Adaptive Control of Parallel Manipulators: Design and Real-Time Experiments

Moussab Bennehar, Ahmed Chemori, Sébastien Krut, François Pierrot

In: Parallel Manipulators: Design, Applications and Dynamic Analysis ISBN 0000000000
Editor: Editor Name, pp. 1-30

Chapter 1

Adaptive Control of Parallel Manipulators: Design and Real-Time Experiments

M. Bennehar, A. Chemori, S. Krut and F. Pierrot[*]
LIRMM, 161 rue Ada, 34095 Montpellier cedex 5, France

Abstract

In this chapter, we address a new control strategy for parallel manipulators based on $\mathcal{L}_1$ adaptive control. This latter is known for its decoupled control and estimation loops, enabling fast adaptation and guaranteed robustness. To improve the tracking performance of parallel manipulators, we propose in this work to include the dynamic model of the robot in the control loop of $\mathcal{L}_1$ adaptive control. The additional dynamics-based term partially compensates for inherent nonlinear dynamics of the robot in order to reduce the impact of uncertainties on the closed-loop system and enhance the overall control performance. Real-time experiments, conducted on a 4-DOF fast parallel manipulator developed in our laboratory, demonstrate the effectiveness of the proposed controller and its superiority compared to standard $\mathcal{L}_1$ adaptive control in terms of tracking performance.

Keywords: Parallel manipulators, adaptive control, nonlinear system, dynamics.

[*]E-mail address: {bennehar, chemori, krut, pierrot}@lirmm.fr

1. Introduction

Adaptive control of parallel manipulators has been a decades-long area of research. It is a very relevant control strategy when it comes to control such systems due to the abundance of uncertainties and variations of their dynamics and the environment they interact with. Undoubtedly, using model-based controllers with constant dynamic parameters (e.g. computed torque [1], augmented PD [2], PD+ [3], etc.) significantly improves the overall performance of parallel manipulators and enhance their tracking capabilities. However, such control schemes are not endowed with automatic adjustment mechanisms (adaptation) to cope with variations and uncertainties in the dynamics of the controlled system and, hence, do not yield the expected performance. As a result, adaptive control seems to be a promising solution to account for such limitation by estimating and compensating the uncertainties in real-time to provide improved tracking and enhanced closed-loop performance.

The first attempts to apply adaptive control to mechanical manipulators were based on restrictive assumptions and hypotheses about their dynamics not necessarily reflecting the real properties of these systems. For instance, the proposed controller in [4], which is based on Model Reference Adaptive Control (MRAC), does not consider the coupling between the joints. Such hypothesis is questionable since mechanical manipulators are known for their coupled nonlinear dynamics. To relax this requirement, another MRAC-based controller has been proposed in [5]. The full dynamics of the manipulator as well as the coupling between the joints were explicitly considered and new adaptive laws were accordingly derived. Unfortunately for this controller, the number of parameters to be estimated is high, even for simple manipulators (11 parameters for the provided 3-DOF serial manipulator example).

One major reason of the failure of MRAC-based controllers lies in the fact that the nonlinear structure of the dynamics of the system is not taken into account. Considering this weakness, many researchers have investigated alternative adaptive control architectures inspired from non-adaptive model-based controllers. In [6], an adaptive version of the computed torque scheme has been proposed. A stability analysis based on the Lyapunov theory was conducted to derive the adaptation law that provided both the tracking of the desired trajectories as well as the convergence of the estimated parameters. Nevertheless, in this work, the control law showed two main limitations, namely the need of the measured actual joint accelerations and the boundedness of the inverse of the estimated inertia matrix. These two limitations were bypassed in [7] by online estimating the measured accelerations while the former issue has been solved by following an approach inspired from robust control theory. Similar adaptive approaches can be found in [8, 9].

Recently, a new control scheme called $\mathcal{L}_1$ adaptive control which seeks to revisit some primary issues in MRAC has been proposed [10, 11]. This control approach, being itself inspired from MRAC, includes a low-pass filtering technique that decouples the estimation and control loops. This is of a tremendous importance since increasing the adaptation gain in conventional adaptive control may hurt the robustness of the closed-loop system. Indeed, control signals in MRAC may have undesired high frequencies lying outside the control channel bandwidth. The low-pass filter in $\mathcal{L}_1$ adaptive control, which is designed by using the $\mathcal{L}_1$ small gain theorem, is consequently introduced to remove these undesired frequencies. The first validations of $\mathcal{L}_1$ adaptive control were through numerical simulations and real-time experiments mainly in aerial vehicles [12]. It was afterwards extended to address more applications such as underwater vehicles [13], mechanical manipulators [14], underactuated systems [15] and parallel manipulators [16]. One major advantage of the $\mathcal{L}_1$ adaptive control is its being model-free; i.e. not requiring any knowledge about the dynamics of the system. However, it would be interesting to include, if available, some knowledge about the dynamics of the system in the control loop in order enhance its tracking performance and improve the overall closed-loop system.

This chapter focuses on the development of a new control scheme based on the $\mathcal{L}_1$ adaptive control. The nominal nonlinear dynamics of the system are included in the control loop in order to improve its performance. Specifically, the proposed controller has a major advantage of improving the tracking performance which is crucial in most robotic applications. In order to demonstrate our claims, real-time experiments are conducted on Veloce; a 4-DOF parallel manipulator developed in our laboratory intended to be used for high-speed pick-and-place applications. The remainder of this chapter is organized as follows. In section 2., we provide a brief state of the art on adaptive controllers proposed in the literature for parallel manipulators. Section 3. is devoted to the modeling and description of Veloce robot. In section 4., the control problem formulation and development of $\mathcal{L}_1$ adaptive control are provided. Section 6. is dedicated to the development of the new proposed control scheme which is based on $\mathcal{L}_1$ adaptive control. Real-time experimental results are presented and discussed in section 7. Finally, conclusions are drawn in section 8.

2. Overview on Adaptive Control of Parallel Manipulators

Adaptive controllers are those control schemes requiring a real-time adjustment (i.e. adaptation) of their parameters in the aim of finding their steady-state values. The development of adaptive strategies is motivated by the abundance of uncertainties in the controlled system and its environment that may deteriorate the control performance. Adaptive controllers can be divided into two categories depending on their requirement of a dynamic model of the robot; non-model-based and model-based strategies.

In what follows, we give a breif overview on the most prominent adaptive control schemes fo parallel manipulators that can be found in the literature.

2.1. Non-model-based adaptive strategies

Strategies belonging to this class do not require the dynamic model of the robot nor its structure to be known. Instead, they consider all the nonlinearities including those of the dynamic model, uncertainties and possible disturbances, whose structure is unknown, as a general disturbance term to be estimated in real-time. Then, the control law is designed such that the effect of this disturbances is minimized.

2.1.1. MRAC-based startegies

The main idea of MRAC is to obtain a closed-loop system with adjustable controller parameters. These parameters have the ability of changing the behavior of the closed-loop system. The time evolution of the adaptive parameters is adjusted by comparing the output

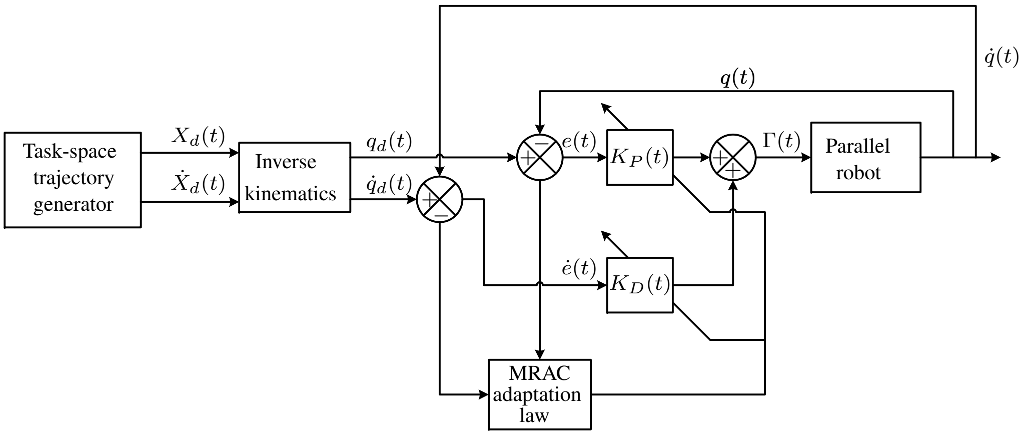

Figure 1. Bloc diagram of the proposed MRAC-based control in [18]

of the controlled system and the desired one obtained from the reference model. The ultimate goal is to make the behavior of the controlled system match a desired reference model despite the eventual variations/uncertainties in the system or its environment [17].

While numerous MRAC-based control schemes for serial manipulators can be found in the literature [19, 20], only few controllers were proposed for parallel ones. In [18], an adaptive control scheme based on MRAC has been proposed to control a 6-DOF parallel manipulator based on the Stewart platform prototype used to emulate space operations. In this work, the control scheme consisted of a joint space PD feedback controller with adjustable gains. The control scheme is further endowed with an adaptation mechanism that controls the evolution of the adaptive feedback gains. The joint tracking error of each axis was supposed to follow a desired user-defined reference second order linear system characterized by its natural frequency and its damping ratio. Based on *Lyapunov* stability analysis, the adaptation law stabilizing the error dynamics is derived. Experiments were conducted on a Stewart-platform parallel manipulator to evaluate the performance of the controller and its robustness towards sudden payload changes. Two case studies were considered; mainly, a vertical motion of the moving platform and a circular one. In both cases the proposed MRAC-based strategy with adaptive gains outperforms the PD with constant feedback gains. Figure 1 illustrates the block diagram summarizing the MRAC-based controller proposed in [18].

2.1.2. Strategies based on artificial neural networks

Artificial Neural Networks (ANN) are known by their powerful universal approximation features [21]. This is why they attracted a big deal of interest in various fields, and have been applied almost everywhere, from identification to estimation and control. Artificial neural networks learn from experience rather than programming, hence, they are typically used in repetitive tasks. Several works can be found in the literature regarding the application of neural networks in control of parallel manipulators [22]. However they remain relatively few compared to what can be found for serial manipulators [23].

Thanks to their learning ability, artificial neural networks are mostly used to approximate the dynamics of the manipulator. Then, the learned dynamics can be included in the control scheme to compensate for the uncertainties and disturbances. [24] proposed to augment a decentralized Cartesian space PID controller with an artificial neural network term. The main motivation of this work is to improve the tracking capabilities of a 2-DOF redundantly actuated parallel manipulator. The provided simulation results demonstrated the superiority of the augmented controller with respect to the original PID. The maximum errors were significantly reduced since the additional neural network term accounted for the nonlinear dynamics of the manipulator.

2.2. Model-based adaptive strategies

Model-based adaptive controllers explicitly include the dynamics of the robot in the control loop. In contrast to their non-adaptive counterparts, the adaptive schemes adjust the parameters of the model-based loop in order to converge them their best steady-space values. Consequently, if the dynamics of the manipulator is uncertain or time-varying, adaptive controllers result in a better closed-loop performance.

