



Operational space trajectory tracking control of robot manipulators endowed with a primary controller of synthetic joint velocity[☆]

Javier Moreno-Valenzuela^{*}, Luis González-Hernández

Centro de Investigación y Desarrollo de Tecnología Digital del IPN, CITEDI-IPN, Ave. del Parque 1310, Mesa de Otay, Tijuana, B.C., 22510, Mexico

ARTICLE INFO

Article history:

Received 25 May 2010

Received in revised form

2 August 2010

Accepted 4 August 2010

Available online 30 August 2010

Keywords:

Operational space

Euler's angles

Tracking control

Robot manipulator

Exponential stability

Real-time experiments

ABSTRACT

In this paper, a new control algorithm for operational space trajectory tracking control of robot arms is introduced. The new algorithm does not require velocity measurement and is based on (1) a primary controller which incorporates an algorithm to obtain synthesized velocity from joint position measurements and (2) a secondary controller which computes the desired joint acceleration and velocity required to achieve operational space motion control. The theory of singularly perturbed systems is crucial for the analysis of the closed-loop system trajectories. In addition, the practical viability of the proposed algorithm is explored through real-time experiments in a two degrees-of-freedom horizontal planar direct-drive arm.

© 2010 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

A robotic task is usually specified through the pose, i.e., the position and orientation of the robot end-effector with respect to the base frame. The operational space is defined by the Cartesian position and the orientation via Euler's angles [1]. The term task space [2] refers the case when the position and orientation of the robot end-effector are described by the Cartesian position and the unit quaternion, respectively.

Many robot pose trajectory tracking controllers require robot full-state measurements, i.e., position and velocity in the joint and operational space. The position and orientation of the robot end-effector is computed through the direct kinematics model from joint position measurements obtained. Unlike optical encoders, velocity sensors (tachometers) are not provided with optical shielding. If a robot equipped with tachometers is working in an environment with high electromagnetic interference, then velocity measurements may be contaminated by noise. A cheap way to artificially obtain velocity measurements is by the numerical derivative of the signals of position and orientation. Although numerical differentiation is a common practice to solve the problem of unmeasurable velocity, sometimes the closed-loop

stability cannot be guaranteed. This situation has motivated Lyapunov-based control designs to assure pose motion control by using synthesized velocity feedback from position measurements.

With respect to pose control of the rigid body dynamics, which is closely related to the manipulator dynamics, a passivity-based control law which uses only position measurements was introduced in [3]. The scheme proposed in that study was extended later in [4]. In [5], an adaptive controller which uses only position measurements was introduced by using the modified Rodrigues parameters.

Concerning the manipulator dynamics, as far as we know the first study on pose trajectory tracking control of robot manipulators by using synthetic velocity feedback was due to Caccavale et al. [6], where an experimental evaluation was included also. They showed how the asymptotical stabilization of the closed-loop system is achieved by using the unit quaternion to represent the end-effector orientation. In Xian et al. [7], a pose trajectory tracking controller based on only position measurements was introduced. In that scheme, the end-effector orientation is described by the unit quaternion.

Human beings perform complex assembly tasks, which require the feedback of information of the position of hands, fingers, and manipulated objects. The correct manipulation of the objects depends on the feedback of visual and tactile information. Velocity feedback is required either to predict or to plan motions in situations of coordination. Let us notice that humans do not have a direct velocity sensor, but the velocity of the objects is sensed indirectly through the visual position information (see [8] and

[☆] Work partially supported by CONACyT, and SIP-Instituto Politécnico Nacional, Mexico.

^{*} Corresponding author. Tel.: +52 664 6231344; fax: +52 664 6231388.

E-mail address: moreno@citedi.mx (J. Moreno-Valenzuela).

the references therein). Then, from the point of view of control theory, it seems to be natural to incorporate that mechanism (bio-cybernetics) in robot manipulators that perform complex tasks encoded by time-varying trajectories specified in the operational space.

We found that industrial robots are provided with a primary (inner) joint velocity loop and a secondary (outer) task-based loop [9–11]. The most common secondary loop is designed with the aim of satisfying either the motion or the force control objective. The papers [11–19], show designs of robot controllers based on a primary joint velocity and secondary task-based loops.

Lyapunov's theory has been widely used to derive control laws for motion control of robot manipulators. Roughly speaking, such a methodology consists in showing that the energy of the differential equation solution remains equal or lower than a constant. As alternative, the theory of singularly perturbed systems has been recognized as a powerful tool in the analysis and design of robot controllers. Essentially, this technique is based on analyzing the convergence of the solution of differential equation in two time-scales. A few examples of application are given as follows. In [20], singular perturbations are used to derive a tracking controller for electrically driven robot manipulators. The popular PD control of robot manipulator is modified in [21]. Specifically, a high-gain observer was proposed for the estimation of the velocity and, by means of singular perturbation analysis, the closed-loop system was studied. More recently, in the work [22], the control problem of a robot manipulator with flexures both in the links and joints was investigated by using the singular perturbation technique, while in paper [23], a two time-scale (singular perturbation-based) control design for trajectory tracking of two cooperating planar rigid robots moving a flexible beam was introduced.

The main objective of this paper is to present a new solution for the pose trajectory tracking control problem. The new controller is based on a primary joint loop of joint velocity control and a secondary loop of operational space position control. The introduced scheme incorporates synthesized velocity feedback from position measurements and is designed to facilitate the closed-loop system stability analysis by using the theory of singularly perturbed systems [24]. In particular, the proposed primary joint velocity controller follows the concept of filtering the joint positions via a stable first order system to obtain a synthetic version of the joint velocity. This idea has been used in [25] to solve the velocity control of direct-current motors.

Real-time experiments in a horizontal two degrees-of-freedom direct-drive arm prototype have been carried out. The performance of the joint space inverse dynamics controller [26,27], is compared with respect to one of the new operational space control algorithms. The requested task in both implementations is to trace a circular contour in the Cartesian space of the robot.

This paper is organized as follows: In Section 2 the robot kinematics and dynamics is discussed. In Section 3 the new motion controller is described. A detailed experimental study on a two degrees-of-freedom direct-drive robot is described in Section 4. Finally, some concluding remarks are drawn in Section 5.

Notation: Throughout this paper the following notation will be adopted. $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$, $\mathbf{x} \in \mathbb{R}^n$, stands for the Euclidean norm. $\lambda_{\min}\{A(\mathbf{x})\}$ and $\lambda_{\max}\{A(\mathbf{x})\}$ denote the minimum and maximum eigenvalues of a symmetric positive definite matrix $A(\mathbf{x}) \in \mathbb{R}^{n \times n}$ for all $\mathbf{x} \in \mathbb{R}^n$, respectively. $\|B(\mathbf{x})\| = \sqrt{\lambda_{\max}\{B(\mathbf{x})^T B(\mathbf{x})\}}$ stands for the induced norm of a matrix $B(\mathbf{x}) \in \mathbb{R}^{m \times n}$ for all $\mathbf{x} \in \mathbb{R}^n$. The symbol B_r denotes the set given by the ball $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$.

