Towards Minimum-Information Adaptive Controllers for Robot Manipulators

Tobia Marcucci[1,2], Cosimo Della Santina[1], Marco Gabiccini[1,2,3], and Antonio Bicchi[1,2]

***Abstract*— The aim of this paper is to move a step in the direction of determining the minimum amount of information needed to control a robot manipulator within the framework of adaptive control. Recent innovations in the state of the art show how global asymptotic trajectory tracking can be achieved despite the presence of uncertainties in the kinematic and dynamic models of the robot. However, a clear distinction between which parameters can be included among the uncertainties, and which parameters can not, has not been drawn yet. Since most of the adaptive control algorithms are built on linearly parameterized models, we propose to reformulate the problem as finding a procedure to determine whether and how a given dynamical system can be linearly parameterized with respect to a specific set of parameters.**

Within this framework, we show how the trajectory tracking problem of a manipulator can be accomplished with the only knowledge of the number of joints of the manipulator. As an illustrative example, we present the end-effector trajectory tracking control of a robot initialized with the kinematic model of a different robot.

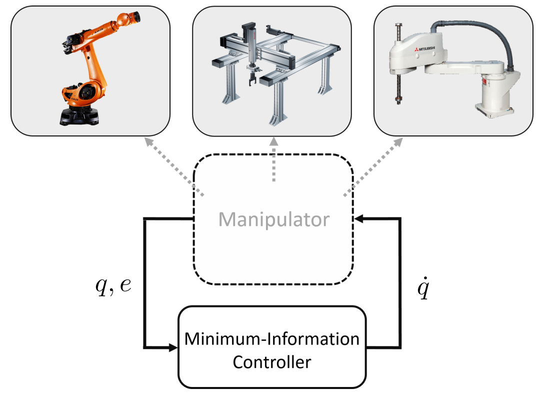

Fig. 1. The only information about the robot structure necessary for the operation of the proposed controller is the number of joints.

I. Introduction

Originally, adaptive control had been applied to robotic manipulators in order to cope with dynamic uncertainties that appears linearly in the equations of motion [1], [2], [3]. To this classical problem, adaptive control provided a strong response guaranteeing global asymptotic stability of the controlled system, regardless the magnitude of the uncertainties. More recently, several solutions have been proposed to the task-space trajectory tracking problem with linearly parameterizable dynamic and kinematic uncertainties [4], [5] and readily improved to take into account further uncertainties, such as actuator dynamics [6]. Also in these cases, global asymptotic stability has been guaranteed. Motivated by these results, we ask ourselves what are the limits of these controllers and how far they can be pushed or, in other words, what is the minimum amount of information needed to design an adaptive controller for a robot manipulator.

As in the case of robotics, the great majority of adaptive control theory has been developed for plant models in which the unknowns parameters appear linearly [7], [8]. In light of this, we found reasonable to reformulate the problem as finding a procedure to determine whether and how a given dynamical system can be linearly parameterized, i.e. described by a model that consists in the product of a regressor matrix, function of the state variables and the known parameters, and a nonlinear vector function of the unknown parameters.

The set of parameters with respect to which the dynamic equations of a robot can be linearly parameterized is well known [9]. Furthermore, many procedures tailored to derive such linear models have been proposed [10], [11], [12]. Nonetheless, we were not able to find exhaustive discussions for other kinds of uncertainties, e.g. kinematic ones. In fact, when such parametrizations are needed (e.g. [4], [5]), linear models are typically derived case-by-case. Moreover, in these situations even the dynamic regressor can not be derived through classical algorithms, since kinematic parameters have to be factorized also at the dynamic level. All these issues limit the applicability of these control algorithms to small scale systems with small number of uncertain parameters.

In this paper we derive sufficient conditions for the existence of a linear parametrization of a given system with respect to a set of parameters. We also describe a general purpose procedure that, given the algorithm used to compute the model of the system, returns such parametrization. The procedure leverages upon some basic properties of Linearly Parameterizable (LP) functions, that we briefly introduce and discuss. Starting from these ideas we develop a trajectory tracking controller for serial manipulators completely independent from the structure of the robot, by showing how not only all the Denavit-Hartenberg (DH) parameters [9], but also the joint types (i.e. Revolute (R) or Prismatic (P)) can be linearly parameterized (see Figure 1). We consider here the kinematic case, leaving the extension to the dynamic case to future works. (Note that, from a theoretical point of view, the dynamic extension directly derives from the here proposed

[1]Research Center "Enrico Piaggio", University of Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy.

[2]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genoa, Italy.

[3]Dipartimento di Ingegneria Civile e Industriale, University of Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy.

