mechanical structure and dynamic modeling is presented in section III-A.

B. The redundantly actuated case: R4 robot

The experimental setup of parallel manipulator R4 is displayed in Figure 2, where:

- the computer ① is used for the development of the control schemes in Matlab/Simulink environment,
- the dedicated target PC ②, is used for the real-time execution of control algorithm,
- the item ③ represents the emergency stop button,
- the R4 parallel manipulator is represented by item ④.

The proposed control schemes have to be implemented in Matlab/Simulink environment, being compiled using XPC Target real-time toolbox, and uploaded to the target PC. This last one manage the real-time task execution with a sampling frequency of 10 KHz (sample time of 0.1 *msec*). The Cartesian position of the platform is calculated based on the forward kinematics of the robot, and the Cartesian velocity is obtained through numerical derivation of the calculated Cartesian position. A more detailed description of the R4 robot manipulator in terms of mechanical structure and dynamic modeling is presented in section III-B.

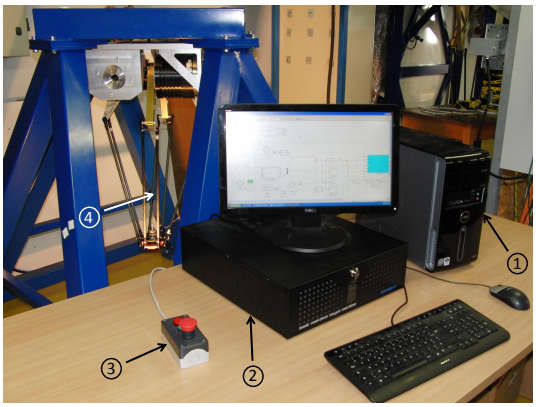


Figure 2. View of the experimental setup of R4 parallel manipulator

III. Description and Modeling of our Demonstrators

A. The non-redundant case: PAR2 robot

Par2 is a two-dof parallel manipulator, illustrated in Figure 3, with the following characteristics:

- the travelling plate ⑥ is a rigid body,
- only the two inner arms ③ are actuated,
- the two other inner arms ④ are linked to the frame ① through passive revolute joints,
- the four inner arms are connected to platform ⑥ with pairs of rods ⑤ mounted on ball joints ⑦,
- the passive inner arms ④ are coupled to guarantee planar motions in the plane xoz.

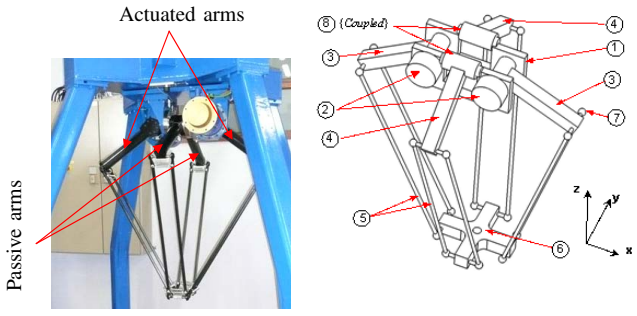


Figure 3. The two-dof parallel manipulator Par2: view of the robot (left), schematic view of its mechanical structure (right)

The functioning of this two-dof parallel robot is ensured by the coupling of the rotation of passive arms ④. This is capable of constraining the robot's platform to evolve in one plane. The *coupling* implies that the rotation of the first arm

in the clockwise direction involves the rotation of second one in the counterclockwise direction.
The Lagrangian nonlinear dynamic model [9], [12] of the manipulator Par2 is given by:

$$I_{eq}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} = \tau - f_d\begin{bmatrix}sign(\dot{q}_1)\\ sign(\dot{q}_2)\end{bmatrix} - f_v\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} - \frac{g}{2}(M_1+M_2)\begin{bmatrix}L_1\cos(q_1)\\ L_2\cos(q_2)\end{bmatrix} + J^T\left(M_p\dot{J}(q,\dot{q})\dot{q} - g\begin{bmatrix}M_p\\ M_p\end{bmatrix}\right) \quad (1)$$

where $I_{eq} = I_{drv} + I_f + I_a + J^T M_p J$, I_{drv} is the motor driver inertia, I_f is the forearm inertia, I_a is the arm inertia, J is the jacobian matrix, M_p is the mass of the platform of the robot, g is the gravitational acceleration, f_v is the viscous friction coefficient, f_d is the dry friction coefficient, M_1 and M_2 are the masses of the arms, L_1 and L_2 are their lengths.
This dynamic model can be written in the following standard matrix form:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f(q,\dot{q}) = \tau \quad (2)$$

where:

$D(q) \in \mathbb{R}^{2\times 2}$ is the inertia matrix,
$C(q,\dot{q}) \in \mathbb{R}^{2\times 2}$ is the Coriolis and centrifugal matrix,
$G(q) \in \mathbb{R}^2$ is the vector of gravitational forces,
$f(q,\dot{q}) \in \mathbb{R}^2$ is the vector of friction forces,
$\tau \in \mathbb{R}^2$ is the vector of torques generated by the actuators,
$q = \begin{bmatrix}q_1\\ q_2\end{bmatrix} \in \mathbb{R}^2$ is the vector of articular positions,
$\dot{q} = \begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} \in \mathbb{R}^2$ is the vector of articular velocities,
$\ddot{q} = \begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} \in \mathbb{R}^2$ is the vector of articular accelerations.