2.2.1. Strategies based on computed torque

Computed-torque-based adaptive schemes are those controllers relying on the classical non-adaptive computed torque controller. The control design starts by considering the ideal non-adaptive controller that assumes perfect cancellation of the system nonlinearities. Due to possible uncertainties and variations in the system dynamics, the compensation of the nonlinearities of the system could not be as perfect as the non-adaptive controller assumes. Then, an additional adaptive estimation loop is considered to account for these uncertainties. The role of the adaptive loop is to estimate, in real-time, the system's parameters and the obtained estimation are used in the controller.

The typical example of a computed-torque-based adaptive controller is the one proposed in [25]. The first step of the control design in this work is to consider a computed torque control law with uncertain parameters. Then, the error equation resulting from the application of this control law is obtained. Based on a thorough stability analysis using the *Lyapunov* theory, an adaptive law for the parameters' estimation is derived. The proposed

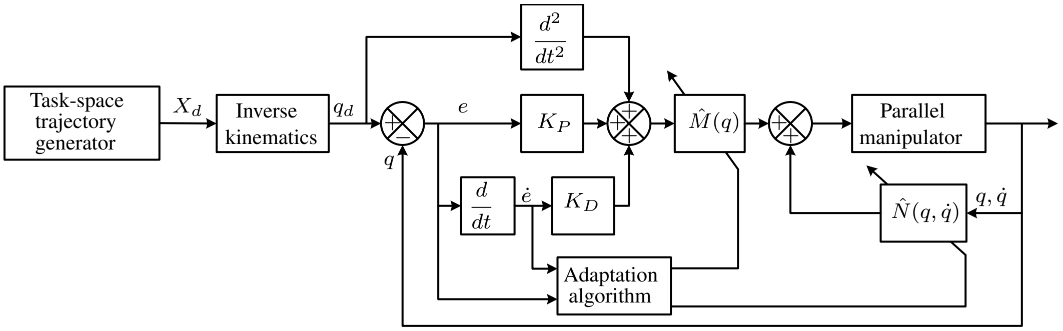

Figure 2. Bloc diagram of adaptive computed torque control

adaptive controller in conjunction with the adaptation law guarantee that the tracking errors vanish and that the estimated parameters converge to their best steady-state values. The block diagram depicted in Figure 2 clarifies the principle of such control strategy.

The computed-torque-based adaptive controller, proposed in [25], was mainly developed for serial manipulators. However, it is known that parallel manipulators share many properties with their serial counterparts [26, 27]. Based on this fact, the controller in [25] has been straightforwardly applied to PKMs in [28]. In this work, an adaptive Cartesian space computed-torque-based scheme is applied to a 2-DOF redundantly actuated parallel manipulator. Real-time experiments were carried out in order to highlight the benefits of the adaptive controller compared to its non-adaptive version. The adaptive controller shows a net superiority and allows to estimate the system dynamic parameters.

A main drawback of computed torque based adaptive controllers is their dependence on the real acceleration of the robot [25]. This shortcoming is crucial from a practical point of view since measuring actual accelerations is tedious.

2.2.2. Based on passivity of the system

Passivity-based controllers use the passivity property of the manipulator [7]. In contrast with computed-torque-based ones, passivity-based controllers do not assume perfect cancellation of the system nonlinearities, even in the ideal case of perfect knowledge of the manipulator's parameters. Hence, they do not lead to a closed-loop linear error system. However, based on the passivity property of the system, they result in a stable closed-loop system.

[8] proposed a passivity-based adaptive controller that holds many advantageous features. This controller can be assimilated to an adaptive version of PD control with computed feedforward. Indeed, the proposed control law consists mainly in a PD feedback loop in addition to an adaptive feedforward term based on the dynamics of the robot and the desired reference trajectories. The main advantage of this scheme is that it does not require the joint acceleration measurement. Moreover, the compensation of the system's nonlinearities is based on desired quantities that can be computed offline. This means that the control scheme does not require heavy computations. Furthermore, the use of the desired values instead of the measured ones, which can be often noisy, may enhance the robustness of the controller toward measurement noise.

In [29], the developed controller in [8] has been embraced for joint space control of a 6-DOF parallel manipulator called the Hexaglide. The proposed adaptive controller as well as a linear PD controller were implemented in real-time for a comparison purpose. It was not surprising that the adaptive controller provided better tracking results since it compensates for the nonlinear dynamics of the manipulator.

In [30], a Cartesian space control strategy called *Dual-Space Adaptive Control* is proposed to control redundantly actuated parallel manipulators. The proposed controller in this work share many similarities with the proposed control strategy in [29]. The main difference is that the weighting gain of the position error with respect to the velocity error is time-varying. The proposed controller was experimentally validated on the R4 parallel manipulator [31]. Real-time experiments showed that the proposed Dual-Space adaptive controller significantly improves the tracking performance and allows the tracking of extremely fast trajectories (up to 100G of maximum acceleration).

Sadegh [32] proposed to extend the joint-space passivity based adaptive controller of [33], mainly developed for serial manipulators, to Cartesian-space control of parallel manipulators with actuation redundancy. Moreover, to deal with friction an additional control term has been added to the conventional feedback and adaptive loops. Real-time experiments, conducted on a 2-DOF redundantly actuated parallel manipulator, showed the superiority of the proposed adaptive controller compared to the augmented PD controller in terms of Cartesian tracking errors, at both low and high accelerations. Unfortunately, the evolution of the estimated parameters was not shown in this work to see if they really converge to their steady-state values.

3. Modeling of Delta-like Parallel Robots: Application to Veloce

Veloce is a 4-DOF Delta-like [34] fully actuated parallel manipulator having four identical kinematic chains [35]. Each kinematic chain can be seen as serial arrangement of the actuator, a rear-arm and a forearm. Its articulated moving platform illustrated in Figure 3, which is connected to the fixed-base through the kinematic chains, can perform three spatial translations and one rotation around the vertical axis. The rotation is obtained via the relative motion of the upper and lower parts of the moving platform. The four actuators responsible for the movement of the mechanical structure are all lying on the same plane. The CAD of Veloce is illustrated in Figure 4.

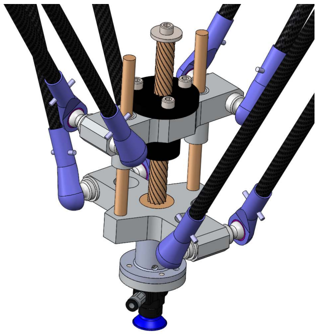

Figure 3. Articulated moving platform of Veloce

3.1. Inverse kinematics model

The inverse kinematics problem consists in finding the actuated joint vector $q = [q_1, q_2, q_3, q_4]^T \in \mathbb{R}^4$ corresponding to a specific pose $X = [x, y, z, s]^T$ of the moving platform of Veloce. Figures 5 and 6 in addition to Table 1 summarize the various geometric parameters involved in the establishment of the inverse kinematic model of Veloce. The actuated joints locations are represented by points $A_i,\ i = 1, \ldots, 4$, whose coordinates are given in the fixed Cartesian frame $O - xyz$ by:

$$A_i = r_b \left[\cos(\alpha_i),\ \sin(\alpha_i),\ 0\right]^T , \tag{1}$$

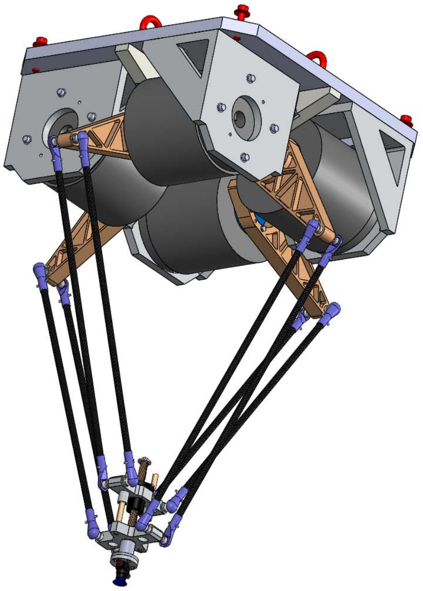

Figure 4. CAD view of Veloce 4-DOF parallel manipulator

where $\alpha_i = (i-1)\pi/2,\ i = 1, \dots, 4$, is the orientation of the ith actuator axis with respect to the fixed Cartesian frame. The locations of the centers of passive ball joints are

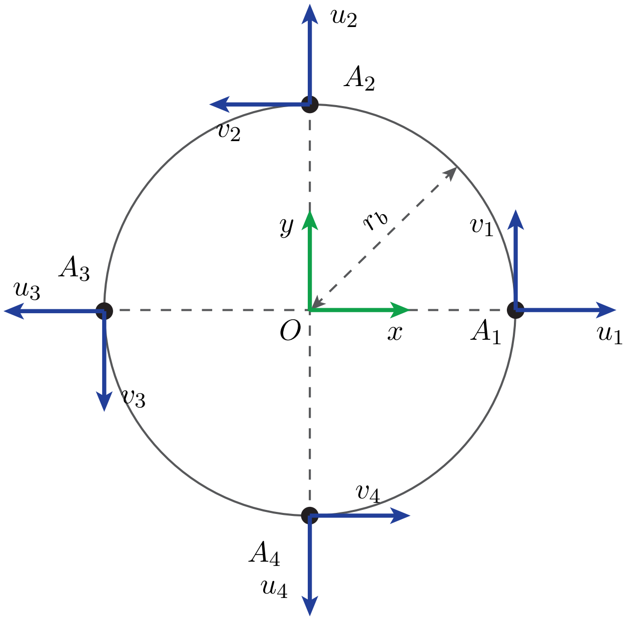

Figure 5. Geometric parameters of Veloce robot: top view

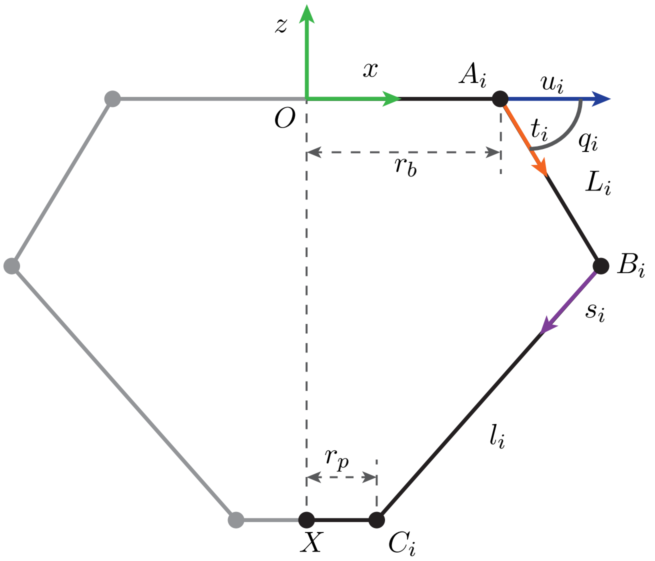

Figure 6. Geometric parameters of Veloce robot: side view

represented by points B_i and C_i which are expressed in the base frame $O - xyz$ by:

$$B_i = A_i + L\left[\cos(\alpha_i)\cos(q_i),\ \sin(\alpha_i)\cos(q_i), \sin(q_i)\right], \tag{2}$$

$$C_i = \left[r_p\cos(\alpha_i) + x,\ r_p\sin(\alpha_i) + y,\ z\right]. \tag{3}$$

To facilitate the subsequent mathematical development, a frame $P_i - u_i v_i z$ is attached to each actuator, where the vectors u_i and v_i are given by:

$$u_i = \left[\cos(\alpha_i),\ \sin(\alpha_i),\ 0\right]^T, \tag{4}$$

$$v_i = \left[-\sin(\alpha_i),\ \cos(\alpha_i),\ 0\right]^T, \tag{5}$$

where v_i is always directed towards the ith actuator's axis. The inverse kinematics solution of Veloce robot can be obtained by finding the intersection of a circle and a sphere which represents the coordinates of the passive joints $B_i = \left[x_{B_i},\ 0,\ z_{B_i}\right]^T$ in the corresponding $A_i - u_i v_i z$ frame [36, 37]. Notice that in this frame, the B_i points always satisfy $y_{B_i} = 0$.