Email: tobia.marcucci@ing.unipi.it.

algorithm, and the adaptive controllers discussed above.)

The paper is organized as follows. In Section II LP functions are introduced, whereas in Section III the proposed procedure is described. In Section IV we apply this technique to design a minimum-information controller for the kinematic trajectory tracking problem. After the presentation of a toy example, the effectiveness of the algorithm is tested in simulation on a fully-actuated manipulator with 3 Degrees of Freedom (DoFs). In Section V conclusions and future works are illustrated.

II. Linearly Parametrizable Functions

In this section we summarize and formalize some elementary concepts about LP functions that are at the core of the algorithm presented in the following section.

Definition 1 (Linearly Parametrizable (LP) function)**.** *Let us consider a scalar function $f(x,p):\mathbb{R}^{n_x}\times\mathbb{R}^{n_p}\to\mathbb{R}$. If there exist two vector-valued functions $f_x(x):\mathbb{R}^{n_x}\to\mathbb{R}^{m_f}$ and $f_p(p):\mathbb{R}^{n_p}\to\mathbb{R}^{m_f}$ such that*

$$f(x,p)=f_x^\top(x)f_p(p), \tag{1}$$

we say that $f(x,p)$ is a LP function with factors $f_x(x)$, $f_p(p)$.

In case $f(x,p)\in\mathbb{R}^{n_f}$ is a vector function, we call it LP if all of its elements $f_i(x,p)$, $i=1,\dots,n_f$ are LP and (1) is generalized as

$$f(x,p)=F_x^\top(x)f_p(p),$$

with $F_x:\mathbb{R}^{n_x}\to\mathbb{R}^{m_f}\times\mathbb{R}^{n_f}$. Starting from the factors $f_{i,x}(x)$, $f_{i,p}(p)$ of $f_i(x,p)$, we can for example derive $F_x(x)$ and $f_p(p)$ as

$$\begin{aligned}F_x(x)=&\operatorname{diag}(f_{1,x}(x),\dots,f_{n_f,x}(x)),\\ f_p(p)=&f_{1,p}(p)\oplus\cdots\oplus f_{n_f,p}(p)=\bigoplus_{i=1}^{n_f}f_{i,p}(p),\end{aligned}$$

where $\oplus$ denotes the vertical concatenation operator.

Some helpful closure properties of LP scalar functions [13] can be formalized as follows, where the symbol $\otimes$ denotes the Kronecker product and $\operatorname{id}(\cdot)$ the identity operator.

Property 1 (Closure under sum of LP functions)**.** *Given two LP functions $g(x,p)$ and $h(x,p)$, their sum*

$$f(x,p)=g(x,p)+h(x,p)$$

is a LP function with factors

$$f_x(x)=g_x(x)\oplus h_x(x),\ \ f_p(p)=g_p(p)\oplus h_p(p).$$

Proof. The thesis follows immediately from he definition of matrix product. □

Property 2 (Closure under multiplication of LP functions)**.** *Given two LP functions $g(x,p)$ and $h(x,p)$, their product*

$$f(x,p)=g(x,p)\cdot h(x,p)$$

is a LP function with factors

$$f_x(x)=g_x(x)\otimes h_x(x),\ \ f_p(p)=g_p(p)\otimes h_p(p).$$

Proof. By some manipulations we obtain

$$\begin{aligned}f(x,p)=&\sum_{i=1}^{m_g}g_{x,i}(x)g_{p,i}(p)(h_x(x)^\top h_p(p))\\ =&\sum_{i=1}^{m_g}(g_{x,i}(x)h_x(x))^\top(g_{p,i}(p)h_p(p))\\ =&\left(\bigoplus_{i=1}^{m_g}g_{x,i}(x)h_x(x)\right)^\top\left(\bigoplus_{i=1}^{m_g}g_{p,i}(p)h_p(p)\right)\\ =&(g_x(x)\otimes h_x(x))^\top(g_p(p)\otimes h_p(p)),\end{aligned}$$

hence the thesis. □

Property 3 (Closure under differentiation of LP functions)**.** *Given a LP function $h(x,p)$, its derivatives*

$$f(x,p)=\mathrm{D}_{x_i}(h(x,p)),\ \ g(x,p)=\mathrm{D}_{p_i}(h(x,p))$$

are LP functions with factors

$$\begin{aligned}&f_x(x)=\mathrm{D}_{x_i}(h_x(x)),\ \ f_p(p)=\operatorname{id}(h_p(p)),\\ &g_x(x)=\operatorname{id}(h_x(x)),\ \ g_p(p)=\mathrm{D}_{p_i}(h_p(p)).\end{aligned}$$

Proof. The thesis follows immediately from product rule of derivatives. □

The closure of LP vector functions with respect to matrix sum, matrix multiplication, and matrix differentiation follows immediately from Properties 1–3.