B. The redundantly actuated case: R4 robot

The R4 robot is a redundantly actuated parallel manipulator (with three dof and four actuators) that was designed to have the capability of reaching $100G$ of acceleration. This robot has a workspace of at least a cylinder of $300\,mm$ radius and $100\,mm$ height, and each of its four motors has a maximum torque of $127\,N.m$. Its CAD schematic as well as its side view are shown in Figure 4. Its dynamic parameters are summarized in table I.
In the dynamic modelling of R4 parallel manipulator, some

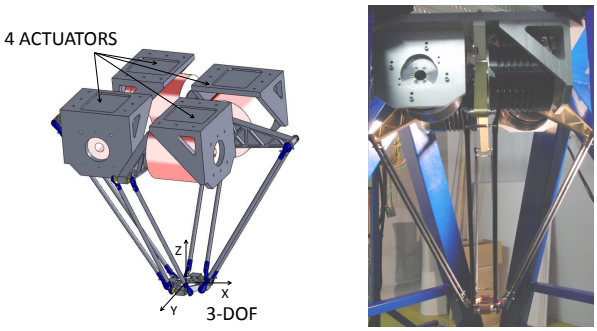


Figure 4. The R4 parallel manipulator: Schematic view of the CAD design (left), side view of the robot prototype (right)

Table I
DYNAMICS PARAMETERS

M_{tp} [kg]	$M_{forearm}$ [kg]	I_{act} [kg.m^2]	I_{arm} [kg.m^2]
0.2	0.065	0.003	0.005

simplifications were made, they are based on the following hypotheses:

H1 : the joint frictions were neglected, as the components of the robot were designed such that they would have very small frictions between them,
H2 : the inertia of the forearms was also neglected, and their masses were split up into two parts each being artificially considered to be located at both ends of the forearms (half of the mass is transferred to the end of the arm, whereas the other half is transferred to the traveling plate),
H3 : gravity acceleration was not taken into account since the case studies considered very high accelerations.

Taking into account these assumptions, the final expression of robot's simplified forward dynamics is derived from a combination of the arms and the traveling plate equilibriums, and is given by the following equation [4]:

$$\ddot{x} = (M_T + {J_m}^T I_T J_m)^{-1} J_m^T(\Gamma - I_T \dot{J}_m \dot{x}) \quad (3)$$

where $\dot{x} \in \mathbb{R}^m$ and $\ddot{x} \in \mathbb{R}^m$ are the vectors of Cartesian velocities and accelerations; $M_T = Diag\{M_{tp} + n\frac{M_{forearm}}{2}\}_{m\times m} = M_{tot} I_{m\times m}$ is a diagonal matrix with m diagonal terms, being M_{tp} the mass of the traveling plate, $M_{forearm}$ the mass of the forearm, M_{tot} the scalar value of the diagonal of M_T, m the number of degrees-of-freedom ($m = 3$) and n the number of motors ($n = 4$); $I_T = Diag\{I_{act} + I_{arm}\}_{n\times n} = I_{tot} I_{n\times n}$ is a diagonal matrix with n diagonal terms, where I_{act} and I_{arm} are the inertia of the actuators and the inertia of the arms, respectively, and I_{tot} is the scalar value of the diagonal of I_T; $J_m \in \mathbb{R}^{n\times m}$ and $\dot{J}_m \in \mathbb{R}^{n\times m}$ are the generalized inverse Jacobian matrix and its first derivative, respectively; and $\Gamma \in \mathbb{R}^n$ represents the torques generated by the actuators.
By multiplying both sides of (3) by $(M_T + {J_m}^T I_T J_m)$, one has:

$$(M_T + {J_m}^T I_T J_m)\ddot{x} = {J_m}^T(\Gamma - I_T \dot{J}_m \dot{x}) \quad (4)$$

which can be rewritten as follows:

$$\Gamma = H^T M_T \ddot{x} + I_T(J_m \ddot{x} + \dot{J}_m \dot{x}) = H^T M_T \ddot{x} + I_T \ddot{q} \quad (5)$$

IV. PROPOSED CONTROL SCHEMES

A. The non-redundant case: A dual mode adaptive controller

A Dual Mode adaptive controller consists basically in using a high gain adaptation together with a projection of the estimated parameters to bound their variation. Then, to large tracking errors in the transitory, the controller behaves approximately as a sliding mode controller, generating an exponential convergence to a residual domain arbitrarily small, and to smaller errors, it behaves as a parametric adaptation law. The other advantages of the adaptation law in dual mode

control with respect to other adaptation techniques or known robust control algorithms are the following:

- the generated control signals are continuous,
- the robustness of the system is improved,
- the variation of the estimated parameters is bounded thanks to the projection, which reduces the effective gain of the controller. Consequently, the sensitivity to measurement noise is reduced.