Table 1. Geometric parameters of Veloce robot

Parameter	Description	Value
L_i	Rear-arm length	200 mm
l_i	Forearm length	530 mm
r_b	Base radius	135 mm
r_p	Moving Platform radius	48 mm

From the motion of one rear-arm, we can write the equation of a circle in the corresponding $A_i - u_i z$ plane of center $A_i = [0,\ 0]$ and of radius L as follows:

$$x_{B_i}^2 + z_{B_i}^2 = L^2 \tag{6}$$

The movements of one forearm can be described by a sphere of center C_i and radius l. In the $A_i - u_i v_i z$ frame, we can write the following equation:

$$(x_{B_i} - x_{C_i})^2 + y_{C_i}^2 + (z_{B_i} - z_{C_i})^2 = l^2 \tag{7}$$

Subtracting (6) from (7) yields:

$$2x_{C_i}x_{B_i} + 2z_{C_i}z_{B_i} = l^2 - L^2 + x_{C_i}^2 + y_{C_i}^2 + z_{C_i}^2 \tag{8}$$

Solving (8) for z_{B_i} yields:

$$z_{B_i} = \frac{S_i - 2x_{C_i}}{2z_{C_i}}, \tag{9}$$

with $S_i \triangleq l^2 - L^2 + x_{C_i}^2 + y_{C_i}^2 + z_{C_i}^2$. Substituting (9) in (6) yields:

$$4\left(z_{C_i}^2 + x_{C_i}^2\right)x_{B_i}^2 - 4S_i x_{C_i} x_{B_i} + (S_i^2 - 4z_{C_i}^2 L^2) = 0 \tag{10}$$

Solving for x_{B_i} results in:

$$x_{B_i} = \frac{S_i x_{C_i} \pm \sqrt{(S_i x_{C_i})^2 - \left(z_{C_i}^2 + x_{C_i}^2\right)(S_i^2 - 4z_{C_i}^2 L^2)}}{2\left(z_{C_i}^2 + x_{C_i}^2\right)} \tag{11}$$

Equations (11) and (9) result in the coordinates of the intersection of a circle and a sphere. For geometrical reasons, only the largest x_{B_i} is kept. Once the coordinates of the points B_i are found, the corresponding actuated joint value can be obtained by:

$$q_i = \text{atan2}\left(z_{B_i}, x_{B_i}\right) \tag{12}$$

Applying (12) to each of the four kinematic chains results in the solution of the inverse kinematic model of Veloce.

3.2. Jacobian matrix

The Jacobian matrix $J(q, X) \in \mathbb{R}^{4\times 4}$ of Veloce relates its moving platform's velocity vector $\dot{X} = [\dot{x}, \dot{y}, \dot{z}, \dot{s}]$ to the actuated joints velocity vector $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4]$ such that:

$$\dot{X} = J\dot{q} \tag{13}$$

Note that q and X are related through the inverse kinematic model. The Jacobian matrix J can be obtained based on the following kinematic relationship:

$$\|B_iC_i\|^2 - l^2 = 0, \tag{14}$$

which means that the length of each forearm remains constant independently of the robot's configuration. Differentiating (14) and rearranging the terms results in:

$$J_x \dot{X} = J_q \dot{q}, \tag{15}$$

where J_q and J_x are given as follows:

$$J_q = \text{diag}\left(l_1 w_1^T s_1, \ldots, l_n w_n^T s_n\right), \tag{16}$$

$$J_x = [s_1 \ \ldots \ s_n]^T, \tag{17}$$

where

$$w_i = t_i \times v_i \tag{18}$$

$$t_i = \frac{A_i B_i}{L_i} \tag{19}$$

$$s_i = \frac{B_i C_i}{l_i} \tag{20}$$

Finally, the Jacobian matrix can be written as follows:

$$J = \begin{bmatrix} \frac{l_1 w_1^T s_1}{s_1} & \ldots & \frac{l_i w_i^T s_i}{s_i} \end{bmatrix} \tag{21}$$

3.3. Inverse dynamic model

In this section, a brief description of the simplified dynamic model of Veloce is presented in the sequel. But first, to simplify the motion equations of the different parts of the mechanical structure of Veloce, the following assumptions, commonly used in Delta-like robots, are considered [37]

Assumption 1. *Both dry and viscous frictions in all passive and active joints are neglected. This is mainly due to the fact that the joints are carefully designed such that friction effects are minimized.*

Assumption 2. *The rotational inertia of the forearms is neglected and their mass is split up into two equivalent parts, one is added to the mass of the arm while the other one is considered with the moving platform. This hypothesis is justified by the small mass of the forearms compared to other components.*

Table 2. Dynamic parameters of Veloce robot

Parameter	Description	Value
m_a	One Rear-arm's weight	0.541 kg
m_f	One Forearm's weight	0.08 kg
m_p	Platform's weight	0.999 kg

The dynamics of the Veloce robot shows a lot of similarities with those of the Delta robot. Nevertheless few differences arise due to the number of kinematic chains and the additional rotational DOF of the moving platform.

Regarding the moving platform, we distinguish two kinds of forces acting on it, the gravity forces $G_p \in \mathbb{R}^4$ and the inertial force $F_p \in \mathbb{R}^4$, they are given by

$$G_p = M_p \, [0\;0\; -g\;0]^T \tag{22}$$

$$F_p = M_p \ddot{X} \tag{23}$$

where $M_p \in \mathbb{R}^{4\times4}$ is the mass matrix of the moving platform that also considers the half-masses of the forearms, $g = 9.81$ m/s^2 is the gravity constant and $\ddot{X} \in \mathbb{R}^4$ is the Cartesian acceleration vector.

The contributions of G_p and F_p to each motor can be computed using the Jacobian matrix $J(q, X) \in \mathbb{R}^{4\times4}$ as follows

$$\Gamma_{G_p} = J^T M_p \, [0\;0\; -g\;0]^T \tag{24}$$

$$\Gamma_p = J^T M_p \ddot{X} \tag{25}$$

From the joints side, the elements that contribute to the dynamics of the actuators are the forces and torques resulting from the movement of the rear-arms in addition to the half-messes of the forearms.

Applying the virtual work principle, which states that the sum of non-inertial forces is equal to that of the inertial ones, and after rearranging the terms, we get

$$J^T M_p \ddot{X} + \Gamma_{G_p} + I_a \ddot{q} + \Gamma_{G_a} = \Gamma \tag{26}$$

being $I_a \in \mathbb{R}^{4\times4}$ a diagonal inertia matrix of the arms accounting for the rear-arms as well as the half-masses of the forearms, $\ddot{q} \in \mathbb{R}^4$ the joint acceleration vector and $\Gamma_{G_a} \in \mathbb{R}^4$ is the force vector resulting from gravity acting on the arms being given by

$$\Gamma_{G_a} = m_a r_{G_a} g cos(q) \tag{27}$$

with m_a the sum of the mass of one rear-arm and one half-mass of a forearm, r_{G_a} the distance between the center of one axis and the center of mass of one arm and q_1, q_2, q_3 and q_4 are the joints positions.

Given the kinematic relationship $\ddot{X} = J\ddot{q} + \dot{J}\dot{q}$, (26) can be rewritten as follows

$$\left(I_a + J^T M_p J\right)\ddot{q} + \left(J^T M_p \dot{J}\right)\dot{q} - \left(J^T G_p + \Gamma_{G_a}\right) + \Gamma_d = \Gamma, \tag{28}$$

where Γ_d is a disturbance term that accounts for non-modeled dynamics and possible external disturbances. It can be written in a standard joint-space form as follows

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \Gamma_d = \Gamma, \tag{29}$$

with:

$M(q) = I_a + J^T M_p J$ the total mass matrix.

$C(q, \dot{q}) = J^T M_p \dot{J}$ the Coriolis and centrifugal forces matrix.

$G(q) = -\left(J^T G_p + \Gamma_{G_a}\right)$ the gravitational forces vector.

The expression of the dynamics in (29) is suitable for joint-space control since it is expressed in terms of the actuated joints.

4. Background on $\mathcal{L}_1$ Adaptive Control

In this section, the basic idea behind the development of the $\mathcal{L}_1$ adaptive control theory is detailed. For the sake of simplicity, we consider a single-input single-output system. Since $\mathcal{L}_1$ adaptive control is inspired from MRAC, we first introduce MRAC for the considered system. Then the architecture of MRAC is changed to achieve the decoupling of robustness and adaptation.

4.1. Control problem formulation

Consider the following single-input single-output linear time-invariant system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + bu(t), \quad x(0) = x_0 \\ y(t) &= c^T x(t) \end{aligned} \tag{30}$$

where

$x(t) \in \mathbb{R}^n$ is the measurable state of the system,

$u(t) \in \mathbb{R}$ is the control input,

$b, c \in \mathbb{R}^n$ are known constant input and output vectors,

$y(t) \in \mathbb{R}$ is the output to be regulated by the controller.

The system state matrix $A \in \mathbb{R}^{n\times n}$ is supposed unknown. The control objective is to design an adaptive control signal $u(t)$ such that the regulated output of the system $y(t)$ tracks a given reference signal $r(t) \in \mathbb{R}$ with desired performance while guaranteeing that the system state and other signals remain bounded under the following assumption

Assumption 3. *There exist a Hurwitz matrix $A_m \in \mathbb{R}^{n\times n}$ and a vector $\theta \in \mathbb{R}^n$ of perfect parameters such that the pair (A_m, b) is controllable and $A_m - A = b\theta^T$. Moreover, assume that this unknown perfect parameters vector θ belongs to a given compact set Θ.*

Under the above assumption, the system in (30) can be rewritten as follows

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + b\left(u(t) - \theta^T x(t)\right), \quad x(0) = x_0 \\ y(t) &= c^T x(t) \end{aligned} \tag{31}$$

4.2. From direct MRAC to direct MRAC with a state predictor

One of the key differences between MRAC and $\mathcal{L}_1$ adaptive control is the introduction of a prediction-based adaptive architecture. In the following, this new structure is highlighted and demonstrated that it leads to the same closed-loop behavior as direct MRAC.