2. Robot modeling and control goal

2.1. Robot dynamics

The dynamics in joint space of a serial-chain n -link robot manipulator considering the presence of friction at the robot joints can be written as [26,27],

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + F_v \dot{\mathbf{q}} + \mathbf{f}_{Cl}(\dot{\mathbf{q}}) = \boldsymbol{\tau}, \quad (1)$$

where \mathbf{q} is the $n \times 1$ vector of joint displacements, $\dot{\mathbf{q}}$ is the $n \times 1$ vector of joint velocities, $\boldsymbol{\tau}$ is the $n \times 1$ vector of applied torque inputs, $M(\mathbf{q})$ is the $n \times n$ symmetric positive definite manipulator inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ is the $n \times 1$ vector of centripetal and Coriolis torques, $\mathbf{g}(\mathbf{q})$ is the $n \times 1$ vector of gravitational torques, F_v is a $n \times n$ diagonal positive definite matrix which contains the viscous friction coefficients of each joint, and $\mathbf{f}_{Cl}(\dot{\mathbf{q}})$ is a continuous and uniformly bounded function, which approaches the behavior of the Coulomb friction.

Based on the assumption that matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ is expressed in terms of the Christoffel symbols, the property

$$C(\mathbf{x}, \mathbf{y})\mathbf{z} = C(\mathbf{x}, \mathbf{z})\mathbf{y}, \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n, \quad (2)$$

is satisfied [27].

2.2. Robot kinematics

Denoting $\mathbf{h}(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ the direct kinematics map, the position and orientation $\mathbf{y} \in \mathbb{R}^m$ of the end-effector is given by

$$\mathbf{y} = \mathbf{h}(\mathbf{q}). \quad (3)$$

A common physical interpretation is the case

$$\mathbf{y} = \begin{bmatrix} \mathbf{p}(\mathbf{q}) \\ \boldsymbol{\phi}(\mathbf{q}) \end{bmatrix}$$

where $\mathbf{p} \in \mathbb{R}^3$ denotes the end-effector position in the three-dimensional Cartesian space and $\boldsymbol{\phi} = [\varphi \ \vartheta \ \psi]^T \in \mathbb{R}^3$ is the set of Euler's angles which describes end-effector orientation. Let us notice that Euler's angles can be extracted from a given rotation matrix R describing the orientation of the end-effector frame by using the closed-loop inversion formula [26].

The time derivative of the direct kinematic model (3) yields the differential kinematic model

$$\dot{\mathbf{y}} = \frac{d}{dt} \mathbf{h}(\mathbf{q}) = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \dot{\mathbf{q}} = J(\mathbf{q})\dot{\mathbf{q}} \quad (4)$$

where $J(\mathbf{q})$ is the so-called analytical Jacobian matrix [26,28]. The robot Jacobian describes a map from velocities in joint space to velocities in operational space. The Jacobian right pseudo-inverse [28], is given by

$$J(\mathbf{q})^\dagger = J(\mathbf{q})^T [J(\mathbf{q})J(\mathbf{q})^T]^{-1},$$

assuming that $J(\mathbf{q})J(\mathbf{q})^T$ is nonsingular.

Assumption 1. The analytical Jacobian $J(\mathbf{q})$ is assumed of full-rank (rank = m) and bounded by $k_j > 0$, i.e.

$$\|J(\mathbf{q})\| \leq k_j \quad \forall \mathbf{q} \in \mathbb{R}^n. \quad (5)$$

At the same time, it is also assumed that

$$\|J(\mathbf{q})^\dagger\| \leq k_j^\dagger \quad \forall \mathbf{q} \in \mathbb{R}^n, \quad (6)$$

with $k_j^\dagger > 0$. \square

In this paper the notation

$$\dot{J}(\mathbf{q}, \dot{\mathbf{q}})^\dagger = \frac{d}{dt} [J(\mathbf{q})^\dagger]$$

is used to denote the time derivative of the Jacobian right pseudo-inverse, which is assumed to satisfy the following.

Assumption 2. Let us suppose that the joint velocity can be expressed as $\dot{\mathbf{q}} = \mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$. Then the map expressed by the time derivative of the Jacobian right pseudo-inverse satisfies the following relation

$$\dot{J}(\mathbf{q}, \dot{\mathbf{q}})^\dagger = \dot{J}(\mathbf{q}, \mathbf{x}_1 + \mathbf{x}_2)^\dagger = \dot{J}(\mathbf{q}, \mathbf{x}_1)^\dagger + \dot{J}(\mathbf{q}, \mathbf{x}_2)^\dagger. \quad \square \quad (7)$$

It is possible to show that a planar two degrees-of-freedom revolute joint robot satisfies Assumption 2.

2.3. Control goal

Once the motion specification is given in terms of the desired trajectory $\mathbf{y}_d(t)$ in the operational space, then the motion control objective in operational space is to achieve:

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{y}}(t) = \mathbf{0}, \quad (8)$$

where

$$\tilde{\mathbf{y}}(t) = \mathbf{y}_d(t) - \mathbf{y}(t) \quad (9)$$

denotes the operational space pose error.

3. Operational space controller using synthetic velocity feedback

The proposed controller responds to the industrial practice of using a primary joint velocity control loop to compute the torques and forces applied by the robot actuators, and a secondary position controller to solve the desired joint acceleration and velocity reproduced by the primary velocity control loop. As pointed out previously, the proposed scheme includes an algorithm for synthesis of the velocity feedback from joint position measurements. A block diagram of the proposed controller is shown in Fig. 1.

3.1. Synthetic velocity controller: primary loop

The task of the primary control loop is to compute the proper torques and forces $\boldsymbol{\tau} \in \mathbb{R}^n$ so that the joint velocity $\dot{\mathbf{q}}$ achieves asymptotic tracking of a desired velocity command $\boldsymbol{\omega}_d$, i.e.,

$$\lim_{t \rightarrow \infty} \tilde{\boldsymbol{\omega}}(t) = 0,$$

where

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_d(t) - \dot{\mathbf{q}}(t) \quad (10)$$

denotes the joint velocity error. In this paper we assume that the desired joint velocity $\boldsymbol{\omega}_d$ is given as function of time $t \in \mathbb{R}^+$ and the actual joint position $\mathbf{q} \in \mathbb{R}^n$, i.e., $\boldsymbol{\omega}_d(t, \mathbf{q})$. Therefore, from the joint velocity error definition (10) we can write

$$\dot{\mathbf{q}}(t) = \boldsymbol{\omega}_d(t, \mathbf{q}(t)) - \tilde{\boldsymbol{\omega}}(t), \quad (11)$$

which is interpreted as the dynamics of the joint position $\mathbf{q}(t)$, where the signal $\tilde{\boldsymbol{\omega}}(t)$ is acting as disturbance. Making sure that $\tilde{\boldsymbol{\omega}}(t)$ vanishes exponentially, and defining a suitable joint velocity signal $\boldsymbol{\omega}_d(t, \mathbf{q}(t))$, then the operational motion control objective (8) can be guaranteed. This will be shown in the discussion of the secondary loop. The first step is to show the exponential convergence of $\tilde{\boldsymbol{\omega}}(t)$.

Usually, robot manipulators are equipped only with position sensors. The reason is that sometimes velocity sensors increase the cost, limit the robot mechanical design and effective payload, and provide signals contaminated by noise (notice that sensor signals may be affected by electromagnetic interference if no proper shielding is provided). A solution adopted in practice is to differentiate numerically the position signal. However, simple numerical differential may be inadequate for low and high speeds [29].