Consider now the following errors definition [10]:

$$s = \dot{\tilde{q}} + \lambda \tilde{q} \quad ; \quad \dot{q}_r = \dot{q}_d - \lambda \tilde{q} \tag{6}$$

where s is an auxiliary error, λ is a positive constant, and $\dot{q}_r$ is the *'reference velocity'* [10],[11]. The control architecture of the dual mode adaptive controller is illustrated in Figure 5. The control input consists in the sum of three terms, namely

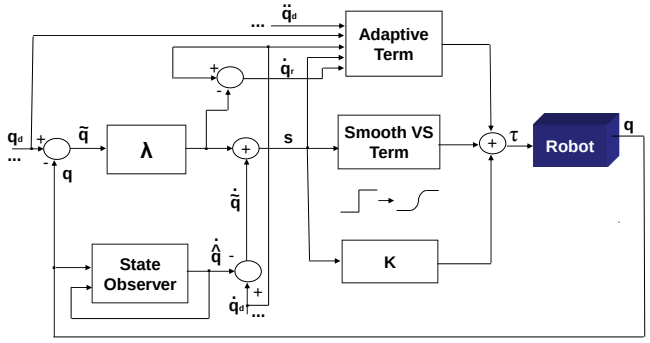


Figure 5. Block diagram of the Dual Mode control scheme

an "*adaptive term*", a "*smooth variable structure term*" and a "*stabilizing term*", it is given by:

$$\tau = Y\hat{a} + \bar{d}Sat(\alpha s) + K.s \tag{7}$$

where

$$Sat(\alpha s) = \frac{\alpha s}{||\alpha s|| + 1} \tag{8}$$

$\bar{d}$, α and K are positive constants. The function $Sat(.)$ is continuous with respect to its argument, it is worth to note that the higher the chosen α, the closer this function will be to a $sgn(.)$ function. The vector $\hat{a}$ represents the vector of the estimated parameters of the system given by the vector a, and Y is the regressor vector (computed from the dynamic model of the system). The adaptation law is given by the following:

$$\dot{\hat{\theta}} = -\sigma\hat{\theta} - \gamma sY \tag{9}$$

where

$$\hat{\theta} = \hat{a} - a_{nom} \tag{10}$$

is the difference between the currently estimated values $\hat{a}$ and the nominal values a_{nom} of the parameters. σ is expressed by the following:

$$\sigma = \begin{cases} 0, & \text{if } ||\hat{\theta}|| < M_\theta \text{ or } \sigma_{eq} < 0 \\ \sigma_{eq}, & \text{if } ||\hat{\theta}|| \geq M_\theta \text{ and } \sigma_{eq} \geq 0 \end{cases} \tag{11}$$

$$\sigma_{eq} = -\frac{\gamma sY\hat{\theta}}{||\hat{\theta}||^2} \tag{12}$$

where M_θ is the maximum possible value (supposed known) of the estimated deviation of the parameters in relation to their nominal values. If we consider the dynamic model (2) with the estimates parameters $\hat{a}$:

$$D(q,\hat{a})\ddot{q} + C(q,\dot{q},\hat{a})\dot{q} + G(q,\hat{a}) + f(q,\dot{q},\hat{a}) = \tau \tag{13}$$

then the regressor $Y = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)$ can be computed as:

$$D(q,\hat{a})\ddot{q}_r + C(q,\dot{q},\hat{a})\dot{q}_r + G(q,\hat{a}) + f(q,\dot{q},\hat{a}) = Y\hat{a} \tag{14}$$

where $\dot{q}_r$ and $\ddot{q}_r$ are used instead of $\dot{q}$ and $\ddot{q}$. The key idea relies in eliminating undesirable steady-state position errors by restricting them to evolve on a sliding surface [10] like in sliding mode control [3]. If we consider neglecting M_{tot} (to avoid the computation of heavy multiplications between the Jacobian and their derivative), these terms become $D(q,\hat{a}) = I_{motor} + I_{arm}$, $f(q,\dot{q},\hat{a}) = f_v$ and:

$$G(q,\hat{a}) = -\frac{g}{2}(M_{arm} + M_{forearm}) \begin{bmatrix} l_1 cos(q_1) \\ l_2 cos(q_2) \end{bmatrix} \tag{15}$$

where $I_{motor} = Diag\{I_m, I_m\}$ and $I_{arm} = Diag\{I_a, I_a\}$. The robot's dynamics rewritten in the form of (14), where:

$$Y = \begin{bmatrix} \ddot{q}_{1r} & 0 & \dot{q}_{1r} & -l_1 cos(q_1) \\ 0 & \ddot{q}_{2r} & \dot{q}_{2r} & -l_2 cos(q_2) \end{bmatrix} \tag{16}$$

and the estimated parameters are:

$$\hat{a} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \end{bmatrix} = \begin{bmatrix} I_m + I_a \\ I_m + I_a \\ f_v \\ \frac{g}{2}(M_{arm} + M_{forearm}) \end{bmatrix} \tag{17}$$

The velocity measurements are not available, therefore a high gain observer is proposed and implemented on the system to estimate them.