4.2.1. Direct MRAC

Consider the nominal nonadaptive controller given by

$$u_{nom}(t) = \theta^T x(t) + k_g r(t) \tag{32}$$

where $k_g \triangleq 1/\left(c^T A_m^{-1} b\right)$ is a feedforward gain that ensures zero steady-state tracking error. Substituting (32) in the system dynamics in (31) leads to the following desired reference system

$$\begin{aligned} \dot{x}_m(t) &= A_m x_m(t) + b k_g r(t), \quad x(0) = x_0 \\ y_m(t) &= c^T x_m(t) \end{aligned} \tag{33}$$

where $x_m(t)$ is the state of the reference system and $y_m(t)$ its output. The nominal control law is impossible to implement due to the uncertainties in the system state matrix A (and, hence, θ) which should be estimated online.

The feasible direct MRAC control law is hence given by

$$u(t) = \hat{\theta}^T x(t) + k_g r(t) \tag{34}$$

where $\hat{\theta} \in \mathbb{R}^n$ is an estimate of θ whose evolution is governed by the following adaptation rule

$$\dot{\hat{\theta}}(t) = \gamma x(t) e^T(t) P b, \quad \hat{\theta}(0) = k_{x0} \tag{35}$$

where $\gamma \in \mathbb{R}^+$ is the adaptation gain, $e(t) \triangleq x_m(t) - x(t)$ and $P = P^T > 0$ is the solution for the algebraic Lyapunov equation $A_m^T P + P A_m = -Q$, for an arbitrary choice of $Q = Q^T > 0$.

Substituting the adaptive control law (34) in the system dynamics in (31) leads to the following error dynamics

$$\dot{e}(t) = A_m e(t) - b\tilde{\theta}^T(t) x(t) \tag{36}$$

where $\tilde{\theta} \triangleq \hat{\theta}(t) - \theta$ is the parameters estimation error. Figure 7 depicts the overall block diagram of MRAC.

4.2.2. MRAC with a state predictor

Now consider the following state predictor that mimics the behavior of the system in (31) with the unknown parameter θ replaced by its estimate $\hat{\theta}(t)$

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m \hat{x}(t) + b(u(t) - \hat{\theta} x(t)), \quad \hat{x}(0) = x_0 \\ \hat{y}(t) &= c^T \hat{x}(t) \end{aligned} \tag{37}$$

where $\hat{x}(t)$ being the state of the predictor. The prediction error dynamics can be obtained by subtracting the system dynamics from those of the predictor as follows

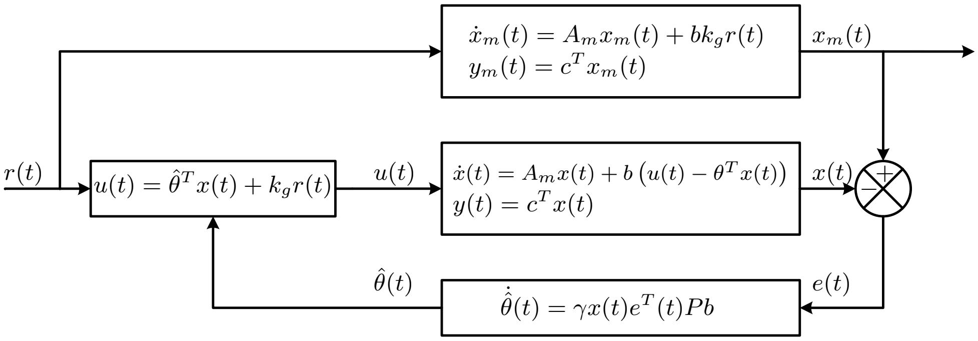
$\dot{x}_m(t) = A_m x_m(t) + bk_g r(t)$
$y_m(t) = c^T x_m(t)$
$x_m(t)$
$r(t)$
$u(t) = \hat{\theta}^T x(t) + k_g r(t)$
$u(t)$
$\dot{x}(t) = A_m x(t) + b\left(u(t) - \theta^T x(t)\right)$
$y(t) = c^T x(t)$
$x(t)$
$+$
$-$
$\hat{\theta}(t)$
$\dot{\hat{\theta}}(t) = \gamma x(t) e^T(t) Pb$
$e(t)$

Figure 7. Block diagram of MRAC

$$\dot{\tilde{x}}(t) = A_m \tilde{x}(t) - b\tilde{\theta}(t)x(t) \tag{38}$$

where $\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$ is the prediction error. It can be seen that the prediction error dynamics is identical to the obtained error dynamics with direct MRAC in (36). The adaptive law of the predictor-based controller is similar to that of direct MRAC, the only difference is the use of the prediction error $\tilde{x}(t)$ instead of the tracking error $e(t)$. It is thus given by

$$\dot{\hat{\theta}}(t) = \gamma x(t)\tilde{x}^T(t)Pb, \quad \hat{\theta}(0) = \theta_0 \tag{39}$$

It can be noticed that the closed-loop state predictor mimics the behavior of the reference model in (33). Indeed, substituting the control law (34) in the predictor dynamics and upon the use of (39) we get the following closed-loop dynamics

$$\begin{aligned} &\dot{\hat{x}}(t) = A_m \hat{x}(t) + bk_g r(t), \quad \hat{x}(0) = x0, \\ &\hat{y}(t) = c^T \hat{x}(t) \end{aligned} \tag{40}$$

which is identical to the dynamics of the reference system of direct MRAC in (33). It is demonstrated in [38] that the tracking (or prediction) errors are upper bounded at any time by

$$\|e(t)\|(= \|\tilde{x}(t)\|) \leq \frac{\|\tilde{\theta}(0)\|}{\sqrt{\lambda_{min}(P)\gamma}} \tag{41}$$

being $\tilde{\theta}(0)$ the initial parameters estimation error and $\lambda_m in(P)$ the minimum eigenvalue of P. This means that the tracking error can be made arbitrarily small by increasing the adaptation gain γ. However, it can be seen from the control law in (34) and the adaptation laws in (35) and (39) that increasing the adaptation gain lead to high gain feedback. The block diagram of MRAC scheme with a state predictor is shown in Figure 8.

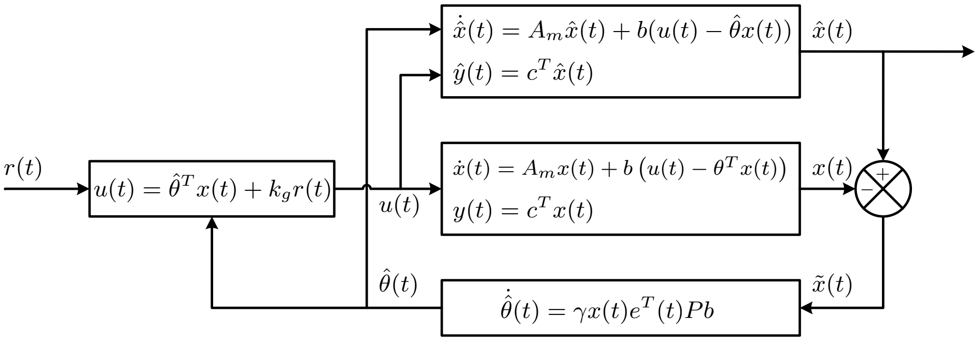
$\dot{\hat{x}}(t) = A_m\hat{x}(t) + b(u(t) - \hat{\theta}x(t))$
$\hat{y}(t) = c^T\hat{x}(t)$
$\hat{x}(t)$
$r(t)$
$u(t) = \hat{\theta}^T x(t) + k_g r(t)$
$u(t)$
$\dot{x}(t) = A_m x(t) + b\left(u(t) - \theta^T x(t)\right)$
$y(t) = c^T x(t)$
$x(t)$
+
−
$\hat{\theta}(t)$
$\dot{\hat{\theta}}(t) = \gamma x(t)e^T(t)Pb$
$\tilde{x}(t)$

Figure 8. Block diagram of the control loop based on MRAC with state predictor

4.3. $\mathcal{L}_1$ adaptive control

In a real system, the unknown parameters vector θ may be a time-varying uncertainty with frequencies that may lie outside the control channel bandwidth [38]. In this case, such frequencies are naturally attenuated due to hardware limitations of the actuators. Consequently, the behavior of the closed-loop system will be different than the dynamics of the reference model in (33).

The control problem in $\mathcal{L}_1$ adaptive control is formulated with the understanding that some uncertainties could never be perfectly compensated. Indeed, while the control objective in MRAC is only stated asymptotically, the closed-loop performance in $\mathcal{L}_1$ adaptive control is specified $\forall t \geq 0$. This means that at any time, the performance of the system can be predicted. Moreover, the control signal never exceeds the available control channel bandwidth.

For the class of systems given by (31), consider the following state predictor that mimics its behavior

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m\hat{x}(t) + b(u(t) - \hat{\theta}x(t)), \quad \hat{x}(0) = x_0 \\ \hat{y}(t) &= c^T\hat{x}(t) \end{aligned} \tag{42}$$

In addition to the state predictor in (42), consider a projection-type adaption law for the estimated parameter vector $\hat{\theta}$ expressed as

$$\dot{\hat{\theta}}(t) = \gamma \operatorname{Proj}\left(\hat{\theta}(t), x(t)\tilde{x}(t)Pb\right), \quad \hat{\theta}(0) = \theta_0 \tag{43}$$

which is adjusted using the prediction error $\tilde{x}(t) = \hat{x}(t) - x(t)$. The projection operator Proj avoids the parameters drift and ensures that they remain inside the compact set Θ. In (43), γ is the positive adaptation gain and P is the symmetric positive definite matrix, solution of the algebraic *Lyapunov* equation $A_m^T P + PA_m = -Q$ for an arbitrary symmetric positive definite matrix Q.