As solution to introduce damping from joint position measurements, is to design either a filter or observer, which should match conditions derived from the stability analysis of the closed-loop system. The proposed primary joint velocity controller follows this key idea and is written as follows

$$\boldsymbol{\tau} = M(\mathbf{q})\dot{\boldsymbol{\omega}}_d^* + C(\mathbf{q}, \boldsymbol{\omega}_d)\boldsymbol{\omega}_d + \mathbf{g}(\mathbf{q}) + K_p\dot{\boldsymbol{\xi}} + K_i\boldsymbol{\xi} + F_v\boldsymbol{\omega}_d + \mathbf{f}_{Cl}(\boldsymbol{\omega}_d), \quad (12)$$

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\omega}_d - \boldsymbol{\vartheta}, \quad (13)$$

where K_p and K_i are $n \times n$ diagonal positive definite matrices, $\boldsymbol{\vartheta} \in \mathbb{R}^n$ is the output of a first order filter, $\boldsymbol{\omega}_d$ is the desired joint velocity and $\dot{\boldsymbol{\omega}}_d^*$ is a signal called pre-compensated acceleration.

The signals $\boldsymbol{\omega}_d$ and $\dot{\boldsymbol{\omega}}_d^*$ are computed by the secondary loop to be defined later. By now, we just will say that the desired joint velocity $\boldsymbol{\omega}_d$ is a function of time t and the joint position \mathbf{q} , i.e., $\boldsymbol{\omega}_d(t, \mathbf{q})$. Hence the actual desired acceleration $\dot{\boldsymbol{\omega}}_d(t, \mathbf{q})$ is given by

$$\dot{\boldsymbol{\omega}}_d(t, \mathbf{q}) = \frac{\partial \boldsymbol{\omega}_d(t, \mathbf{q})}{\partial t} + \frac{\partial \boldsymbol{\omega}_d(t, \mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}}. \quad (14)$$

Since the actual joint velocity $\dot{\mathbf{q}}$ is not available for measurement, the signal $\dot{\boldsymbol{\omega}}_d$ in (14) cannot be used in the control law. Instead, the pre-compensated acceleration is used $\dot{\boldsymbol{\omega}}_d^*$, where the symbol \star has been used to remark that it does not depend on the actual joint velocity $\dot{\mathbf{q}}$.

The signal $\boldsymbol{\vartheta}$ involved in the joint velocity controller (12)–(13) is obtained from the following first order filter

$$\dot{\boldsymbol{x}} = -\boldsymbol{\vartheta}, \quad (15)$$

$$\boldsymbol{\vartheta} = \boldsymbol{\omega}_d + A\boldsymbol{x} + A\mathbf{q}, \quad (16)$$

where $A = \text{diag}\{a_1, \dots, a_n\}$ is positive definite.

3.2. Operational space trajectory tracking controller: secondary loop

To guarantee that a task specified with desired trajectories in the operational space is achieved, an appropriate definition of the signals $\boldsymbol{\omega}_d(t)$ and $\dot{\boldsymbol{\omega}}_d^*(t)$ should be

$$\boldsymbol{\omega}_d(t, \mathbf{q}) = J(\mathbf{q})^\dagger [\dot{\mathbf{y}}_d(t) + K_o \tanh(\tilde{\mathbf{y}})], \quad (17)$$

$$\dot{\boldsymbol{\omega}}_d^* = \dot{J}(\mathbf{q}, \boldsymbol{\omega}_d)^\dagger [\dot{\mathbf{y}}_d(t) + K_o \tanh(\tilde{\mathbf{y}})] + J(\mathbf{q})^\dagger [\ddot{\mathbf{y}}_d(t)]. \quad (18)$$

where $K_o \in \mathbb{R}^{m \times m}$ is a positive definite matrix and $\tilde{\mathbf{y}}$ in (9) denotes the operational space tracking error. Note that $\mathbf{y} \in \mathbb{R}^m$ can be measured through the direct kinematics map (3). In fact, if we compute $\dot{\boldsymbol{\omega}}_d$ from (18) and remove the terms associated with the signals $\dot{\mathbf{y}}$ and $\dot{\mathbf{q}}$, then $\dot{\boldsymbol{\omega}}_d^*$ is obtained.

The proposed control method is summarized as follows: the primary control loop is given by the controller (12)–(13) and the filter (15)–(16), while the operational space secondary controller is defined by Eqs. (17) and (18). A block diagram of the new motion controller is shown in Fig. 1.

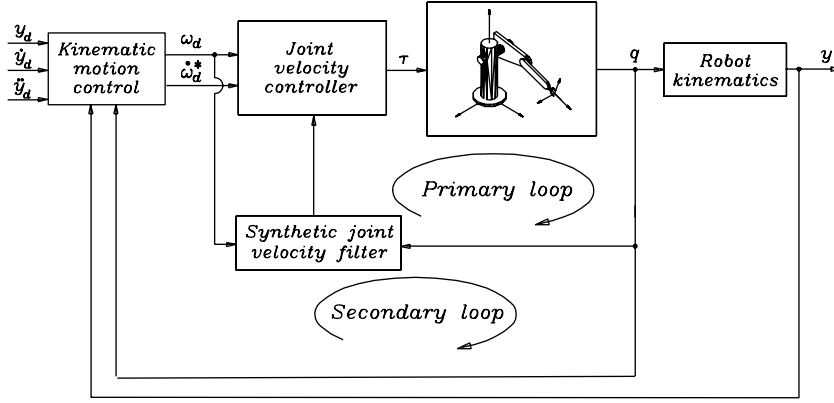


Fig. 1. Block diagram of the proposed controller based on a primary loop of velocity control and a secondary loop of operational space position control.

3.3. Closed-loop system derivation

Multiplying both sides of system (11) by $J(\mathbf{q})$ and using the definition of differential kinematics in (4), we are able to write

$$\dot{\mathbf{y}} = J(\mathbf{q}) [\boldsymbol{\omega}_d(t, \mathbf{q}) - \tilde{\boldsymbol{\omega}}]. \quad (19)$$

Let us notice that the kinematic control concept [30] considers system (19) as the robot model, with control input $\boldsymbol{\omega}_d(t, \mathbf{q})$. Substituting (17) into (19), the following error equation is obtained

$$\dot{\tilde{\mathbf{y}}} = -K_0 \tanh(\tilde{\mathbf{y}}) + J(\mathbf{q})\tilde{\boldsymbol{\omega}}. \quad (20)$$

Differentiating (16) with respect to time, and substituting (15) in the resulting expression, we can write

$$\frac{d}{dt} \boldsymbol{\vartheta} = \dot{\boldsymbol{\omega}}_d - A\boldsymbol{\vartheta} + A\dot{\mathbf{q}}. \quad (21)$$

Using the definition of the joint velocity error $\tilde{\boldsymbol{\omega}}$ in (10) and the signal $\dot{\boldsymbol{\xi}}$ in (13), it is possible to write (21) as

$$\frac{d}{dt} \dot{\boldsymbol{\xi}} = -A[\dot{\boldsymbol{\xi}} - \tilde{\boldsymbol{\omega}}]. \quad (22)$$

Eq. (22) summarizes the dynamics of the filter (15)–(16).