The practical meaning of LP functions is clear interpreting x as the state vector of a dynamic system, p as the vector gathering all the unknown coefficients, $f_x(x)$ as the linear regressor and $f_p(p)$ as the parameter vector.

In this work, we limit our analysis to the operations necessary to model mechanical systems, however following conclusions are drawn considering generic operators. For this reason, we define $\mathbb{O}$ as the set containing all the operators with respect to which LP functions are closed.

III. Linear Parametrization of Algorithmically-Computed Functions

Given a generic function $f(x,p)$, the factorization process might be very demanding from a computational point of view; moreover, in some cases, it is not even obvious to determine whether a function $f(x,p)$ is LP. In this section we show how, for the special class of Algorithmically-Computed Functions (ACFs) (defined below), these issues can be addressed in an systematic manner.

A. Existence of a LP Model

Let us denote with $\mathcal{F}$ an algorithm that composes a set of input functions $g^i(x,p)\in\mathbb{R}$, $i=1,\dots,n_g$, through a set of operators $\mathrm{op}^j:\mathbb{R}^{n_j}\to\mathbb{R}$, $j=1,\dots,n_o$, with n_j arity of the jth operator. We call the output function of the algorithm ACF and we represent it as[1]

$$\mathcal{F}(g^1(x,p),\dots,g^{n_g}(x,p);\mathrm{op}^1,\dots,\mathrm{op}^{n_o}).$$

[1]For a more formal representation of algorithms, the reader can refer to [14].

TABLE I
OPERATORS AND INDUCED OPERATORS FROM PROPERTIES 1–3

Scalar function operator (op)	Regressor vector operator (op_x)	Parameter vector operator (op_p)
$+$	$\oplus$	$\oplus$
$\cdot$	$\otimes$	$\otimes$
D_{x_i}	D_{x_i}	id
D_{p_i}	id	D_{p_i}

Property 4 (Sufficient condition for the existence of a linear parametrization)**.** *A scalar function $f(x,p)$ is LP if it is the output of an algorithm $\mathcal{F}$ whose input functions $g^i(x,p)$ are LP $\forall i$ and whose operators $\mathrm{op}^j \in \mathbb{O}\ \forall j$.*

In all the situations where an algorithm is used to derive the model of a system, Property 4 allows to determine if this model can be factorized in a linear regressor and a vector of parameters only by checking the inputs and the operators of the algorithm.

B. Computation of a LP Model

Properties 1–3 show that, for each one of the three operations analysed, there are two induced operators that can be applied to the factors of the operand functions to obtain the factors of the result (Table I summarizes these relations).

We generalize this concept by calling op_x, op_p the operators defined between regressor vectors and parameter vectors, respectively, that are induced by the generic operator $\mathrm{op} \in \mathbb{O}$. Furthermore, let us denote with $\mathbb{O}_{\mathrm{ind}} \subseteq \mathbb{O}$ the set of all the operators op that produce such induction. We have that, given a set of operators $\mathrm{op}^1, \ldots, \mathrm{op}^{n_o} \in \mathbb{O}_{\mathrm{ind}}$, their composition $(\mathrm{op}^1 \circ \cdots \circ \mathrm{op}^{n_o}) \in \mathbb{O}_{\mathrm{ind}}$ induces the operators $(\mathrm{op}_x^1 \circ \cdots \circ \mathrm{op}_x^{n_o})$ and $(\mathrm{op}_p^1 \circ \cdots \circ \mathrm{op}_p^{n_o})$, defined between regressor vectors and parameter vectors, respectively. On the basis of on this observation, if Property 4 holds, the computation of the factors $f_x(x)$ and $f_p(p)$ can be accomplished with the procedure illustrated in Algorithm 1.

The following example demonstrates Algorithm 1 in case of a very simple ACF.

Example 1. *Consider the ACF*

$$f(x,p) = \mathrm{D}_{x_i}(g(x,p) \cdot h(x,p) + g(x,p)),$$

where $g(x,p) = g_x(x)^\top g_p(p)$ and $h(x,p) = h_x(x)^\top h_p(p)$. From Properties 1–3, all the operators employed to compute $f(x,p)$ belong to $\mathbb{O}_{\mathrm{ind}}$. Since both the input functions are LP, from Property 4, we conclude that $f(x,p)$ is LP. Thus we can apply Algorithm 1 with the induced operators listed in Table I, deriving

$$\begin{aligned} f_x(x) =& \mathrm{D}_{x_i}((g_x(x) \otimes h_x(x)) \oplus g_x(x)), \\ f_p(p) =& \mathrm{id}((g_p(p) \otimes h_p(p)) \oplus g_p(p)). \end{aligned}$$

From a practical view point, Algorithm 1 can be implemented on a computer very easily: it is only necessary to overload the operators of the original algorithm in accordance with Table I, so that the operations they perform are functions of the data type of the inputs.