The last stage which is one of the notable unique features of the $\mathcal{L}_1$ adaptive control, is the control input characterized by the introduction of a low-pass filter. It is given by its *Laplace* form as follows

$$u(s) = C(s)\left(\hat{\eta}(s) + k_g r(s)\right) \tag{44}$$

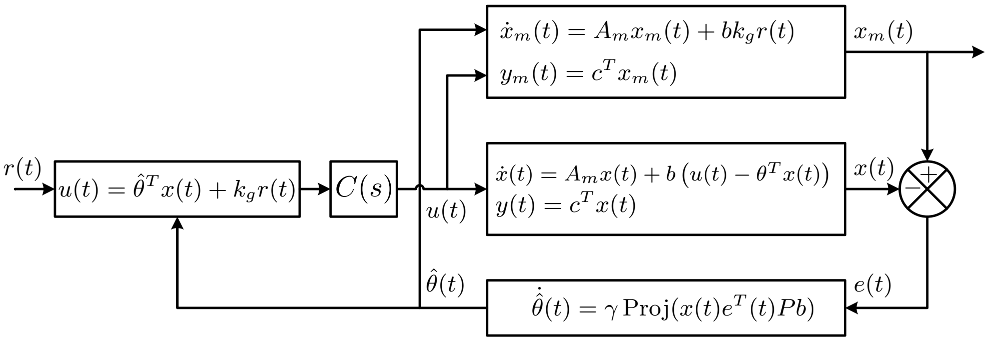
$\dot{x}_m(t) = A_m x_m(t) + bk_g r(t)$
$y_m(t) = c^T x_m(t)$
$x_m(t)$
$r(t)$
$u(t) = \hat{\theta}^T x(t) + k_g r(t)$
$C(s)$
$u(t)$
$\dot{x}(t) = A_m x(t) + b\left(u(t) - \theta^T x(t)\right)$
$y(t) = c^T x(t)$
$x(t)$
+
−
$\hat{\theta}(t)$
$\dot{\hat{\theta}}(t) = \gamma\,\mathrm{Proj}(x(t)e^T(t)Pb)$
$e(t)$

Figure 9. Block diagram of $\mathcal{L}_1$ adaptive control.

where $C(s)$ is a bounded-input bounded-output strictly proper transfer function with $C(0) = 1$, $r(s)$ is the *Laplace* transform of the reference trajectory $r(t)$ and $\hat{\eta}(s)$ is the *Laplace* transform of $\eta\hat{(t)} = \hat{\theta}^T(t)x(t)$.

It is demonstrated in [38] that the controlled system in (30) under the $\mathcal{L}_1$ adaptive control law in (44) is bounded-input bounded-state with respect to $r(t)$ and x_0 if the following $\mathcal{L}_1$-norm-based equality is satisfied

$$\|G(s)\|_{\mathcal{L}_1} L < 1 \tag{45}$$

where

$$G(s) \triangleq H(s)(1 - C(s)), \quad H(s) \triangleq (s\mathbb{I} - A_m)^{-1}b, \quad L \triangleq \max_{\theta\in\Theta} \|\theta\|_1 \tag{46}$$

The block diagram of the $\mathcal{L}_1$ adaptive control strategy for the class of systems given by (31) is displayed in Figure 9.

5. Application of $\mathcal{L}_1$ adaptive control to Parallel Manipulators

Recall the inverse dynamics for parallel mechanical manipulators which are expressed as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Gamma_d(t) = \Gamma(t) \tag{47}$$

In the aim of developing a $\mathcal{L}_1$ adaptive controller for the class of systems given by (47), introduce the following combined position-velocity tracking error given by

$$r = (\dot{q} - \dot{q}_d) + \Lambda\,(q - q_d) \tag{48}$$

where $\Lambda \in \mathbb{R}^{4\times 4}$ is a symmetric positive-definite weighting matrix.

5.1. Control law

Consider the following control input vector $\Gamma(t)$ consisting of a combination of two distinct terms [39]

$$\Gamma(t) = \Gamma_m(t) + \Gamma_{ad}(t), \quad \Gamma_m(t) \triangleq A_m r(t) \tag{49}$$

where $\Gamma_m \in \mathbb{R}^n$ is a state-feedback term characterizing the desired transient response of the tracking error $r(t)$ and Γ_{ad} is the adaptive control term which will be designed subsequently. Taking the first time derivative of (48) with respect to time we get

$$\dot{r}(t) = (\ddot{q} - \ddot{q}_d) + \Lambda\,(\dot{q} - \dot{q}_d) \tag{50}$$

Substituting the control law (49) into the dynamics of the parallel manipulator in (47) and solving for $\ddot{q}$ along with substituting into (50), we get the following error dynamics

$$\dot{r}(t) = A_m r(t) + \Gamma_{ad}(t) - \eta(t, \zeta(t)), \quad r(0) = r_0 \tag{51}$$

where $\zeta = [r, q]^T$ and $\eta(t, \zeta(t))$ is a nonlinear function that gathers all the nonlinearities of the system including possible external disturbances. Before proceeding to the development of the adaptive control law, some important assumptions and properties regarding the unknown function $\eta(t, \zeta(t))$ have to be made, they can be summarized as follows

Assumption 4. ***(Uniform boundedness of*** $\eta(\mathbf{t}, \mathbf{0})$***)*** *There exist $B > 0$ such that $\forall t \geq 0$, $\|\eta(t, 0)\| \leq B$.*

Assumption 5. ***(Semiglobal uniform boundedness of partial derivatives of*** $\eta(\mathbf{t}, \zeta(\mathbf{t}))$***)*** *The unknown nonlinear function $\eta(t, \zeta(t))$ is continuous with respect to its arguments, and for arbitrary $\delta > 0$, there exist $d_{\eta_t}(\delta)$ and $d_{\eta_x}(\delta)$ such that*

$$\left\| \frac{\partial \eta(t, \zeta)}{\partial t} \right\|_{\infty} \leq d_{\eta_t}(\delta), \quad \left\| \frac{\partial \eta(t, \zeta)}{\partial \zeta} \right\|_{\infty} \leq d_{\eta_\zeta}(\delta) \tag{52}$$

Assumption 6. *for $t \geq 0$, $\|r_t\|_{\mathcal{L}_\infty} \leq \rho$ and $\|\dot{r}_T\|_{\mathcal{L}_\infty} \leq d_r$, for some positive constants ρ and d_r.*

It follows from Lemma A.9.2 in [38] that the unknown nonlinear function $\eta(t, \zeta(t))$ can be parametrized as follows

$$\eta(t, \zeta(t)) = \theta(t)\|r_t\|_{\mathcal{L}_\infty} + \sigma(t) \tag{53}$$

where $\theta(t), \sigma(t) \in \mathbb{R}^n$ are continuous, piecewise-differentiable and uniformly bounded unknown functions. Therefore, the error dynamics in (51) can be rewritten as:

$$\dot{r}(t) = A_m r(t) + \Gamma_{ad}(t) - \left(\theta(t)\|r_t\|_{\mathcal{L}_\infty} + \sigma(t)\right), \quad r(0) = r_0. \tag{54}$$

Notice that the substitution of $\Gamma_{ad}(t) = (\theta(t)\|r_t\|_{\mathcal{L}_\infty} + \sigma(t))$ in the error dynamics in (54) results in the desired performance characterized by $\dot{r}(t) = A_m r(t)$. However, since the nonlinear function $\eta(t, \zeta(t)) = (\theta(t)\|r_t\|_{\mathcal{L}_\infty} + \sigma(t))$ is unknown due to uncertainties and unmeasured disturbances, the control term $\Gamma_{ad}(t)$ should be designed in a way that it estimates the unknown functions $\theta(t)$ and $\sigma(t)$ in real-time.

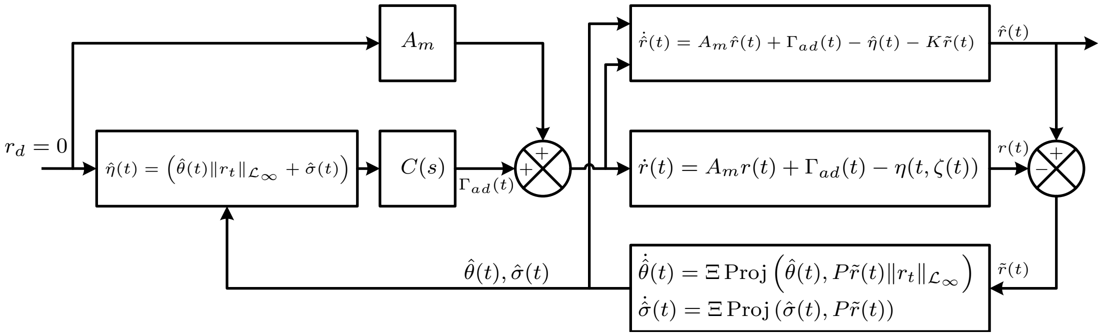
A_m
$\dot{\hat{r}}(t) = A_m\hat{r}(t) + \Gamma_{ad}(t) - \hat{\eta}(t) - K\tilde{r}(t)$
$\hat{r}(t)$
$r_d = 0$
$\hat{\eta}(t) = \left(\hat{\theta}(t)\|r_t\|_{\mathcal{L}_\infty} + \hat{\sigma}(t)\right)$
$C(s)$
$\Gamma_{ad}(t)$
$+$
$+$
$\dot{r}(t) = A_m r(t) + \Gamma_{ad}(t) - \eta(t, \zeta(t))$
$r(t)$
$+$
$-$
$\hat{\theta}(t), \hat{\sigma}(t)$
$\dot{\hat{\theta}}(t) = \Xi\,\mathrm{Proj}\left(\hat{\theta}(t), P\tilde{r}(t)\|r_t\|_{\mathcal{L}_\infty}\right)$
$\dot{\hat{\sigma}}(t) = \Xi\,\mathrm{Proj}\left(\hat{\sigma}(t), P\tilde{r}(t)\right)$
$\tilde{r}(t)$

Figure 10. Block diagram of $\mathcal{L}_1$ adaptive control for parallel manipulators

5.2. State predictor

To predict the behavior of the combined tracking error $r(t)$ and according to the $\mathcal{L}_1$ adaptive control theory, a state predictor that mimics the behavior of the error dynamics in (54) is formulated by [39]:

$$\dot{\hat{r}}(t) = A_m\hat{r}(t) + \Gamma_{ad}(t) - \left(\hat{\theta}(t)\|r_t\|_{\mathcal{L}_\infty} + \hat{\sigma}(t)\right) - Z\tilde{r}(t), \quad \hat{r}(0) = r_0 \tag{55}$$

where $\hat{r}(t)$ is the state of the predictor, $\tilde{r}(t) \triangleq \hat{r}(t) - r(t)$ is the prediction error and $\hat{\theta}(t), \hat{\sigma}(t)$ are estimates of $\theta(t)$ and $\sigma(t)$, respectively. Notice the introduction of the additional term $Z\tilde{r}(t)$ responsible of rejecting the estimation error.

5.3. Adaptation laws

The evolution of the adaptive estimates $\hat{\theta}(t), \hat{\sigma}(t)$, required for the $\mathcal{L}_1$ adaptive control architecture, is governed by the following projection-type adaptation laws

$$\dot{\hat{\theta}}(t) = \Xi\,\mathrm{Proj}\left(\hat{\theta}(t), P\tilde{r}(t)\|r_t\|_{\mathcal{L}_\infty}\right), \qquad \hat{\theta}(0) = \hat{\theta}_0 \tag{56}$$

$$\dot{\hat{\sigma}}(t) = \Xi\,\mathrm{Proj}\left(\hat{\sigma}(t), P\tilde{r}(t)\right), \qquad \hat{\sigma}(0) = \hat{\sigma}_0 \tag{57}$$

where $\Xi > 0$ is the adaptive gain, $P = P^T > 0$ is the solution to the algebraic *Lyapunov* equation $A_m^T P + PA_m = -Q$ for some arbitrary matrix $Q = Q^T > 0$.