B. The redundantly actuated case: A dual-space adaptive controller

The proposed control scheme, for this case, is a dual-space adaptive controller. This control scheme is an extended version of the dual-space feedforward controller illustrated in Figure 6. The basic idea if the dual-space feedforward controller consists in using a PID controller in Cartesian space, with the utilization of the pseudo-inverse matrix to deal with the actuation redundancy. Then two feedforwards in both spaces (joint and Cartesian) are used to improve the tracking performance of the controller. In order to enhance the robustness of the control scheme an adaptation technique inspired from [8] is proposed to extend the dual-space feedforward controller. The obtained scheme is named *'dual-space adaptive controller'* and is detailled in the following.

Based on the dynamics (5), the following control input is proposed:

$$\Gamma = H^T\hat{M}_T\ddot{x}_d + \hat{I}_T\ddot{q}_d + K_p e + K_d\dot{e} \tag{18}$$

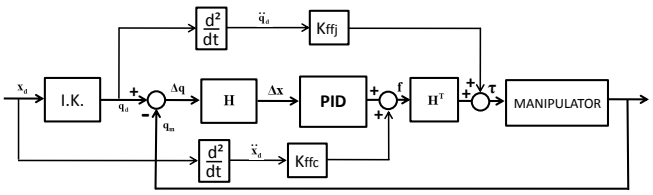


Figure 6. The dual-space feedforward controller

this expression can be rewritten in the Cartesian space as:

$$F = Y_r\hat{\theta} + K_p e_c + K_d \dot{e}_c \tag{19}$$

K_p and K_d are positive feedback gains, $e_c = x_d - x$, $\dot{e}_c = \dot{x}_d - \dot{x}$, and:

$$Y_r = \left[\begin{array}{cc} I_{3\times3}\ddot{x}_d & J_m^T I_{4\times4}\ddot{q}_d \end{array}\right];\ \hat{\theta} = \left[\begin{array}{c} \hat{M}_{tot} \\ \hat{I}_{tot} \end{array}\right] \tag{20}$$

where Y_r and $\hat{\theta}$ the regressor vector and the estimated parameters vector, respectively. The adaptation of the estimated parameters is performed according to the following adaptation rule ($i = 1,2$):

$$\dot{\hat{\theta}}_i = \begin{cases} \gamma_{ii}\phi_i, & \text{if } \begin{array}{l} a_i < \hat{\theta}_i < b_i \text{ or} \\ \hat{\theta}_i \geq b_i \text{ and } \phi_i \leq 0 \text{ or} \\ \hat{\theta}_i \leq a_i \text{ and } \phi_i \geq 0 \end{array} \\ \gamma_{ii}(1+\frac{b_i-\hat{\theta}_i}{\delta}), & \text{if } \hat{\theta}_i \geq b_i \text{ and } \phi_i \geq 0. \\ \gamma_{ii}(1+\frac{\hat{\theta}_i-a_i}{\delta}), & \text{if } \hat{\theta}_i \leq a_i \text{ and } \phi_i \leq 0. \end{cases} \tag{21}$$

with

- γ_{ii}: the i^{th} element of the diagonal adaptation gain matrix γ,
- $\hat{\theta}_i$: the estimates of each unknown parameter,
- a_i and b_i: the lower and upper bounds of each estimative, respectively,
- ϕ_{ii}: the i^{th} element of the column matrix $\phi = -Y_r^T s$; being $s = \dot{e}_c + \lambda e_c$, where $\lambda = \frac{\lambda_0}{1+||e||}$, being λ_0 and δ positive constants.