On the other hand, substitution of Eq. (12) in the robot equation (1), and using the robot model property (2), yield

$$M(\mathbf{q})\dot{\tilde{\boldsymbol{\omega}}} + [C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \boldsymbol{\omega}_d)]\tilde{\boldsymbol{\omega}} + K_p\dot{\boldsymbol{\xi}} + K_i\boldsymbol{\xi} + F_v\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\omega}}^T [f_{Cl}(\boldsymbol{\omega}_d) - f_{Cl}(\dot{\mathbf{q}})] - M(\mathbf{q})\boldsymbol{\eta}(t, \mathbf{q}, \tilde{\boldsymbol{\omega}}) = \mathbf{0}. \quad (23)$$

Eqs. (20), (22) and (23) represent the closed-loop dynamics, which can be written as

$$\frac{d}{dt} \tilde{\mathbf{y}} = -K_0 \tanh(\tilde{\mathbf{y}}) + J(\mathbf{q})\tilde{\boldsymbol{\omega}}, \quad (24)$$

$$\frac{d}{dt} \begin{bmatrix} \dot{\boldsymbol{\xi}} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\xi}} \\ -A\dot{\boldsymbol{\xi}} + A\tilde{\boldsymbol{\omega}} \\ -M(\mathbf{q})^{-1}\zeta_1(t, \mathbf{q}, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}, \tilde{\boldsymbol{\omega}}) + \boldsymbol{\eta}(t, \mathbf{q}, \tilde{\boldsymbol{\omega}}) \end{bmatrix}, \quad (25)$$

where

$$\zeta_1(t, \mathbf{q}, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}, \tilde{\boldsymbol{\omega}}) = [C(\mathbf{q}, \boldsymbol{\omega}_d - \tilde{\boldsymbol{\omega}}) + C(\mathbf{q}, \boldsymbol{\omega}_d)]\tilde{\boldsymbol{\omega}} + K_p\dot{\boldsymbol{\xi}} + K_i\boldsymbol{\xi} + F_v\tilde{\boldsymbol{\omega}} + f_{Cl}(\boldsymbol{\omega}_d) - f_{Cl}(\dot{\mathbf{q}}),$$

and $\boldsymbol{\omega}_d(t, \mathbf{q})$ is the desired velocity provided by the secondary loop in (17), and

$$\begin{aligned} \boldsymbol{\eta}(t, \mathbf{q}, \tilde{\boldsymbol{\omega}}) &= \dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{\omega}}_d^* \\ &= -\dot{J}(\mathbf{q}, \tilde{\boldsymbol{\omega}})^\dagger [\dot{\mathbf{y}}_d(t) + K_0 \tanh(\tilde{\mathbf{y}})] \\ &\quad - J(\mathbf{q})^\dagger K_0 \text{Sech}^2(\tilde{\mathbf{y}}) J(\mathbf{q})\tilde{\boldsymbol{\omega}}, \end{aligned} \quad (26)$$

which is obtained by using assumption (7) with $\mathbf{x}_1 = \boldsymbol{\omega}_d$ and $\mathbf{x}_2 = -\tilde{\boldsymbol{\omega}}$, and Eq. (19).

The state-space origin $[\tilde{\mathbf{y}}^T \dot{\boldsymbol{\xi}}^T \tilde{\boldsymbol{\omega}}^T]^T = \mathbf{0} \in \mathbb{R}^{m+3n}$ is the only equilibrium point of the closed-loop system (24)–(25). This can be shown by observing from the expression of $\boldsymbol{\eta}(t, \mathbf{q}, \tilde{\boldsymbol{\omega}})$ in (26) that if $\tilde{\boldsymbol{\omega}} = \mathbf{0}$ then $\boldsymbol{\eta}(t, \mathbf{q}, \mathbf{0}) = \mathbf{0}$.

3.4. Stability analysis

In order to analyze the stability property of the state-space origin of the closed-loop (24)–(25), we need to parametrize the gains of the primary controller (12)–(13) and (15)–(16) as follows:

$$K_p = \bar{K}_p/\epsilon, \quad (27)$$

$$K_i = \bar{K}_i/\epsilon, \quad (28)$$

$$A = \bar{A}/\epsilon, \quad (29)$$

where ϵ is a strictly positive constant which is denoted as perturbation parameter. Thus, the closed-loop system (24)–(25) can be rewritten as

$$\frac{d}{dt} \tilde{\mathbf{y}} = -K_0 \tanh(\tilde{\mathbf{y}}) + J(\mathbf{q})\tilde{\boldsymbol{\omega}}, \quad (30)$$

$$\frac{d}{dt} \dot{\boldsymbol{\xi}} = \dot{\boldsymbol{\xi}}, \quad (31)$$

$$\epsilon \frac{d}{dt} \begin{bmatrix} \dot{\boldsymbol{\xi}} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -\bar{A}\dot{\boldsymbol{\xi}} + \bar{A}\tilde{\boldsymbol{\omega}} \\ -\epsilon [M(\mathbf{q})^{-1}\zeta_2(t, \mathbf{q}, \tilde{\boldsymbol{\omega}}) + \boldsymbol{\eta}(t, \mathbf{q}, \tilde{\boldsymbol{\omega}})] - M(\mathbf{q})^{-1} [\bar{K}_p\dot{\boldsymbol{\xi}} + \bar{K}_i\boldsymbol{\xi}] \end{bmatrix}, \quad (32)$$

where

$$\zeta_2(t, \mathbf{q}, \tilde{\boldsymbol{\omega}}) = [C(\mathbf{q}, \boldsymbol{\omega}_d - \tilde{\boldsymbol{\omega}}) + C(\mathbf{q}, \boldsymbol{\omega}_d)]\tilde{\boldsymbol{\omega}} + F_v\tilde{\boldsymbol{\omega}} + f_{Cl}(\boldsymbol{\omega}_d) - f_{Cl}(\dot{\mathbf{q}}).$$

Note that the closed-loop system (30)–(32) has the standard form of a singularly perturbed system [24]. A singularly perturbed system can be decoupled in two subsystems: a slow dynamics subsystem and a fast dynamics subsystem. Under certain conditions, the exponential stability of each subsystem implies that the overall system is exponentially stable. This claim has been stated formally in Theorem 9.3 of [24], which is reproduced in Appendix as Theorem 1.

As shown later, to facilitate the stability analysis, the standard form of a singularly perturbed system has been conveniently derived through the parametrization of the controller gains (27)–(29). The slow states correspond to $[\tilde{\mathbf{y}}^T \dot{\boldsymbol{\xi}}^T]^T$ and the fast ones to

$[\dot{\xi}^T \ \tilde{\omega}^T]^T$. It is possible to show that if $\dot{\xi}(t)$ converges fast to zero, it means that $\vartheta(t)$ converges rapidly to actual joint velocity \dot{q} , while fast convergence of the joint velocity error $\tilde{\omega}(t)$ means that the kinematic error dynamics (30) becomes more stable since the perturbing term $J(\mathbf{q})\tilde{\omega}$ is less dominant.