Finally, the adaptive control term $\Gamma_{ad}(t)$ in (49) is given by its *Laplace* form as follows

$$\Gamma_{ad}(s) = C(s)\hat{\eta}(s) \tag{58}$$

where $\hat{\eta}(s)$ is the Laplace transform of $\hat{\eta}(t) = \left(\hat{\theta}(t)\|r_t\|_{\mathcal{L}_\infty} + \hat{\sigma}(t)\right)$ and $C(s)$ is a diagonal matrix whose elements are bounded-input bounded-output stable strictly proper transfer functions satisfying $C(0) = \mathbb{I}_n$ and zero initialization for their state-space realizations.

To sum up, the block diagram shown in Figure 10 illustrates the architecture of $\mathcal{L}_1$ adaptive control for the class of systems given by (47).

6. Extended $\mathcal{L}_1$ adaptive control of Parallel Manipulators

Let $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathbb{R}^4$ be the desired joint position, velocity and acceleration trajectories respectively. These reference trajectories to be tracked by the active joints of Veloce, are generated in task-space according to a specific task of the moving platform (e.g. pick-and-place). Then, adequate kinematic relationships are used to compute the corresponding joint quantities. The control objective is to ensure that the traveling plate of the manipulator tracks, as accurately as possible, the desired trajectories regardless of any inherent nonlinearities or external disturbances. To quantify the control objective, consider the combined joint tracking error $r(t) \in \mathbb{R}^4$ defined by

$$r(t) = (\dot{q} - \dot{q}_d) + \Lambda\,(q - q_d) \tag{59}$$

where $\Lambda \in \mathbb{R}^{4\times 4}$ is a symmetric positive-definite design matrix. Since $-\Lambda$ is Hurwitz, it follows that (59) is BIBO-stable; i.e. there exist $L_1,\ L_2 > 0$ such that [40]

$$\|q_\tau\|_{\mathcal{L}_\infty} \leq L_1 \|r_\tau\|_{\mathcal{L}_\infty} + L_2 \tag{60}$$

where $\|(.)_\tau\|_{\mathcal{L}_\infty}$ denotes the truncated $\mathcal{L}_\infty$-norm of $(.)$. The proposed control law to achieve the desired tracking performance is given by:

$$\Gamma = M_0(q_d)\ddot{q}_d + C_0(q_d, \dot{q}_d)\dot{q}_d + G_0(q_d) + A_m r + \Gamma_{ad}, \tag{61}$$

The proposed control law in (61) can be thought of as the combination of three distinct terms. Each term has a specific role in the closed-loop system:

- The first term (i.e. $M_0(q_d)\ddot{q}_d + C_0(q_d, \dot{q}_d)\dot{q}_d + G_0(q_d)$) is a nominal feedforward inverse dynamics term intended to minimize the effect of the inherent nonlinearities of the system and hence, to improve the tracking performance.
- The second term (i.e. $A_m r$) consists of a stabilizing state-feedback term specifying the desired closed-loop behavior of the system.
- More importantly, the last term Γ_{ad}, which will be detailed in the subsequent development, is the adaptive signal responsible of rejecting all the remaining disturbances such as friction effects, external disturbances, etc.

Substituting the control law (61) into (29) results in the following error dynamics

$$\dot{r}(t) = A_m r(t) + \Gamma_{ad}(t) - \eta(t, \zeta(t)), \quad r(0) = r_0 \tag{62}$$

where $\eta(t, \zeta(t)),\ \zeta = \left[r^T, q^T, \dot{q}^T\right]^T$ is a nonlinear function that gathers all the remaining nonlinearities that result from applying the control law (61) to (29) and is given by

$$\begin{aligned} \eta(t, \zeta(t)) &= M^{-1}(q)(\tilde{M}(q)\ddot{q}_d + \tilde{N}(q, \dot{q})) \\ &+ (\mathbb{I} - M^{-1}(q))(A_m r + \Gamma_{AD}) \\ &- \Lambda r + \Lambda^2(q - q_d) \end{aligned} \tag{63}$$

where $\tilde{M}(q) \triangleq M(q) - \hat{M}(q_d)$, $\tilde{N}(q, \dot{q}) \triangleq N(q, \dot{q}) - \hat{N}(q_d, \dot{q}_d)$ and $\mathbb{I} \in \mathbb{R}^{4\times 4}$ denotes the identity matrix. For the subsequent parametrization of $\eta(t, \zeta(t))$, let's consider the following non-restricting assumptions [41]

Assumption 7. *There exist positive B such that $\|\eta(t,0)\| \leq B$ holds for all $t \geq 0$.*

Assumption 8. *η is continuous in its arguments, and for arbitrary $\delta > 0$, there exist $d_{\eta_t}(\delta)$ and $d_{\eta_x}(\delta)$ such that*

$$\left\| \frac{\partial \eta(t,\zeta)}{\partial t} \right\|_\infty \leq d_{\eta_t}(\delta), \quad \left\| \frac{\partial \eta(t,\zeta)}{\partial \zeta} \right\|_\infty \leq d_{\eta_\zeta}(\delta) \tag{64}$$

Assumption 9. *for $\tau \geq 0$, $\|r_\tau\|_{\mathcal{L}_\infty} \leq \rho$ and $\|\dot{r}_\tau\|_{\mathcal{L}_\infty} \leq d_r$, for some positive constants ρ and d_r. Now, for some arbitrary $\gamma > 0$, let*

$$\bar{\rho} \triangleq \max\left\{\rho + \gamma, L_1(\rho+\gamma) + L_2\right\}, \quad L_\rho \triangleq \frac{\bar{\rho}}{\rho} d_{\eta_\zeta}(\bar{\rho}) \tag{65}$$

It follows from Lemma A.9.2 in [41] that $\eta(t,\zeta(t))$ can be parametrized as follows

$$\eta(t,\zeta(t)) = \theta(t)\|r_\tau\|_{\mathcal{L}_\infty} + \sigma(t) \tag{66}$$

where $\theta(t), \sigma(t) \in \mathbb{R}^4$ are differentiable functions. It follows that (62) can be rewritten as

$$\dot{r}(t) = A_m r(t) + \Gamma_{AD}(t) - \left(\theta(t)\|r_\tau\|_{\mathcal{L}_\infty} + \sigma(t)\right), \quad r(0) = r_0 \tag{67}$$

Since the nonlinear function $\eta(t,\zeta(t))$ is unknown due to uncertainties and unmeasured disturbances, the control term $\Gamma_{ad}(t)$ should be adaptively designed in order to obtain estimates of $\theta(t)$ and $\sigma(t)$. Now, consider the following state predictor of the combined tracking error

$$\begin{aligned} \dot{\hat{r}}(t) =& A_m \hat{r}(t) + \Gamma_{ad}(t) - \left(\hat{\theta}(t)\|r_\tau\|_{\mathcal{L}_\infty} + \hat{\sigma}(t)\right) \\ & - K\tilde{r}(t), \quad \hat{r}(0) = r_0 \end{aligned} \tag{68}$$

where $\tilde{r}(t) \triangleq \hat{r}(t) - r(t)$ and $K \in \mathbb{R}^{4\times 4}$ is a design parameter introduced to reject high-frequency noise [14]. $\hat{\theta}(t)$ and $\hat{\sigma}(t)$ are real-time estimates of $\theta(t)$ and $\sigma(t)$ respectively, their evolution is governed by the following projection-based adaptive laws in order to ensure their boundedness [41]

$$\dot{\hat{\theta}}(t) = \Sigma\mathbf{Proj}\left(\hat{\theta}(t), P\tilde{r}(t)\|r_\tau\|_{\mathcal{L}_\infty}\right), \quad \hat{\theta}(0) = \hat{\theta}_0 \tag{69}$$

$$\dot{\hat{\sigma}}(t) = \Sigma\mathbf{Proj}\left(\hat{\sigma}(t), P\tilde{r}(t)\right), \quad \hat{\sigma}(0) = \hat{\sigma}_0 \tag{70}$$

where $\Sigma \in \mathbb{R}^+$ is the adaptive gain, $P = P^T > 0$ is the solution to the algebraic Lyapunov equation $A_m^T P + PA_m = -Q$ for some arbitrary $Q = Q^T > 0$. Since the estimated parameters are bounded thanks to the projection operator, they satisfy $\|\hat{\theta}(t)\|_\infty < \theta_b, \quad \|\hat{\sigma}(t)\|_\infty < \sigma_b, \quad \forall t \in [0,T]$. The adaptive control signal $\Gamma_{ad}(t)$ is the output of the following system given in Laplace domain

$$\Gamma_{ad}(s) = C(s)\hat{\eta}(s) \tag{71}$$

where $\hat{\eta}(s)$ is the Laplace transform of $\hat{\eta}(t) = \left(\hat{\theta}(t)\|r_t\|_{\mathcal{L}_\infty} + \hat{\sigma}(t)\right)$ and $C(s)$ is a diagonal matrix of filters whose elements are BIBO-stable strictly proper transfer functions satisfying a unity DC gain and zero initialization for their state-space realizations.

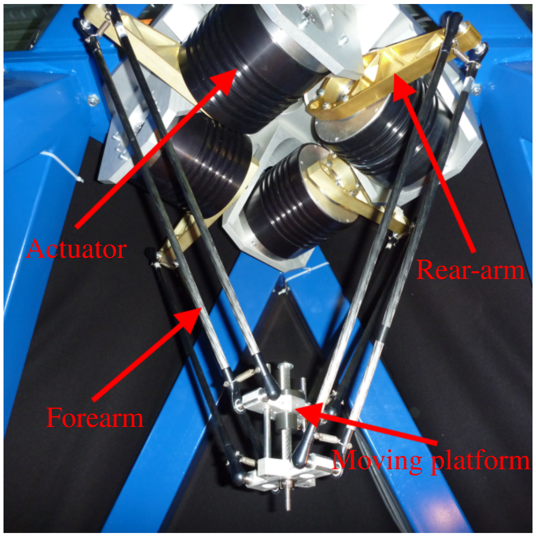

Figure 11. View of the experimental setup of the VELOCE robot.

7. Real-Time Experiments and Results

In order to demonstrate the relevance of the proposed extended $\mathcal{L}_1$ adaptive controller, real-time experiments were conducted on an experimental testbed consisting of the Veloce robot.

7.1. Experimental Testbed of the VELOCE robot

The actuators of VELOCE are the TMB0140-100-3RBS ETEL direct-drive motors. They can provide a maximum peak torque of 127 Nm and they are able to reach 550 rpm of speed. Each actuator is equipped with a non-contact incremental optical encoder providing a total number of 5000 pulses per revolution. The global structure of the manipulator is capable of reaching 10 m/s of maximum velocity of the traveling plate, 200 m/s^2 of maximum acceleration and is able to handle a maximum payload of 10 Kg. The control architecture is implemented using Simulink from Mathworks and compiled using the XPC Target real-

Table 3. Summary of the controllers' parameters

Parameter	Description	Value
Λ	position error weight	$10 \times \text{diag}(65, 65, 65, 65)$
A_m	transient response matrix	$\text{diag}(-6, -6, -6, -6)$
θ_{max}	upper bound on θ	50
σ_{max}	upper bound on σ	30
Σ	adaptation gain	10^6
K	noise rejection gain	$10^3 \times \text{diag}(6, 6, 6, 6)$
$C(s)$	$\mathcal{L}_1$ low-pass design filter	$144/(s^2 + 21.6s + 144)$

time toolbox. The resulting low-level code is then uploaded to the target PC; an industrial computer cadenced at 10 KHz (i.e. sample time of 0.1 ms). The experimental testbed is displayed in Figure 11.