Algorithm 1: Computation of LP Model

```
Result: f_x(x), f_p(p)
if Property 4 holds then
    for j = 1,...,n_o do
        if op^j ∈ O_ind then
            compute op_x^j, op_p^j
        else
            procedure not applicable
        end
    end
    for i = 1,...,n_g do
        compute g_x^i(x), g_p^i(p)
    end
    return
        f_x(x) = F(g_x^1(x),...,g_x^{n_g}(x); op_x^1,...,op_x^{n_o})
        f_p(p) = F(g_p^1(p),...,g_p^{n_g}(p); op_p^1,...,op_p^{n_o})
else
    procedure not applicable
end
```

IV. ADAPTIVE KINEMATIC CONTROLLER FOR TRAJECTORY TRACKING

Starting from the general methodology proposed in the previous section, we consider here the End-Effector (EE) trajectory tracking problem for a robotic manipulator.

A. End-Effector Velocity as a LP Function

In this section we apply LP functions to show that the linear and angular velocities of the EE of a robot manipulator, as well as any other point of it, are LP with respect to a given set of geometric parameters. Even if this property is widely used in the context of adaptive control [4], [5], [6], we were not able to find any proof of it. In addition, we show how, assuming the robot joints to be either R or P, the EE velocity is also LP with respect to each joint type.

The robot manipulator is assumed to be parameterized with DH convention [9], with parameters d_i, a_i, θ_i, α_i, $i = 1, \ldots, n_q$; the transformation matrix from the ith frame to the $(i-1)$th is then

$$A_i^{i-1} = \begin{bmatrix} \mathrm{c}_{\theta_i} & -\mathrm{s}_{\theta_i}\mathrm{c}_{\alpha_i} & \mathrm{s}_{\theta_i}\mathrm{s}_{\alpha_i} & a_i\mathrm{c}_{\theta_i} \\ \mathrm{s}_{\theta_i} & \mathrm{c}_{\theta_i}\mathrm{c}_{\alpha_i} & -\mathrm{c}_{\theta_i}\mathrm{s}_{\alpha_i} & a_i\mathrm{s}_{\theta_i} \\ 0 & \mathrm{s}_{\alpha_i} & \mathrm{c}_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \in SE(3),$$

where $SE(3)$ is the special Euclidean group of dimension 3. Furthermore, we define a binary parameter useful to describe in a unique form the transformations related to P and R joints

$$r_i = \begin{cases} 0 & \text{if the } i\text{th joint is P} \\ 1 & \text{if the } i\text{th joint is R} \end{cases}.$$

DH parameters can be now expressed as

$$\theta_i = \theta_{0,i} + r_i q_i, \quad d_i = d_{0,i} + q_i - r_i q_i,$$

where q_i is the ith element of the generalized coordinate vector $q \in \mathbb{R}^{n_q}$, whereas $\theta_{0,i}$ and $d_{0,i}$ represent the joint offsets for $q_i = 0$.

Collecting in a vector $p \in \mathbb{R}^{n_p}$ all the geometric parameters introduced so far, the EE position $x(q,p) \in \mathbb{R}^3$ and the rotation matrix $R(q,p) \in SO(3)$ of the EE (with $SO(3)$ denoting the special orthogonal group of dimension 3) can be obtained as

$$\begin{bmatrix} R(q,p) & x(q,p) \\ 0_{1\times 3} & 1 \end{bmatrix} = \prod_{i=1}^{n_q} A_i^{i-1},$$

where each A_i^{i-1} has to be right-multiplied to the previous. Up to this point, we can obtain the linear and angular velocities of the EE as follows

$$\begin{aligned} \dot{x}(q,\dot{q},p) =& \mathrm{D}_q(x(q,p))\dot{q}, \\ \mathrm{S}(\omega(q,\dot{q},p)) =& \left(\sum_{i=1}^{n_q} \mathrm{D}_{q_i}(R(q,p))\dot{q}_i\right) R^\top(q,p), \end{aligned}$$

where $\mathrm{S}(\cdot)$ is the skew-symmetric operator. The linear function of the joint velocities $\xi(q,\dot{q},p) = \dot{x}(q,\dot{q},p) \oplus \omega(q,\dot{q},p) = J(q,p)\dot{q}$ is called twist of the EE, with $J(q,p) \in \mathbb{R}^{6\times n_q}$ geometric Jacobian matrix.