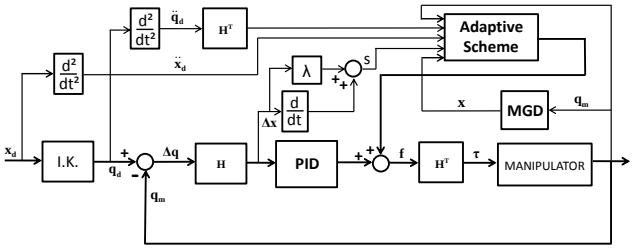


Figure 7. The proposed dual-space adaptive controller ($\tau = \Gamma$)

The adaptive gains are $\gamma_{11} = 0.2$ and $\gamma_{22} = 1.5\text{x}10^{-4}$. The best K_{ffc} gains of the feedforward controller were 0.625 for the case without a payload and 0.825 for the case with a payload of $200\,g$, and bigger payloads may be used in future experiments, the chosen range for the $\hat{M}_{tot}$ parameter (in kg) was of $[0.525;1]$, which means $a_1 = 0.525$ and $b_1 = 1$. Concerning the inertia $\hat{I}_{tot}$ parameter (in $kg.m^2$), which is equivalent to the K_{ffj} feedforward gain, the range was chosen as $[0.006;0.018]$, which means $a_2 = 0.006$ and $b_2 = 0.018$. The parameters δ and λ were equal to 0.0001 and 100, respectively. In this work, more attention is given to the behaviour of the $\hat{M}_{tot}$ parameter, as this is the parameter that directly compensates for the load changes.

V. Real-time experimental results

A. The non-redundant case: 2D pick-and-place task

The proposed dual mode adaptive controller is compared with a PD controller. The trajectory tracking obtained by both controllers for $20G$ of maximum acceleration is shown in Figure 8. The corresponding tracking errors are displayed in Figure 9, and the control inputs in Figure 10. The real-time implementation of both controllers was considered with a sample time of $0.5\,msec$. The objective in this scenario was to obtain the best possible tracking accuracy, especially on stop points (final positions of the trajectory). These positions occur for $t \in [0.125, 0.175]s$ (final desired position) and for $t \in [0.325, 0.375]s$ (back to the initial desired position), being repeated in the equivalent intervals of the subsequent cycles. The used parameters of the proposed control approach are summarized in table II. The evolution of the control inputs

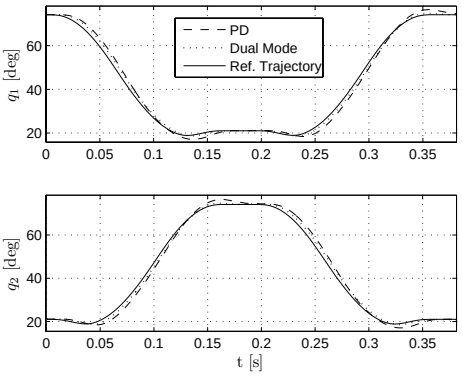


Figure 8. A pick-and-place trajectory tracking for $20G$ with a PD controller (dashed line) and the DM controller (dotted line)

Table II
Summary of the parameters of the Control Approach

Parameter	Description
$K_p = 94.5$	Proportional gain
$K_d = 2.1$	Derivative gain
$\lambda = 25$	Positive constant
$K = 2I$	Matrix gain
$d = 2.5$	Smooth variable structure gain
$\alpha = 0.05$	Smooth variable structure slope
$\varepsilon = 0.002,\ \alpha_{HGO1} = \alpha_{HGO2} = 1$	HGO gains
$M_\theta = 0.25$	Maximum adaptative parameters' error
$\gamma = 0.3345$	Adaptive gain
$T_s = 0.0005$	Sampling time (s)
$n = 3$	Number of cycles

(torques generated by controllers) is shown in Figure 10, where

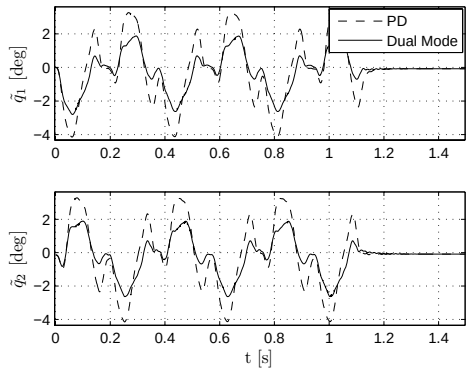


Figure 9. Tracking errors for 20G with a PD controller (dashed line) and the DM controller (solid line)

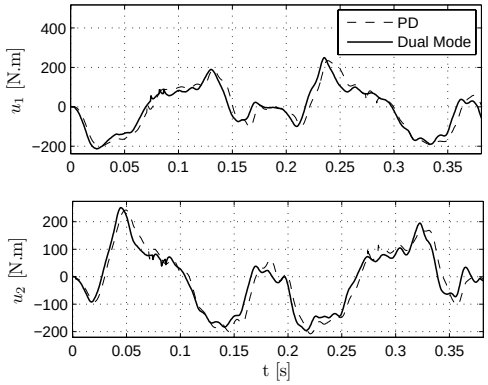


Figure 10. Control inputs versus time for the PD controller (dashed line) and the DM controller (solid line)

it is worth to note that the PD controller is slightly delayed w.r.t. the DM controller, and the amplitudes of both signals are roughly similar. Besides, it is important to emphasize that the generated torques are always within the admissible limits of the actuators (a maximum torque of approximately 500$N.m$). The performance comparison of both controllers is summarized in table III.