7.2. Real-time Experimental Results

To demonstrate the relevance of the proposed contribution and to highlight its benefits, both the $\mathcal{L}_1$ adaptive controller in [16] and the proposed extended one were implemented in real-time on VELOCE. As a reminder, the implemented control law in [16] consists only of the stabilizing term and the adaptive term, it is expressed as

$$\Gamma(t) = A_m r(t) + \Gamma_{ad}(t) \tag{72}$$

The traveling plate of the manipulator had to perform several spatial point-to-point displacements as well as rotations inside the manipulator's workspace. We use 5^{th} order polynomials to generate the desired trajectories in Cartesian space, then the inverse kinematics problem is solved online to determine the corresponding joint trajectories. The duration of each point-to-point trajectory was fixed to 0.2 sec. The desired joint trajectories were obtained by solving the Inverse Kinematics problem in real-time while the actual ones are available from the encoders measurements. The manipulator is not equipped with external sensors, hence the actual Cartesian position is obtained by solving the Forward Kinematics problem in real-time as well. We choose the same control parameters for both controllers, they are summarized in Table 3. The estimated functions were initialized to zero ($\hat{\theta}(0) = [0,0,0,0]^T$, $\hat{\sigma}(0) = [0,0,0,0]^T$).

A comparison of the Cartesian tracking errors between both controllers is depicted in Figure 12. The plots are zoomed in the interval $[5,7]$ seconds for clarity. It can be clearly seen that the augmented controller performs much better than the original $\mathcal{L}_1$ adaptive controller especially on translations on the x and y-axes, while the enhancement is smaller on z-axis and the rotational DOF. To quantify the improvement brought by the proposed extended $\mathcal{L}_1$ adaptive controller, we formulate the following criteria based on the Root Mean Square (RMS) of the tracking errors:

$$\mathrm{RMS}_J = \left(\sum_{i=1}^{4} \mathrm{RMS}^2(\epsilon_{qi})\right)^{\frac{1}{2}} \tag{73}$$

$$\mathrm{RMS}_C = \left(\mathrm{RMS}^2(\epsilon_x) + \mathrm{RMS}^2(\epsilon_y) + \mathrm{RMS}^2(\epsilon_z) + + \mathrm{RMS}^2(\epsilon_s)\right)^{\frac{1}{2}} \tag{74}$$

$$\tag{75}$$

where $\epsilon_{(.)}$ is the error between the desired and actual position and $\mathrm{RMS}(.)$ is the standard root mean square function. The obtained results are summarized in Table 4. It can be noticed that the improvement is very significant regarding the joint tracking errors (up to 52%) and the translational movements (up to 68.8%) while only small improvement is observed on the rotation of the traveling plate (only 4.2%). This result highlights the benefits of using the dynamics of the manipulator in the control loop in terms of tracking performance.

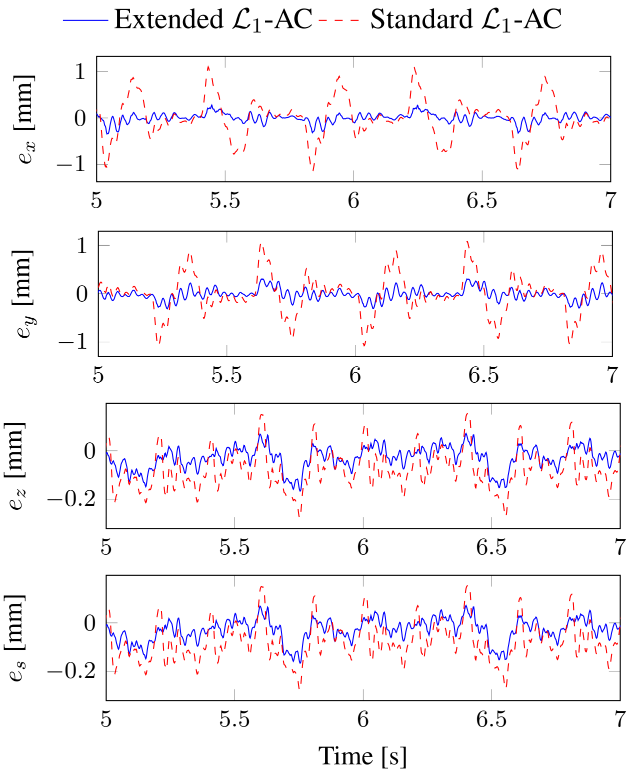

Figure 12. Cartesian tracking errors: Extended $\mathcal{L}_1$ adaptive controller (solid line), $\mathcal{L}_1$ adaptive controller (dashed line).

The generated control inputs are shown in Figure 15. The control inputs remain within the allowable range and do not exceed the limits of the actuators. Overall, it can be observed that the amplitudes of the control inputs in the case of the augmented controller are slightly smaller than those of the standard $\mathcal{L}_1$ adaptive one. Hence, the following conclusion can be drawn: the proposed controller significantly improves the tracking performance of the closed-loop system while consuming less energy than the standard one.

The evolution of the estimated nonlinear functions $\hat{\theta}(t)$ and $\hat{\sigma}(t)$ versus time is depicted in Figure 13 and Figure 14 respectively. These figures clearly demonstrate the relevance of the proposed contribution. Indeed, the addition of the model-based feedforward considerably helps in compensating for the modeling inherent nonlinearities of the manipulator. Therefore, the remaining nonlinearities that have to be compensated for by the adaptive signal are of smaller amplitudes. This result is further illustrated in Figure 16 where only the evolution of the adaptive component of (61) and (72) are plotted (i.e $\Gamma_{ad}(t)$). We can see that the adaptive signal, needed to compensate for the remaining nonlinearities, of the proposed controller is of lower amplitude than that of the conventional one.

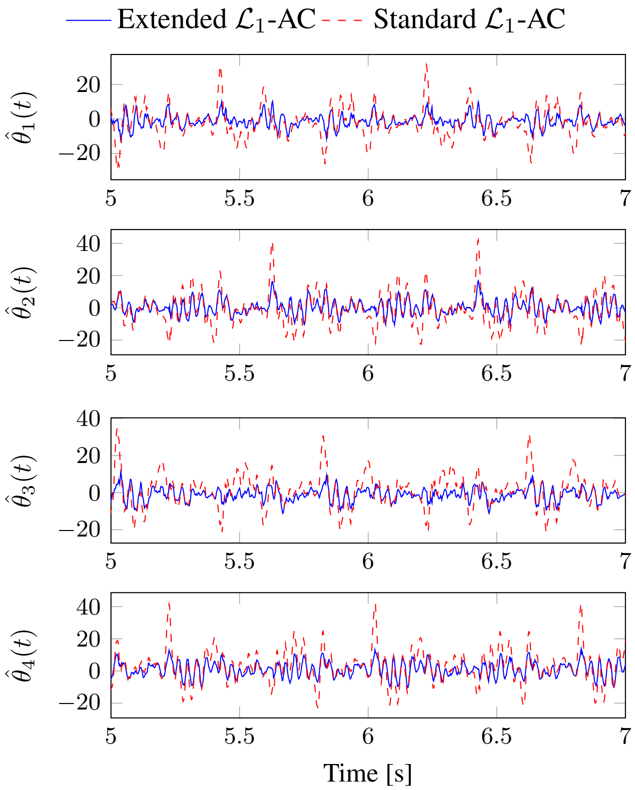

Figure 13. Evolution of $\hat{\theta}$ versus time.

8. Conclusions and Future Work

In this work, a new controller based on $\mathcal{L}_1$ adaptive control for parallel manipulators is presented. Standard $\mathcal{L}_1$ adaptive control do not rely on any knowledge about the dynamics of the controlled system which may result in a poor tracking perfromance. Since a simplified dynamic model for parallel manipulators is relatively easy to derive, we propose to take advantage of the modelled dynamics in the control loop to enhance the performance of $\mathcal{L}_1$ adaptive control. For this reason, we augment the standard $\mathcal{L}_1$ adaptive controller with a computed feedforward term based on the nominal dynamics of the parallel manipulator and the desired trajectories. The main motivation behind such a proposition is to improve the tracking performance of the manipulator. In fact, the included dynamics partially com-

Table 4. Tracking performance comparison

	$\mathcal{L}_1$-AC	Extended $\mathcal{L}_1$-AC	Improvement
RMS$_J$ [deg]	0.1012	0.0486	52 %
RMS$_C$ [mm]	0.4634	0.195	69.7 %

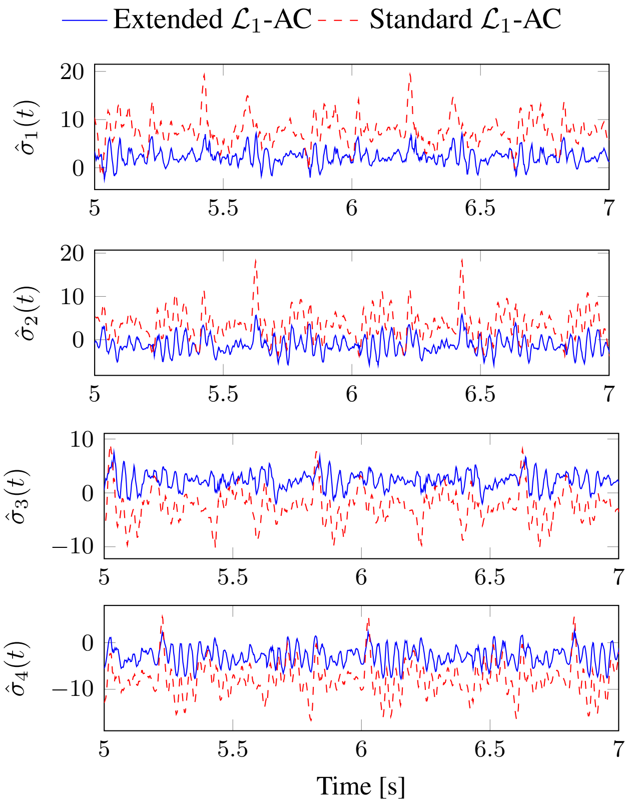

Figure 14. Evolution of $\hat{\sigma}$ versus time.

pensates for the inherent nonlinear dynamics which results in an improved tracking performance. Real-time experiments on a 4-DOF parallel manipulator show that the proposed augmented control scheme significantly reduces the tracking errors as compared to standard $\mathcal{L}_1$ adaptive control. Future work may consider evaluating the proposed controller in other working scenarios and on other parallel manipulators prototypes.