Since LP functions are closed with respect to all the operators employed to compute $\xi(q,\dot{q},p)$ (namely: sum, multiplication, and differentiation), in order to state that $\xi(q,\dot{q},p)$ is LP we only have to check that the elements of A_i^{i-1} are LP for all i. Taking, as an example, the term $A_{i,11}^{i-1}$, we have

$$A_{i,11}^{i-1} = \mathrm{c}_{\theta_{0,i}}\mathrm{c}_{r_i q_i} - \mathrm{s}_{\theta_{0,i}}\mathrm{s}_{r_i q_i},$$

from which it would seem that the parameter r_i is not linearly factorizable (since its product with q_i is the argument of a trigonometric function). Nonetheless, we can exploit the fact that r_i is a binary variable noticing that

$$r_i \in \{0,1\} \Rightarrow \begin{cases} \mathrm{s}_{r_i q_i} \equiv r_i \mathrm{s}_{q_i} \\ \mathrm{c}_{r_i q_i} \equiv 1 - r_i + r_i \mathrm{c}_{q_i} \end{cases}. \quad (2)$$

A linear parametrization of $A_{i,11}^{i-1}$ can be readily found as

$$A_{i,11}^{i-1} = \begin{bmatrix} \mathrm{c}_{q_i} & \mathrm{s}_{q_i} & 1 \end{bmatrix} \begin{bmatrix} r_i \mathrm{c}_{\theta_{0,i}} \\ -r_i \mathrm{s}_{\theta_{0,i}} \\ \mathrm{c}_{\theta_{0,i}}(1-r_i) \end{bmatrix}.$$

This proves that $A_{i,11}^{i-1}$ is LP. Analogous considerations can be made for all the other elements of A_i^{i-1}, proving that $\xi(q,\dot{q},p)$ is a linearly parameterizable as

$$\xi(q,\dot{q},p) = Y(q,\dot{q})\pi(p),$$

where $Y(q,\dot{q}) \in \mathbb{R}^{6\times n_\pi}$ is a kinematic regressor matrix and $\pi(p) \in \mathbb{R}^{n_\pi}$ is a nonlinear function of the parameters. The computation of both $Y(q,\dot{q})$ and $\pi(p)$ is made very simple by the overloading procedure illustrated in Section III.

Remark 1. *Except for particular cases, the function $\pi(p)$ is a one-to-many map, so that original parameters p can not be determined even when $\pi(p)$ converges to its real value.*

B. Adaptive Control Law

In this section we derive the trajectory tracking adaptive controller we use in the simulations of Section IV-D. The results drawn here can be seen as a kinematic version of the controller presented in [4].

Let us denote with $x_d(t)$ the desired trajectory of the EE and with $\hat{\pi}(t)$ the time-varying estimate of the constant geometric parameters π. The position error of the EE (supposed to be available as a feedback signal) and the parameter estimation error are consequently defined as $e = x_d - x$ and $\tilde{\pi} = \pi - \hat{\pi}$. We have

$$\dot{e} = \dot{x}_d - Y_P\pi = \dot{x}_d - \hat{J}_P\dot{q} - Y_P\tilde{\pi}, \quad (3)$$

where $Y_P(q,\dot{q})$ is the block of the kinematic regressor related to the linear velocity (position regressor) and, similarly, $\hat{J}_P(q,\hat{\pi})$ is the estimated position Jacobian, with $\hat{J}_P(q,\hat{\pi})\dot{q} = Y_P(q,\dot{q})\hat{\pi}$. Notice that the product $Y_P\tilde{\pi}$ represents the error in the estimation of the end effector velocity, denoted as $\tilde{\dot{x}}$.

Let us consider the kinematic control law and the parameter update law

$$\dot{q}(q,e,\hat{\pi},t) = \hat{J}_P^{-1}(\dot{x}_d + Ke), \quad (4)$$

$$\dot{\hat{\pi}}(q,e,\hat{\pi},t) = -QY_P^\top e, \quad (5)$$

where $\hat{J}_P$ is assumed to be invertible, whereas K and Q are symmetric positive definite gain matrices. Closing the loop with (4), the error dynamics (3) becomes

$$\dot{e} = -Ke - Y_P\tilde{\pi}. \quad (6)$$

From (6) and (5) it can be seen that $e = \tilde{\dot{x}} = 0$ is a positively invariant set for the controlled system. To prove the stability of this set, we consider the Lyapunov-like function candidate

$$V = \frac{1}{2}e^\top e + \frac{1}{2}\tilde{\pi}^\top Q^{-1}\tilde{\pi}. \quad (7)$$

Differentiating twice (7) with respect to time along the trajectories (4)–(5) we obtain

$$\dot{V} = -e^\top Ke, \quad \ddot{V} = 2e^\top K^2 e + 2e^\top KY_P\tilde{\pi}. \quad (8)$$

We are now in the position to conclude for the stability of the controlled system.