To prove that the controller (12)–(13), (15)–(16) and (17)–(18) guarantees the control objective (8), we show that there exists $\epsilon^* > 0$ such that the state-space origin of the closed-loop system (30)–(32) is locally exponentially stable with $\epsilon^* > \epsilon > 0$. Such a proof is obtained by verifying the conditions of Theorem 1 in Appendix, which will be set down in the following four items:

- (1) The closed-loop system (30)–(32) has a unique equilibrium point at

$$\begin{bmatrix} \tilde{\mathbf{y}} \\ \xi \\ \dot{\xi} \\ \tilde{\omega} \end{bmatrix} = \mathbf{0}.$$

- (2) One important step in the application of Theorem 1 (see Appendix) is to verify that subsystem (32) has isolated roots of the form

$$\mathbf{z} = \mathbf{z}^* = \mathbf{h}(t, \mathbf{x}), \quad (33)$$

with $\mathbf{x} = [\tilde{\mathbf{y}}^T \ \xi^T]^T$ and $\mathbf{z} = [\dot{\xi}^T \ \tilde{\omega}^T]^T$, for $\epsilon = 0$. Eq. (33) is called *quasi-steady-state* solution [24]. Substituting $\epsilon = 0$ into (32) we have that any isolated root $\mathbf{z}^* \in \mathbb{R}^{2n}$ satisfies

$$\dot{\xi}^* = -\bar{K}_p^{-1} \bar{K}_i \xi, \quad (34)$$

$$\tilde{\omega}^* = \dot{\xi}^* = -\bar{K}_p^{-1} \bar{K}_i \xi. \quad (35)$$

- (3) The desired joint velocity $\omega_d(t, \mathbf{q})$ is bounded globally in the sense that, for all $t \geq 0$ and $\mathbf{q} \in \mathbb{R}^n$,

$$\|\omega_d(t, \mathbf{q})\|_M \leq k_j^\dagger [\|\dot{\mathbf{y}}_d\|_M + \lambda_{\max}\{K_o\}\sqrt{n}],$$

where $\|\dot{\mathbf{y}}_d\|_M \geq \|\mathbf{y}_d(t)\|$ for all $t \geq 0$. Thus, system (30)–(32) and the isolated roots (34)–(35) have bounded partial derivatives in compact sets.

- (4) The slow dynamics is obtained by substituting Eqs. (34)–(35) into the closed-loop equations (30)–(31), i.e.,

$$\frac{d}{dt} \tilde{\mathbf{y}} = -K_o \tanh(\tilde{\mathbf{y}}) - J(\mathbf{q}) \bar{K}_p^{-1} \bar{K}_i \xi, \quad (36)$$

$$\frac{d}{dt} \xi = -\bar{K}_p^{-1} \bar{K}_i \xi. \quad (37)$$

Let us consider the Lyapunov function

$$V(\tilde{\mathbf{y}}, \xi) = \alpha \sum_{i=1}^m \ln(\cosh(\tilde{y}_i)) + \frac{1}{2} \xi^T \xi, \quad \alpha > 0, \quad (38)$$

whose time derivative along the trajectories of the slow dynamics (36)–(37) achieves

$$\begin{aligned} \dot{V}(\tilde{\mathbf{y}}, \xi) &= -\alpha \tanh(\tilde{\mathbf{y}})^T K_o \tanh(\tilde{\mathbf{y}}) \\ &\quad - \alpha \tanh(\tilde{\mathbf{y}})^T J(\mathbf{q}) \bar{K}_p^{-1} \bar{K}_i \xi - \xi^T \bar{K}_p^{-1} \bar{K}_i \xi, \end{aligned}$$

where the property $\frac{d}{dx} \ln(\cosh(x)) = \tanh(x)\dot{x}$, $x \in \mathbb{R}$, was used. By using the fact that there always exists a scalar $\delta > 0$ small enough such that $\|\tanh(\mathbf{x})\| \geq \delta \|\mathbf{x}\|$ for all $\|\mathbf{x}\| \leq r$, with r arbitrarily large, and property (5), an upper bound on $\dot{V}(\tilde{\mathbf{y}}, \xi)$ can be obtained as follows

$$\begin{aligned} \dot{V}(\tilde{\mathbf{y}}, \xi) &\leq - \left[\frac{\|\tilde{\mathbf{y}}\|}{\|\xi\|} \right]^T \\ &\quad \times \begin{bmatrix} \alpha \delta^2 \lambda_{\min}\{K_o\} & -\frac{\alpha}{2} k_j \lambda_{\max}\{\bar{K}_p^{-1} \bar{K}_i\} \\ -\frac{\alpha}{2} k_j \lambda_{\max}\{\bar{K}_p^{-1} \bar{K}_i\} & \lambda_{\min}\{\bar{K}_p^{-1} \bar{K}_i\} \end{bmatrix} \begin{bmatrix} \|\tilde{\mathbf{y}}\| \\ \|\xi\| \end{bmatrix}. \quad (39) \end{aligned}$$

Now, by Sylvester's Theorem, the right-hand side of inequality (39) is globally negative definite for small enough $\alpha > 0$. Finally, by using the property $\|\tilde{\mathbf{y}}\|^2 \geq \sum_{i=1}^n \ln(\cosh(\tilde{y}_i))$ and a relation between (38) and (39), it is possible to prove that there exists small enough $\kappa > 0$ such that the inequality

$$\dot{V}(\tilde{\mathbf{y}}, \xi) \leq -\kappa V(\tilde{\mathbf{y}}, \xi)$$

is satisfied for all $[\tilde{\mathbf{y}}^T \ \xi^T]^T \in B_r$. Therefore, the state-space origin of the slow dynamics (36)–(37) is exponentially stable. In other words,

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\mathbf{y}}(t) \\ \xi(t) \end{bmatrix} = \mathbf{0}$$

with exponential convergence rate as time t increases.

- (5) To obtain the boundary-layer system, the change of variable

$$\mathbf{p} = \mathbf{z} - \mathbf{h}(t, \mathbf{x}),$$

is defined. Let us remember that $\mathbf{x} = [\tilde{\mathbf{y}}^T \ \xi^T]^T$ and $\mathbf{z} = [\dot{\xi}^T \ \tilde{\omega}^T]^T$, and $\mathbf{z}^* = \mathbf{h}(t, \mathbf{x})$ is given in (34)–(35).