Acknowledgments.

This research was supported by the French National Research Agency, within the project ARROW (ANR-2011-BS3-006-01-ARROW).

References

[1] J. Luh, M. Walker, and R. Paul, “Resolved-acceleration control of mechanical manipulators,” *IEEE Trans. Autom. Control*, vol. 25, no. 3, pp. 468–474, Jun. 1980.

[2] D. Koditschek, “Natural motion for robot arms,” in *Proc. The 23rd IEEE Conference on Decision and Control (CDC’84)*, Las Vegas, Nevada, USA, Dec. 1984, pp. 733–735.

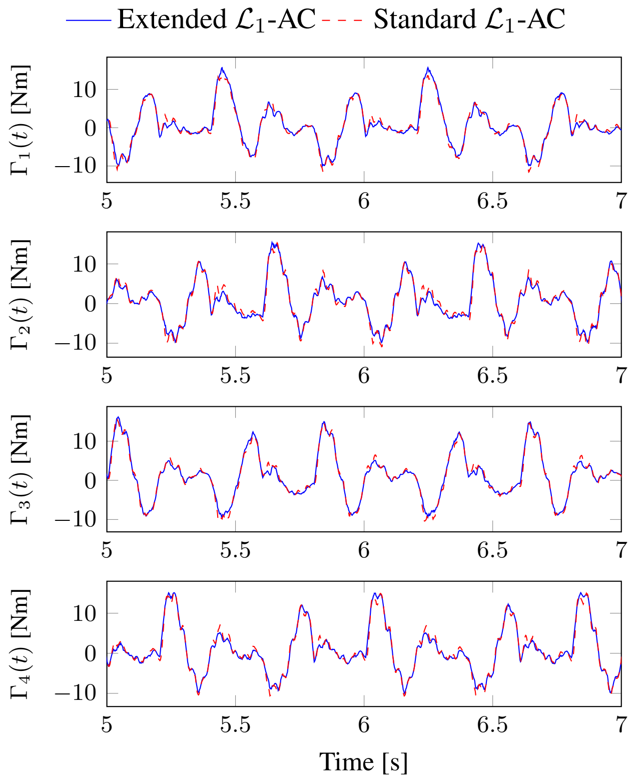

Figure 15. Evolution of the control input torques Γ versus time.

[3] B. Paden and R. Panja, "Globally asymptotically stable pd+ controller for robot manipulators," *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, Jun. 1988.

[4] S. Dubowsky and D. DesForges, "The application of model-referenced adaptive control to robotic manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 101, no. 3, pp. 193–200, Jun. 1979.

[5] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulators - compensation of nonlinearity and decoupling control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 2, pp. 127–135, Jun. 1986.

[6] J. Craig, P. Hsu, and S. Sastry, "Adaptive control of mechanical manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, Jun. 1987.

[7] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, no. 6, pp. 877–888, Oct. 1989.

[8] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators," *The International Journal of Robotics Research*, vol. 9, no. 3, pp. 74–92, Jun. 1990.

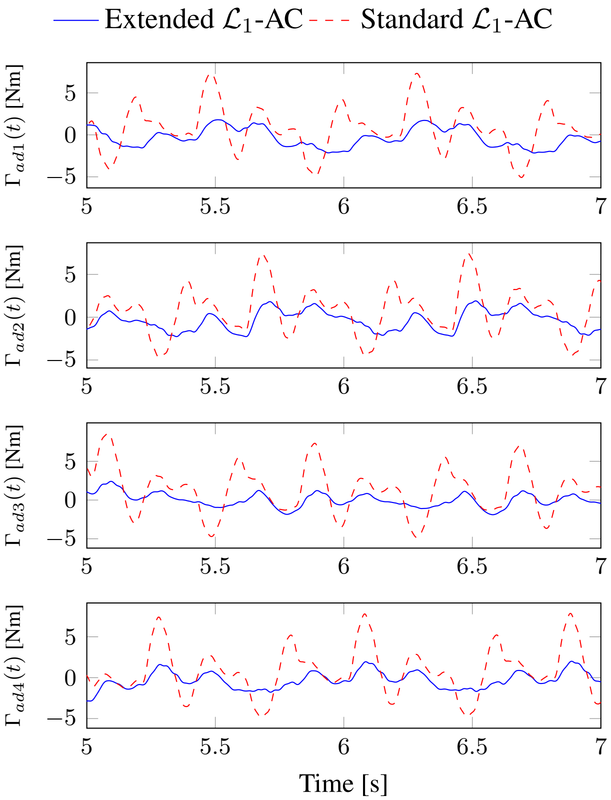

Figure 16. Evolution of the adaptive control input torques Γ_{ad} versus time.

[9] M. Bennehar, A. Chemori, and F. Pierrot, "A new extension of direct compensation adaptive control and its real-time application to redundantly actuated pkms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)*, Chicago, Illinois, USA, Sep. 2014, pp. 1670–1675.

[10] C. Cao and N. Hovakimyan, "Design and analysis of a novel $\mathcal{L}_1$ adaptive controller, part i: Control signal and asymptotic stability," in *American Control Conference (ACC'06)*, Minneapolis, USA, Jun. 2006, pp. 3397–3402.

[11] ——, "Design and analysis of a novel $\mathcal{L}_1$ adaptive controller, part ii: Guaranteed transient performance," in *American Control Conference (ACC'06)*, Minneapolis, USA, Jun. 2006, pp. 3403–3408.

[12] I. Gregory, C. Cao, E. Xargay, N. Hovakimyan, and X. Zou, "L1 adaptive control design for nasa airstar flight test vehicle," *AIAA Guidance, Navigation, and Control Conference*, p. 5738, Aug. 2009.

[13] D. Maalouf, V. Creuze, and A. Chemori, "A novel application of multivariable $\mathcal{L}_1$ adaptive control: From design to real-time implementation on an underwater vehicle,"

in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'12)*, Vilamoura, Algarve, Portugal, Jan. 2012, pp. 76–81.

[14] K. D. Nguyen, H. Dankowicz, and N. Hovakimyan, “Marginal stability in $\mathcal{L}_1$ adaptive control of manipulators,” in *Proceedings of the 9th ASME International Conference on Multibody Systems, Nonlinear Dynamics, and Control*, Portland, OR, USA, 2013.

[15] L. Techy, C. K. Reddy, C. A. Woolsey, C. Cao, and N. Hovakimyan, “Nonlinear control of a novel two-link pendulum,” in *American Control Conference (ACC'07)*, New York, USA, 2007, pp. 19–24.

[16] M. Bennehar, A. Chemori, and F. Pierrot, “L1 adaptive control of parallel kinematic manipulators: Design and real-time experiments,” in *IEEE International Conference on Robotics and Automation (ICRA'15)*, Seattle, Washington, USA, May 2015, pp. 1587–1592.

[17] K. Astrom, *Adaptive Control (Second Edition)*. Addison-Wesley, 1995.

[18] C. C. Nguyen, S. S. Antrazi, Z. L. Zhou, and C. E. Campbell, “Adaptive control of a stewart platform-based manipulator,” *Journal of Robotic Systems*, vol. 10, no. 5, pp. 657–687, 1993.

[19] S. Dubowsky and D. DesForges, “The application of model-referenced adaptive control to robotic manipulators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 101, no. 3, pp. 193–200, Sep. 1979.

[20] S. Nicosia and P. Tomei, “Model reference adaptive control algorithms for industrial robots,” *Automatica*, vol. 20, no. 5, pp. 635 – 644, 1984.

[21] B. Yegnanarayana, *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[22] X. Qingsong and L. Yangmin, “A 3-prs parallel manipulator control based on neural network,” in *Advances in Neural Networks ISNN 2007*, D. Liu, S. Fei, Z.-G. Hou, H. Zhang, and C. Sun, Eds. Springer Berlin Heidelberg, 2007, pp. 757–766.

[23] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC Press, 1998.

[24] Y. Li and Y. Wang, “Trajectory tracking control of a redundantly actuated parallel robot using diagonal recurrent neural network,” in *International Conference on Natural Computation (ICNC'09)*, vol. 2, Aug 2009, pp. 292–296.

[25] J. J. Craig, P. Hsu, and S. S. Sastry, “Adaptive control of mechanical manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.

[26] H. Cheng, Y. Yiu, and Z. Li, “Dynamics and control of redundantly actuated parallel manipulators,” *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 4, pp. 483–491, Dec. 2003.

[27] J. Merlet, *Parallel Robots, Second Edition*. Dordrecht, Netherlands: Springer, 2006.

[28] W. Shang, S. Cong, and Y. Ge, “Adaptive computed torque control for a parallel manipulator with redundant actuation,” *Robotica*, vol. 30, no. 3, pp. 457–466, May 2012.

[29] M. Honegger, R. Brega, and G. Schweiter, “Application of a nonlinear adaptive controller to a 6 dof parallel manipulator,” in *IEEE International Conference on Robotics and Automation (ICRA’00)*, vol. 2, San Francisco, CA, USA, Apr. 2000, pp. 1930–1935.

[30] G. Sartori Natal, A. Chemori, and F. Pierrot, “Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1520–1535, Jul. 2015.

[31] D. Corbel, M. Gouttefarde, O. Company, and F. Pierrot, “Actuation redundancy as a way to improve the acceleration capabilities of 3t and 3t1r pick-and-place parallel manipulators,” *Journal of Mechanisms and Robotics*, vol. 2, no. 4, pp. 1–13, 2010.

[32] W. Shang and S. Cong, “Nonlinear adaptive task space control for a 2-dof redundantly actuated parallel manipulator,” *Nonlinear Dynamics*, vol. 59, no. 1-2, pp. 61–72, 2010.

[33] J.-J. Slotine and L. Weiping, “On the adaptive control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, Sep. 1987.

[34] R. Clavel, “Device for the movement and positioning of an element in space,” Dec. 1985, uS Patent 4,976,582. [Online]. Available: https://patents.google.com/patent/US4976582A

[35] J. Merlet, *Parallel Robots, Second Edition*. Dordrecht, Netherlands: Springer, 2006.

[36] A. Codourey, “Dynamic modeling of parallel robots for computed-torque control implementation,” *The International Journal of Robotics Research*, vol. 17, pp. 1325–1336, 1998.

[37] D. Corbel, M. Gouttefarde, and O. Company, “Towards 100g with pkm. is actuation redundancy a good solution for pick and place?” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’10)*, Anchorage, Alaska, May 2010, pp. 4675–4682.

[38] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, Pennsylvania, USA: SIAM, 2010.

[39] M. Bennehar, A. Chemori, and F. Pierrot, “L1 adaptive control of parallel kinematic manipulators: Design and real-time experiments,” in *IEEE International Conference on Robotics and Automation (ICRA’15)*, Seattle, Washington, USA, May 2015, pp. 1587–1592.

[40] H. Khalil, *Nonlinear Systems*. Prentice-Hall, 2002.

[41] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, Pennsylvania, USA: SIAM, 2010.