Theorem 1. *Let us assume the approximated Jacobian matrix $\hat{J}_P$ to be invertible and the structure of the robot to be such that the boundedness of x implies the boundedness of $\hat{J}_P$, for a finite $\hat{\pi}$. The control law* (4) *and the parameter update law* (5) *for the kinematic model* (3) *and the parameter dynamics* (5) *result in global asymptotic convergence of the position error (i.e. $\lim_{t\to\infty} e = 0$).*

Proof. Since V is lower bounded and $\dot{V} \le 0$, e and $\tilde{\pi}$ are bounded vectors. Being x_d a finite signal, x is bounded and, by hypothesis, so is $\hat{J}_P$. It follows that $\hat{J}_P^{-1}$ is bounded and, since $\dot{x}_d$ is limited, from (4), so is $\dot{q}$. We can now conclude for the boundedness of Y_P and, from (8), of $\ddot{V}$. Since $\ddot{V}$ is finite and $\dot{V}$ is uniformly continuous, Barabalat's lemma can be used to conclude that $\lim_{t\to\infty} \dot{V} = 0$ and the thesis with it. $\square$

Remark 2. *The convergence of the parameter estimation error $\tilde{\pi}$ can not be guaranteed. The only conclusion that can be drawn is that, in the set where $e = 0$, the error in the estimation of the EE velocity $\tilde{\dot{x}} = Y_P\tilde{\pi}$ has to be equal to zero.*

C. Toy example

In order to clarify the behavior of the proposed controller, we apply it to a toy example: a 2D single-joint robot that tracks a desired position $x_d(t) = 0.5\cos(1\frac{\text{rad}}{\text{s}}t)$ m in the horizontal axis. The only unknown parameter is the joint type r. In this trivial case, all the ingredients necessary to synthesize the controller can be shown explicitly.

The horizontal position of the manipulator, exploiting (2), is parameterized as

$$x = (1-r)(q+d_0) + r d_0 \mathrm{c}_q$$

where $d_0 = 1$ m is the length offset, whereas the angle offset with respect to the horizontal axis is supposed to be $\theta_0 = 0$ rad. Coherently with this parametrization, it results that if $r = 0$ (P joint) we have $x = d_0 + q$, whereas if $r = 1$ (R joint) we have $x = d_0\mathrm{c}_q$.

Computing the horizontal velocity of the EE, we derive

$$\begin{aligned} J_P(q,r) &= 1 - r(1 + d_0\mathrm{s}_q), \\ Y_P(q,\dot{q}) &= \begin{bmatrix}\dot{q} & -\dot{q}(1+d_0\mathrm{s}_q)\end{bmatrix}, \end{aligned}$$

with $\pi = \begin{bmatrix}1 & r\end{bmatrix}^\top$. Since the first entry in π is constant, we estimate only its second component considering $\hat{\pi} = \begin{bmatrix}1 & \hat{r}\end{bmatrix}^\top$, with $\hat{r}$ estimate of the joint type. The error in the estimation of the EE velocity is $\tilde{\dot{x}} = Y_P\tilde{\pi} = -\dot{q}(1+d_0\mathrm{s}_q)\tilde{r}$, with $\tilde{r} = r - \hat{r}$, which shows that, in this particular case, the convergence of $\tilde{\dot{x}}$ to zero implies the convergence of $\hat{r}$ to r.

The control law (4) and the parameter update law (5) are

$$\begin{aligned} \dot{q}(q,e,\hat{r},t) &= \frac{\dot{x}_d + Ke}{1 - \hat{r}(1+d_0\mathrm{s}_q)}, \\ \dot{\hat{r}}(q,e,\hat{r},t) &= Q\frac{\dot{x}_d + Ke}{1 - \hat{r}(1+d_0\mathrm{s}_q)}(1+d_0\mathrm{s}_q)e. \end{aligned}$$

In Figure 2 we present the results of a simulation performed supposing the real robot to have a revolute joint. The estimate parameter $\hat{r}$ is initialized at 0, pretending that the controlled robot has a prismatic joint. Consistently, the initial configuration is $q(0) = -0.5$ m, which is the configuration such that the EE of the P robot is in the position $x_d(0)$. The control gains are $K = 0.1$ and $Q = 10$. The tracking error decreases to 0 in about 60 s, and the estimation of the joint type converges in approximately 40 s.