By deriving \mathbf{p} with respect to $\sigma = t/\epsilon$ and setting $\epsilon = 0$, we obtain

$$\frac{d}{d\sigma} \mathbf{p}_1 = -\bar{A}[\mathbf{p}_1 - \mathbf{p}_2], \quad (40)$$

$$\frac{d}{d\sigma} \mathbf{p}_2 = -M^*(\mathbf{y}_d - \tilde{\mathbf{y}}) - \frac{1}{2} \bar{K}_p \mathbf{p}_2, \quad (41)$$

where $M^*(\mathbf{y}_d - \tilde{\mathbf{y}}) = M(\mathbf{h}^{-1}(\mathbf{y}_d - \tilde{\mathbf{y}})) \in \mathbb{R}^{n \times n}$. The vectors \mathbf{y}_d and $\tilde{\mathbf{y}}$ are interpreted as fixed parameters. The Lyapunov function for system (40)–(41) is defined as

$$W(\mathbf{p}_1, \mathbf{p}_2) = \beta \frac{1}{2} \mathbf{p}_1^T \mathbf{p}_1 + \frac{1}{2} \mathbf{p}_2^T M^*(\tilde{\mathbf{y}}) \mathbf{p}_2, \quad \text{with } \beta > 0,$$

whose time derivative achieves

$$\dot{W}(\mathbf{p}_1, \mathbf{p}_2) = - \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}^T \begin{bmatrix} \beta \bar{A} & -\frac{1}{2} \beta \bar{A} \\ -\frac{1}{2} \beta \bar{A} & \bar{K}_p \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}.$$

By the Schur complement condition for positive definiteness, $\bar{K}_p - \frac{1}{4} \beta^2 \bar{A} > 0$ implies that $\dot{W}(\mathbf{p}_1, \mathbf{p}_2)$ is globally negative definite. Such a condition is satisfied for small $\beta > 0$ and any \bar{K}_p and \bar{A} positive definite.

Therefore, and because (40)–(41) is a linear system, we have sufficient conditions to claim that $[\mathbf{p}_1(\sigma)^T \ \mathbf{p}_2(\sigma)^T]^T \rightarrow \mathbf{0}$ with exponential rate as the scaled time $\sigma \rightarrow \infty$.

By Theorem 1 in Appendix, there are sufficient conditions to claim that there exists ϵ^* such that $\epsilon^* > \epsilon > 0$ guarantees that there is a compact set of initial conditions R_A where the trajectories of the closed-loop system (30)–(32) achieves the following limit

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix} = \lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\mathbf{y}}(t) \\ \xi(t) \\ \dot{\xi}(t) \\ \tilde{\omega}(t) \end{bmatrix} = \mathbf{0},$$

whereby the proposed control algorithm described in Eqs. (12)–(13), (15)–(16) and (17)–(18), guarantees the operational space motion control objective (8).

4. Experimental results

A planar two degrees-of-freedom direct-drive arm has been used to test the task of drawing a circle in the Cartesian space. See Fig. 2 for a CAD drawing and picture. The system is composed of two DC *Pittman* motors operated in current mode with two *Advanced Motion Controls* servo amplifiers. A *Sensoray 626* I/O card

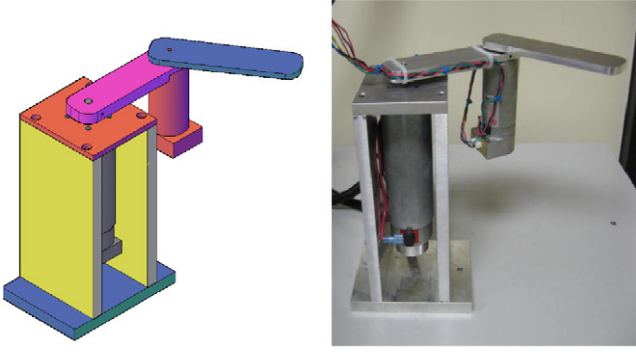


Fig. 2. Experimental robot manipulator actuated by direct-current motors.

is used to read encoder signals with quadrature included and to transfer control commands through the D/A channels. A PC running Windows XP, Matlab, Simulink and Real-Time Windows Target is used to execute controllers in real time with a 1 kHz sampling rate.

4.1. Practical considerations

The experimental tests are concerned with the robot pose \mathbf{y} . The direct kinematics is given by

$$\mathbf{h}(\mathbf{q}) = \begin{bmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\ -l_2 \cos(q_1) - l_2 \cos(q_1 + q_2) \end{bmatrix},$$

where $l_1 = l_2 = 0.15$ [m] are the link length. Thus, the corresponding Jacobian matrix results on

$$J(\mathbf{q}) = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ l_2 \sin(q_1) + l_2 \sin(q_1 + q_2) & l_2 \sin(q_1 + q_2) \end{bmatrix}.$$

In the experimental robot system, $l_1 = 0.115$ [m] and $l_2 = 0.130$ [m].

In particular, the experimental robot is equipped with direct-current motor actuators operated in current mode. Hence the torque delivered is given by

$$\boldsymbol{\tau} = K\mathbf{u}, \quad (42)$$

with $K = \text{diag}\{k_1, k_2\}$ [Nm/V] a positive definite matrix that contains the motor constants and $\mathbf{u} \in \mathbb{R}^2$ is the applied control voltage, whereby the DC motor actuators can be modeled as ideal torque sources.

By using (42), the robot model (1) can be written as

$$\overline{M}(\mathbf{q})\ddot{\mathbf{q}} + \overline{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \overline{F}_v\dot{\mathbf{q}} + \overline{f}_{cl}(\dot{\mathbf{q}}) + \overline{\mathbf{g}}(\mathbf{q}) = \mathbf{u}. \quad (43)$$

where the bar symbol denotes left multiplication by K^{-1} . Specifically, the elements of the experimental robot model (43) are

$$\overline{M}(\mathbf{q}) = \begin{bmatrix} \theta_1 + 2\theta_2 \cos(q_2) & \theta_3 + \theta_2 \cos(q_2) \\ \theta_4 + \theta_5 \cos(q_2) & \theta_6 \end{bmatrix} \quad (44)$$

$$\overline{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\theta_2 \sin(q_2)\dot{q}_2 & -\theta_2 \sin(q_2)[\dot{q}_1 + \dot{q}_2] \\ \theta_5 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \quad (45)$$

$$\overline{F}_v = \text{diag}\{\theta_7, \theta_8\}, \quad (46)$$

$$\overline{f}_{cl}(\dot{\mathbf{q}}) = \begin{bmatrix} \overline{f}_{cl1}(\dot{q}_1) \\ \overline{f}_{cl2}(\dot{q}_2) \end{bmatrix} = \begin{cases} \theta_9 \tanh(50\dot{q}_1) & \text{if } \dot{q}_1 \geq 0, \\ \theta_{10} \tanh(50\dot{q}_1) & \text{if } \dot{q}_1 < 0, \\ \theta_{11} \tanh(50\dot{q}_2) & \text{if } \dot{q}_2 \geq 0, \\ \theta_{12} \tanh(50\dot{q}_2) & \text{if } \dot{q}_2 < 0, \end{cases} \quad (47)$$

$$\overline{\mathbf{g}}(\mathbf{q}) = \mathbf{0}. \quad (48)$$

The parameters $\theta_i \in \mathbb{R}$ are the lumped equivalent parameters of the robot which include the characteristics of the motor constants.

Table 1

Estimated parameters of the experimental robot arm; see Eqs. (44)–(47) for reference.

| Parameter | Value | Unit | Parameter | Value | Unit |
|------------|--------|----------|---------------|--------|---------|
| θ_1 | 0.0480 | kg m V/N | θ_7 | 0.0070 | V s/rad |
| θ_2 | 0.0037 | kg m V/N | θ_8 | 0.0071 | V s/rad |
| θ_3 | 0.0033 | kg m V/N | θ_9 | 0.0571 | V |
| θ_4 | 0.0161 | kg m V/N | θ_{10} | 0.0067 | V |
| θ_5 | 0.0220 | kg m V/N | θ_{11} | 0.0554 | V |
| θ_6 | 0.0162 | kg m V/N | θ_{12} | 0.0105 | V |

By using an adequate estimation of such a parameter, many model-based control laws can be implemented in the form of voltage input \mathbf{u} , including the control algorithm introduced in this paper. Table 1 shows a numerical estimation of the parameters $\theta_i \in \mathbb{R}$, which was obtained by using the weighted least squares identification method [31].