Table III
PERFORMANCE COMPARISON BETWEEN THE ADAPTIVE DUAL MODE AND A CONVENTIONAL PD CONTROLLER

Performance	**PD**	**DM**
Error peaks	$[-4.15°, 3.3°]$	$[-2.8°, 1.9°]$
Stop point errors	$[-2.3°, 2.3°]$	$[-0.7°, 0.7°]$
Control signals	PD controller delayed i.r.t. the DM controller	
	Roughly similar amplitude values	

B. The redundantly actuated case: 3D pick-and-place task

For this experiment, we consider the case of a 3D pick-and-place trajectory tracking, where both dual-space controllers (feedforward and adaptive) are compared. The trajectory to be tracked is illustrated in Figure 11. In the following experiments, a detailed comparison between the two controllers

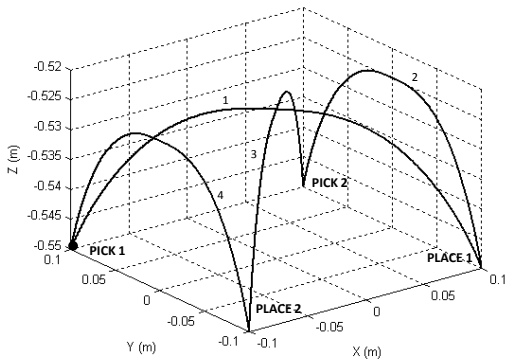


Figure 11. View of the 3D pick-and-place trajectory

is discussed. Firstly, the performance of both controllers is compared for the case with a payload of $200\,g$ at 20G, and then they are compared for the case without a payload at 30 G. For both scenarios, the robot has to perform two cycles of the proposed 3D pick-and-place trajectory.

1) Scenario with a payload of $200\,g$ *at* 20G*:*

In this experiment, the adaptive controller is compared with the feedforward controller (best configured for the case with a payload of $200\,g$, i.e $K_{ffc} = 0.825$).

The obtained results are presented in Figures 12-17.

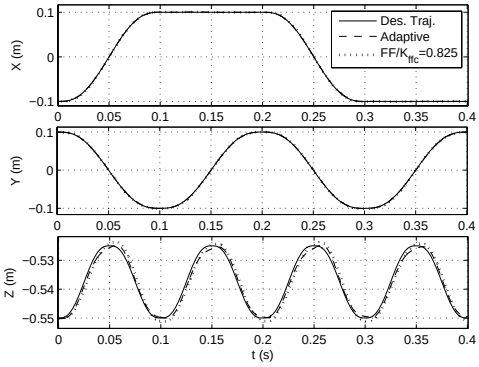


Figure 12. Trajectory tracking with a payload of $200\,g$, with 20G of maximum acceleration

Figure 12 shows the pick-and-place trajectory tracking obtained by both controllers and for the situation with a payload of $200\,g$, with a maximum acceleration of 20G. According to Figure 13, it is worth to note that the adaptive controller is able to provide a better overall tracking performance than the feedforward controller even with its best value of feedforward gains. For X, and Y axes, the adaptive controller is able to keep the tracking errors within the interval $[-1, 1]mm$, while the feedforward controller keeps them within $[-0.75, 1.6]mm$, as shown in table IV. For the Z-axis, the difference between the controllers is easily visible. Indeed, the adaptive controller keeps the tracking errors within $[-1.77, 2]mm$, while the feed-

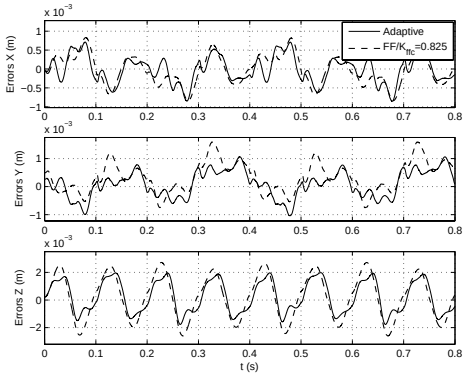


Figure 13. Tracking errors versus time with a payload of $200\,g$, with $20G$ of maximum acceleration

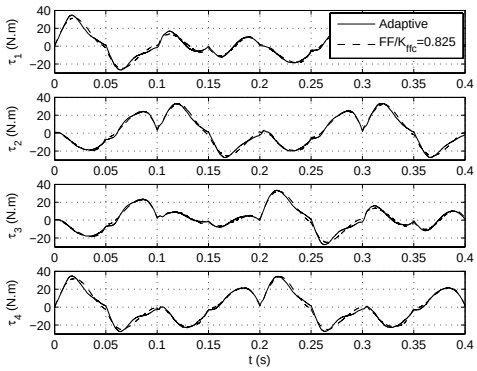


Figure 14. Torques generated by the actuators in case of a payload of $200\,g$, with $20G$ of maximum acceleration

forward controller keeps them within $[-2.6, 2.7]mm$. If we are interested in the dynamic accuracy of the trajectory tracking, the RMSE (*Root Mean Square Error*) can be considered as a criterion of comparison. This error takes into consideration the errors in all axes equally, it is equal to 1.33 *mm* for the adaptive controller, versus 1.7 *mm* for the feedforward controller. These results are summarized in table IV. The torques generated by each actuator are shown in Figure 14, which shows that the adaptive controller generates a signal with a slightly bigger amplitude than the feedforward controller. It is worth to emphasize that all torques remain far from the admissible limits of the actuators (a maximum torque of $127\,N.m$).