D. Simulation of a 3-DoF Manipulator

In this section we test the controller proposed in Section IV-B on a much more challenging problem: a 3-DoF 3D fully-actuated robot with all the DH parameters and the joint types unknown. The only information available is the number of joints of the manipulator.

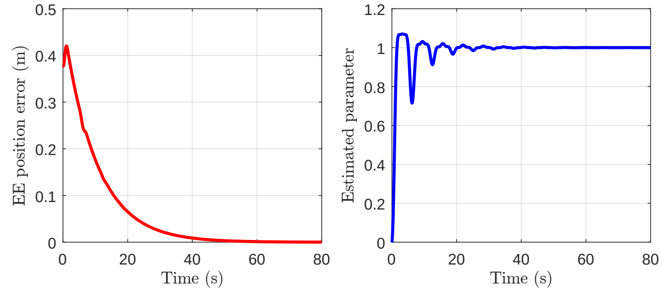

Fig. 2. Position error of the end-effector and estimated parameter as functions of time.

The EE is required to track the desired position $x_d(t) = \begin{bmatrix}0 & 1 & 0.5 + 0.05\sin(2\frac{\text{rad}}{\text{s}}t)\end{bmatrix}^\top$ m. The application of Algorithm 1 results in a kinematic model with $n_\pi = 76$. The estimates $\hat{\pi}(t)$ of these parameters are initialized with arbitrary values, in particular, the ones of a PRP manipulator (Figure 3) with given link lengths and joint offsets (Table II). The initial configuration, $q(0) = \begin{bmatrix}0.5 & 0 & 0.5\end{bmatrix}^\top$ rad, is such that the EE of the PRP robot is in the position $x_d(0)$. We choose the real robot to be a RRR manipulator (Figure 3) with the DH parameters listed in Table II. The controller gains are $K = 6I_3$ and $Q = 0.005 I_{n_\pi}$.

Figure 4 shows the position error of the EE as a function of time. Although the initial configuration is such that the EE is in a completely wrong position, the desired trajectory is reached in less then 10 s. Figure 5 shows the joint velocities, and Figure 6 depicts the error in the estimation of the velocity of the EE or, equivalently, the projection of the parameter error through the regressor matrix. In conclusion, Figure 7 represents the parameter estimation error ($|\tilde{\pi}_i|$, $i = 1,\dots,n_\pi$) at the initial time (i.e. the absolute value of the difference between the parameters of the RRR maipulator and the PRP one) and at $t = 10$ s. It can be noticed that the

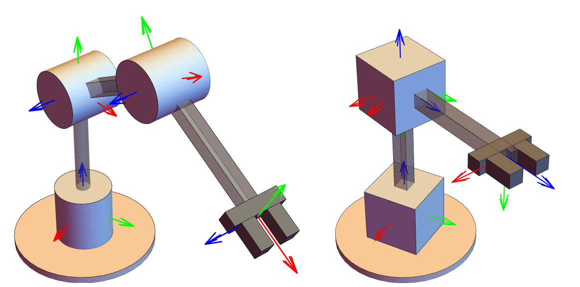

Fig. 3. RRR manipulator (real robot) and PRP manipulator (initialization of the kinematic model).

TABLE II
DH TABLES OF THE RRR MANIPULATOR AND THE PRP MANIPULATOR

	RRR manipulator				PRP manipulator			
Joint num.	$d_{0,i}$ (m)	a_i (m)	$\theta_{0,i}$ (rad)	α_i (rad)	$d_{0,i}$ (m)	a_i (m)	$\theta_{0,i}$ (rad)	α_i (rad)
1	1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0	0	0	0
2	0	1	0	0	0	0	0	$-\frac{\pi}{2}$
3	0	1	0	0	0.5	0	0	0

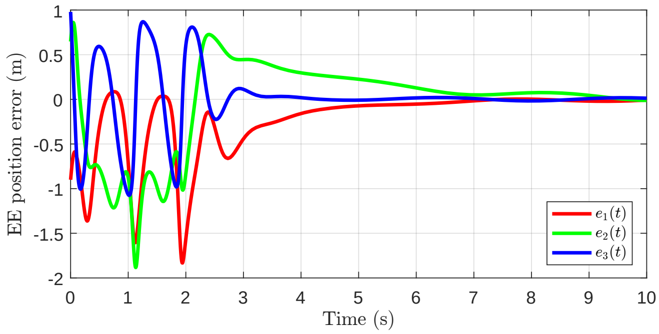

Fig. 4. Position error of the end-effector as a function of time.