4.2. Desired trajectory

The task consists in tracing a circle of radius r_0 centered at coordinate \mathbf{y}_c of the operational configuration space at a given speed v_0 . To this end, the following desired Cartesian position trajectory $\mathbf{y}_d(t)$ was specified to be used during the experiments

$$\mathbf{y}_d(t) = \begin{bmatrix} y_{d1c} + r_0 \sin\left(\frac{v_0}{r_0}t\right) \\ y_{d2c} + r_0 \cos\left(\frac{v_0}{r_0}t\right) \end{bmatrix} [\text{m}], \quad (49)$$

with parameters $y_{d1c} = 0.1061$ [m], $y_{d2c} = 0.1061$ [m], $r_0 = 0.05$ [m] and $v_0 = 0.30$ [m/s]. The robot initial conditions were $\mathbf{q}(0) = [45 \ 90]^T$ [degrees].

The control objective is that the arm tip tracks the position specified by (49). So, the requested task of tracing a circle with constant tangent speed will be attained.

On the other hand, a joint space trajectory tracking controller can be used to solve the requested task as alternative to operational space control. Thus, it is necessary to translate the task space desired trajectory $\mathbf{y}_d(t)$ in (49) to a suitable joint space desired position trajectory $\mathbf{q}_d(t)$. With this aim, we use the inverse kinematics of the experimental two degrees-of-freedom arm. It can be shown that the desired joint position trajectory \mathbf{q}_d that encodes the requested task is given by

$$\mathbf{q}_d = \mathbf{h}^{-1}(\mathbf{y}_d) = \begin{bmatrix} \frac{\pi}{2} + \tan^{-1}\left(\frac{y_{d2}}{y_{d1}}\right) - \tan^{-1}\left(\frac{l_2 \sin(q_{d2})}{l_1 + l_2 \cos(q_{d2})}\right) \\ \cos^{-1}\left(\frac{y_{d1}^2 + y_{d2}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \end{bmatrix}, \quad (50)$$

where $\mathbf{h}^{-1}(\mathbf{y}_d)$ denotes the inverse kinematics as function of the desired task space trajectory \mathbf{y}_d in (49).

The desired joint velocity and acceleration are given as

$$\dot{\mathbf{q}}_d = J(\mathbf{q}_d)^{-1} \begin{bmatrix} \dot{y}_{d1} \\ \dot{y}_{d2} \end{bmatrix}, \quad (51)$$

$$\ddot{\mathbf{q}}_d = \left[\frac{d}{dt} J(\mathbf{q}_d)^{-1} \right] \begin{bmatrix} \dot{y}_{d1} \\ \dot{y}_{d2} \end{bmatrix} + J(\mathbf{q}_d)^{-1} \begin{bmatrix} \ddot{y}_{d1} \\ \ddot{y}_{d2} \end{bmatrix}. \quad (52)$$

Eqs. (51) and (52) remain valid as long as $\mathbf{y}_d(t)$ is kept away from singular configurations.

4.3. Inverse dynamics controller: joint space

The task is accomplished if the limit

$$\lim_{t \rightarrow \infty} [\mathbf{q}_d(t) - \mathbf{q}(t)] = \mathbf{0}, \quad (53)$$

is satisfied. Let us consider the inverse dynamics control algorithm given in [26,27],

$$\begin{aligned} \boldsymbol{\tau} = M(\mathbf{q}) \left[\ddot{\mathbf{q}}_d + K'_v \dot{\tilde{\mathbf{q}}} + K'_p \tilde{\mathbf{q}} \right] \\ + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + F_v \dot{\mathbf{q}} + \mathbf{f}_{cl}(\dot{\mathbf{q}}) \end{aligned} \quad (54)$$

where $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ is the joint position error, K'_v and K'_p are $n \times n$ symmetric positive definite matrices. The time-varying functions \mathbf{q}_d , $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$ are given in Eqs. (50)–(52), respectively.

By using Eq. (42), controller (54) can be implemented in the form of voltage input as

$$\mathbf{u} = \bar{M}(\mathbf{q}) \left[\ddot{\mathbf{q}}_d + K'_v \dot{\tilde{\mathbf{q}}} + K'_p \tilde{\mathbf{q}} \right] + \bar{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \bar{F}_v \dot{\mathbf{q}} + \bar{\mathbf{f}}_{cl}(\dot{\mathbf{q}}). \quad (55)$$

Since the robot model parameters in Eqs. (44)–(48) are assumed to be known, the implementation of the inverse dynamics controller (55) can be carried out without requiring the explicit value of the motor constants in the matrix K . In other words, the only information required to implement the inverse dynamics controller (55) are the numerical values of the parameters $\theta_1, \dots, \theta_{12}$ in Table 1 and the numerical values of the gains K_p and K_v .

On the other hand, the implementation of controller (55) requires the joint velocity $\dot{\mathbf{q}}$, which has been assumed to be unmeasurable. A practical solution to this problem is to assume that a good approach of joint velocity $\dot{\mathbf{q}}$ is given by Euler's approximation

$$\dot{\mathbf{q}}(kT) = \frac{\mathbf{q}(kT) - \mathbf{q}((k-1)T)}{T}, \quad (56)$$

where the product kT is the discrete time, $k = 0, 1, 2, \dots$ is the integer time index and $T = 0.001$ [s] is the sampling period.

The inverse dynamics controller in (55) was implemented in the experimental robot arm using the gains

$$\begin{aligned} K'_p &= \text{diag}\{100.0, 100.0\} [1/s^2], \\ K'_v &= \text{diag}\{20.0, 20.0\} [1/s]. \end{aligned}$$

The values of K'_p and K'_v used in the experimental test were selected by trial and error, until a relatively acceptable performance in the tracking error was obtained. Since the inverse dynamics controller (55) produces a decoupled linear second order closed-loop system, a critically damped system was desired.

4.4. New controller: operational space

Using similar arguments as in the implementation of the inverse dynamics controller (55), the proposed control law in (12) can be written as

$$\begin{aligned} \mathbf{u} = \bar{M}(\mathbf{q}) \dot{\boldsymbol{\omega}}_d^* + \bar{C}(\mathbf{q}, \boldsymbol{\omega}_d) \boldsymbol{\omega}_d + \bar{F}_v \boldsymbol{\omega}_d + \bar{\mathbf{f}}_{cl}(\boldsymbol{\omega}_d) + K'_p \dot{\boldsymbol{\xi}} + K'_i \boldsymbol{\xi}, \quad (57) \\ \dot{\boldsymbol{\xi}} = \boldsymbol{\omega}_d - \boldsymbol{\vartheta}, \quad (58) \end{aligned}$$

where K'_i and K'_p are 2×2 diagonal positive definite matrices, $\boldsymbol{\vartheta}$ is the output of the filter (15)–(16), and $\boldsymbol{\omega}_d$ and $\dot{\boldsymbol{\omega}}_d^*$ are defined in (17) and (18), respectively. In analogous form, the only information required to implement the new velocity controller (57) is the numerical values of the parameters $\theta_1, \dots, \theta_{12}$ in Table 1 and the numerical values of the gains K'_i and K'_p .