Table IV
TRACKING PERFORMANCE FOR A PICK-AND-PLACE TRAJECTORY WITH A PAYLOAD OF $200\,g$ AND $20G$ OF MAXIMUM ACCELERATION

Performance	**Adaptive**	**FF** $(K_{ffc} = 0.825)$
Error peaks (X-Y)	$[-1, 1]mm$	$[-0.75, 1.6]mm$
Error peaks (Z)	$[-1.77, 2]mm$	$[-2.6, 2.7]mm$
RMSE	1.33 *mm*	1.7 *mm*
Control Signals	Smooth/far from limits Adaptive controller: Slightly bigger amplitude	

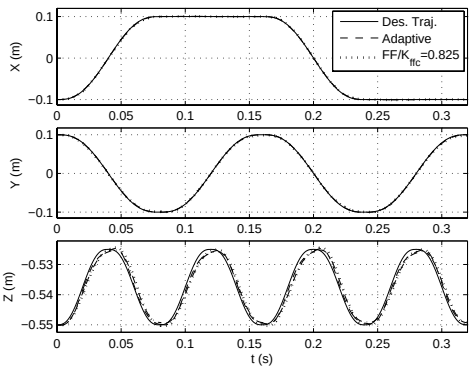


Figure 15. Trajectory tracking without a payload, with $30G$ of maximum acceleration

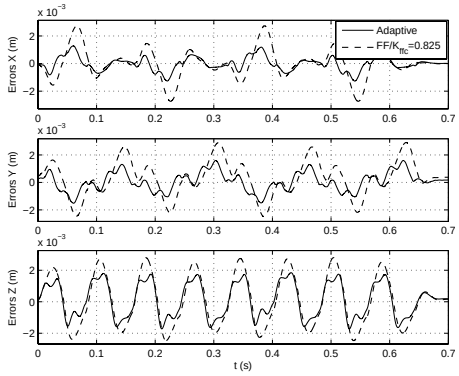


Figure 16. Tracking errors without a payload, with $30G$ of maximum acceleration

2) Scenario without a payload at $30G$*:*
In this experiment, the superiority in terms of robustness of the dual-space adaptive controller (w.r.t. the feedforward controller) is confirmed. Figure 16 shows that the feedforward controller, when not reconfigured to the new situation, has an important loss of tracking performance (both with respect to the previous scenario and also w.r.t. the adaptive controller). For the X-Y axes, the adaptive controller keeps the errors within $[-1.51, 1.6]mm$, while the feedforward controller keeps them within $[-2.73, 2.9]mm$ (peak-to-peak difference of more than 80%). The robustness of the adaptive controller and the lack of robustness of the feedforward controller are further confirmed by the RMSE results. The adaptive controller has almost the same RMSE as in the previous scenario (1.33 *mm* versus 1.4 *mm*), whereas the feedforward controller has a loss of almost 40% (i.e. 1.7 *mm* versus 2.4 *mm*). Figure 17 shows the evolution of the control inputs versus time, generated by both controllers. To sum up, these results are recapitulated in Table V.

3) Extreme scenario: Up to $100G$ *of maximum acceleration:*
In this experiment, the objective is validate one of the two

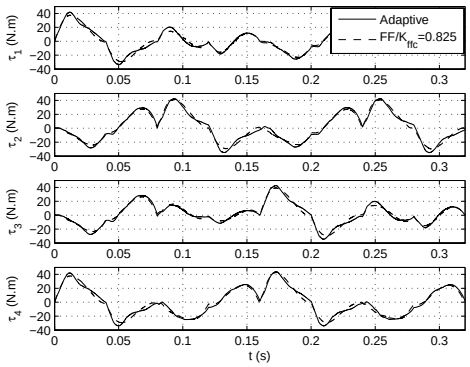


Figure 17. Torques generated by the actuators without a payload, with $30G$ of maximum acceleration