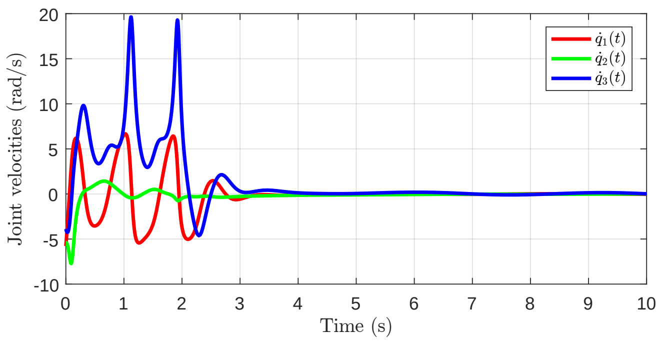

Fig. 5. Joint velocities as functions of time.

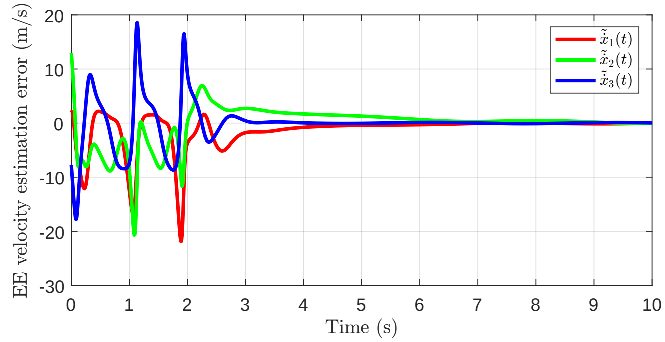

Fig. 6. Estimation error of the end-effector velocity as a function of time.

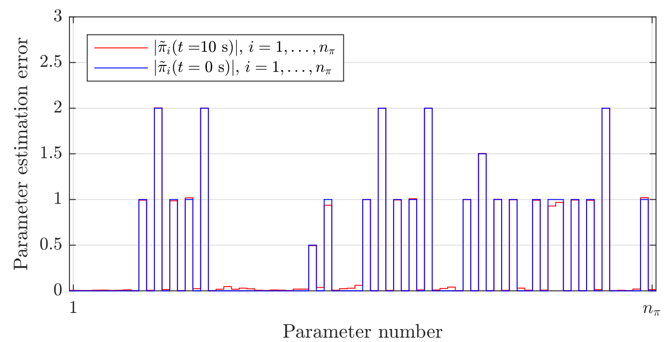

Fig. 7. Parameter estimation error at intial and final time of the simulation.

estimated parameters remains almost unchanged.

V. Conclusions and Future Prospects

In this paper we analyzed the problem of determining the minimum amount of information necessary to control a robot manipulator within the framework of adaptive control. We demonstrated that the kinematic trajectory tracking problem can be accomplished only knowing the number of joints of the manipulator, validating the algorithm in simulation on a 3-DoF robot.

The adaptive control law we propose is based on the customarily-used Barbalat's lemma; object of future works is the exploitation of tools tailored for the stability analysis of nonautonomous systems which may lead to stronger conclusions under milder assumptions. We also aim to extend this result to the dynamic control of manipulators and to implement this technique on real robots.

References

[1] 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.

[2] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.

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

[4] C.-C. Cheah, C. Liu, and J.-J. E. Slotine, "Approximate jacobian adaptive control for robot manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3075–3080, IEEE, 2004.

[5] D. Braganza, W. E. Dixon, D. M. Dawson, and B. Xian, "Tracking control for robot manipulators with kinematic and dynamic uncertainty," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 5293–5297, IEEE, 2005.

[6] C.-C. Cheah, C. Liu, and J.-J. E. Slotine, "Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models," *IEEE transactions on automatic control*, vol. 51, no. 6, pp. 1024–1029, 2006.

[7] K. J. Åström and B. Wittenmark, *Adaptive control*. Courier Corporation, 2013.

[8] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and adaptive control design*. Wiley, 1995.

[9] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[10] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[11] W.-S. Lu and Q.-H. Meng, "Regressor formulation of robot dynamics: computation and applications," *IEEE transactions on robotics and automation*, vol. 9, no. 3, pp. 323–333, 1993.

[12] G. Garofalo, C. Ott, and A. Albu-Schäffer, "On the closed form computation of the dynamic matrices and their differentiations," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2364–2359, IEEE, 2013.

[13] M. Bridges, D. M. Dawson, and C. Abdallah, "Control of rigid-link, flexible-joint robots: a survey of backstepping approaches," *Journal of Robotic Systems*, vol. 12, no. 3, pp. 199–216, 1995.

[14] J. C. Mitchell, *Foundations for programming languages*, vol. 1. MIT press Cambridge, 1996.