Table V
TRACKING PERFORMANCE FOR A PICK-AND-PLACE TRAJECTORY WITHOUT A PAYLOAD, WITH $30G$ OF MAXIMUM ACCELERATION

Performance	**Adaptive**	**FF** ($K_{ffc} = 0.825$)
Error peaks (X-Y)	$[-1.51, 1.6]mm$	$[-2.73, 2.9]mm$
Error peaks (Z)	$[-1.72, 1.84]mm$	$[-2.26, 2.77]mm$
RMSE	1.4 mm	2.4 mm
Control Signals	Smooth/far from limits Adaptive controller: Slightly bigger amplitude	

proposed controllers for an extreme scenario in terms of maximum acceleration. For that, the dual-space feedforward controller is chosen for a trajectory tracking with a maximum acceleration of $100G$. This trajectory corresponds to a vertical mouvement along the Z axis with an increasing amplitude until reaching a maximum of $10\,cm$ peak-to-peak, then it decreases until zero. The obtained result is displayed in Figure 18, where it is easily to see the steering from the rest position to the desired initial position, as well as the natural descent of the end-effector after the motors are turned off at the end of the experiment due to the gravity. The evolution of the Cartesian acceleration versus time is depicted in Figure 19, where we can clearly see the maximum acceleration of $100G$ (i.e. $1000\,m/s^2$) of the traveling plate of the robot.

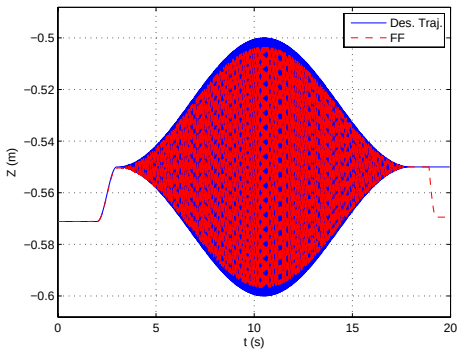


Figure 18. Trajectory tracking obtained with the dual-space feedforward controller for the $100G$ vertical trajectory

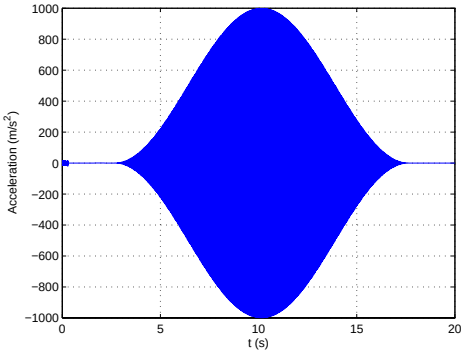


Figure 19. Evolution of the Cartesian acceleration for the $100G$ trajectory

VI. CONCLUSION

In this work we are interested in control of control of parallel manipulators to perform tasks with very high accelerations. The proposed study includes two main parts, namely the non-redundant and the redundantly actuated parallel manipulators. To resolve the control problem of the first part, the case of `PAR2` parallel manipulator has been considered. The proposed solution in this case was a dual-mode adaptive controller. The control problem of the second part was resolved in the case of `R4` parallel manipulator, where a dual-space adaptive controller was proposed. All the proposed control schemes have been implemented in real-time for different scenarios to show their effectiveness. The last experimental scenario demonstrates the tracking of a reference trajectory with a maximum extremely high acceleration of $100\,G$.

REFERENCES

[1] I. Bonev. Delta parallel robot - the story of success. *http://www.parallemic.org/Reviews/Review002.html*, 2001.

[2] I. Bonev. The true origins of parallel robots. *http://www.parallemic.org/Reviews/Review007.html*, 2003.

[3] S. K. Spurgeon C. Edwards. *Sliding mode control: theory and applications*. CRC Press, 1998.

[4] D. Corbel, M. Gouttefarde, O. Company, and F. Pierrot. Towards 100g with pkm. is actuation redundancy a good solution for pick-and-place? In *Proc. IEEE Conf. Robotics Automat.*, pages 4675–4682, 2010.

[5] J. J. Craig. *Adaptive control of mechanical manipulators*. Addison-Wesley Publishing Company, 1988.

[6] M. Honegger, R. Brega, and G. Schweitzer. Application of a nonlinear adaptive controller to a 6-dof parallel manipulator. In *Proc. IEEE Conf. Robotics Automat.*, volume 2, pages 1930–1935, 2000.

[7] M. Honegger, A. Codourey, and E. Burdet. Adaptive control of the hexaglide, a 6-dof parallel manipulator. In *Proc. IEEE Conf. Robotics Automat.*, pages 543–548, 1997.

[8] K. W. Lee and H. K. Khalil. Adaptive output feedback control of robot manipulators using high-gain observer. *Int. Journal of Control*, 67(6):869–886, 1997.

[9] L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators*. McGraw Hill, New York, 1996.

[10] J.J. Slotine and W. Li. Adaptive manipulator control: a case study. *IEEE Trans. on Automat. Contr.*, 33(11):995–1003, 1988.

[11] J.J. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall, 1991.

[12] M. Spong and M. Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, New York, 1989.

[13] B. Xian, M. S. Queiroz, D. M. Dawson, and M. L. McIntyre. A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems. *Automatica*, 40:695–700, 2004.