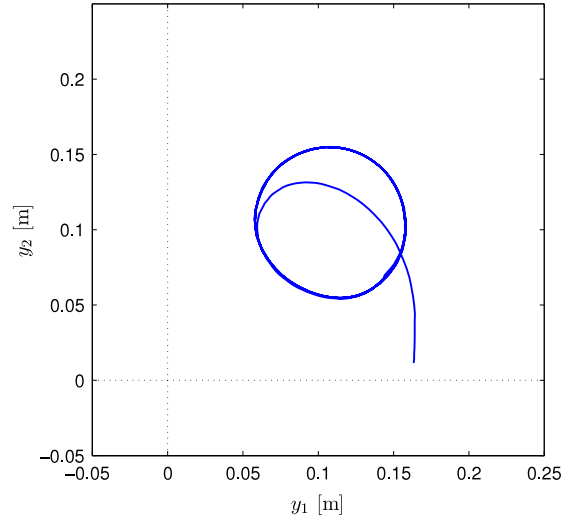


Fig. 3. Inverse dynamics controller: robot path in Cartesian coordinates.

By taking into account the gain parametrization $K'_i = K^{-1}K_i$, $K'_p = K^{-1}K_p$, and Eqs. (27)–(29), it is easy to see that the closed-loop system equation (32) is also obtained by substituting the controller equation (57) into the experimental robot model (43).

We have implemented the new joint velocity controller (57)–(58) together with filter (15)–(16) with gains

$$\begin{aligned} K'_i &= \text{diag}\{8.0, 8.0\} [\text{V/rad}] \\ K'_p &= \text{diag}\{0.8, 0.8\} [\text{V s/rad}], \\ A &= \text{diag}\{500.0, 500.0\} [1/s], \end{aligned}$$

while the kinematic controllers $\boldsymbol{\omega}_d$ and $\dot{\boldsymbol{\omega}}_d^*$ in (17) and (18), respectively, were implemented with

$$K_o = \text{diag}\{8.0, 6.0\} [1/s].$$

The initial condition for the filter (15)–(16) was $\mathbf{x}(0) = [-0.7303 \ -1.6901]^T$ [rad], making that $\boldsymbol{\vartheta}(0) = \mathbf{0}$.

Similarly to the case of the inverse dynamics controller (55), the gains K'_i , K'_p , A and K_o related to a new controller (57) were selected by trial and error. It is worthwhile to notice that the numerical values of the gains used in the inverse dynamics controller (55) cannot be used in the new controller (57) because the structure of one is quite different with respect to the other. However, the stability analysis shown in Section 3.4 suggests that the values of K'_i , K'_p , and A should be numerically high.

Furthermore, notice that in the inverse dynamics controller (55), the gains K'_p and K'_v are multiplied by the inertia matrix $\bar{M}(\mathbf{q})$, which allows a higher order of magnitude with respect to the gains K'_i , K'_p , used in controller (57).

4.5. Results and discussions

The experimental results using the inverse dynamics controller (55), which uses Euler's approximation (56), are shown in Figs. 3–5. They show the Cartesian robot path, the time history of the Cartesian positions $y_1(t)$ and $y_2(t)$, and the applied voltages $u_1(t)$ and $u_2(t)$, respectively.

On the other hand, the results of the implementation of the new joint velocity controller (57)–(58), filter (15)–(16), and kinematic controllers (17)–(18), are shown in Fig. 6, that depicts the Cartesian robot path, and Fig. 7, which shows the time history of the robot Cartesian positions $y_1(t)$ and $y_2(t)$. Furthermore, Fig. 8 describes the applied control voltages $u_1(t)$ and $u_2(t)$.

It is clear from Figs. 5 and 8 that high frequency components appear in the experimental control signals $u_1(t)$ and $u_2(t)$. The

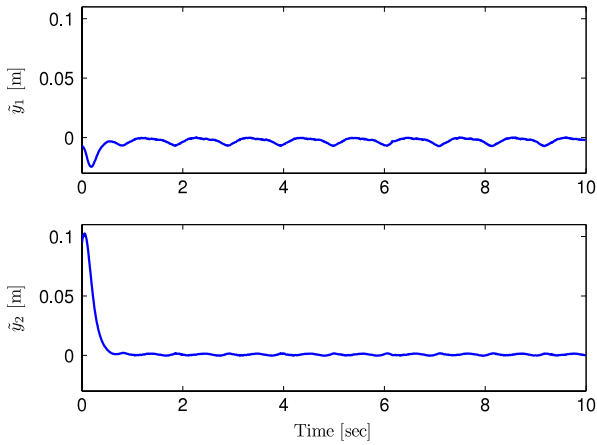


Fig. 4. Inverse dynamics controller: time history of the robot Cartesian errors $\hat{y}_1(t)$ and $\hat{y}_2(t)$.

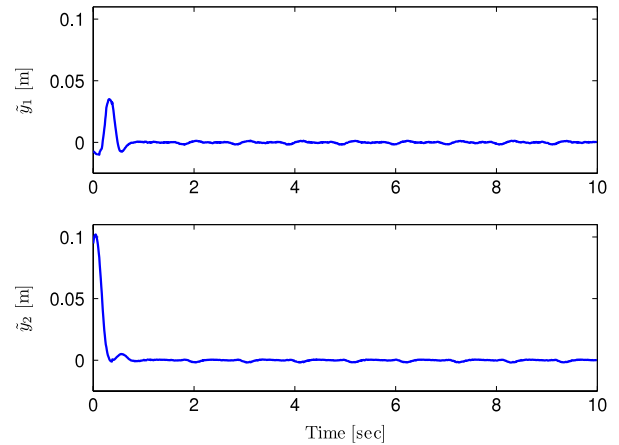


Fig. 7. New controller: time history of the robot Cartesian positions $\tilde{y}_1(t)$ and $\tilde{y}_2(t)$.

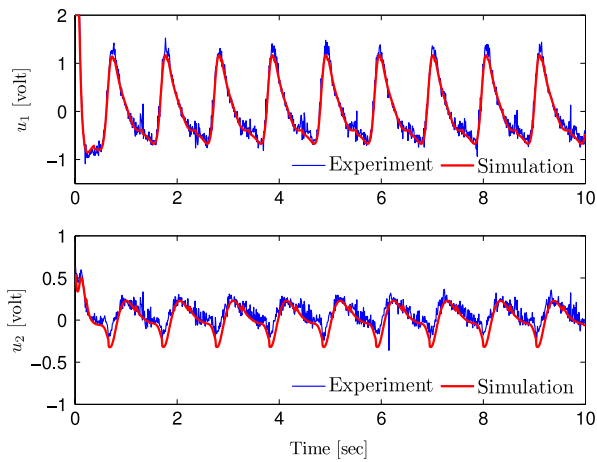


Fig. 5. Inverse dynamics controller: comparison of the applied voltages $u_1(t)$ and $u_2(t)$ obtained by experiment and numerical simulation.

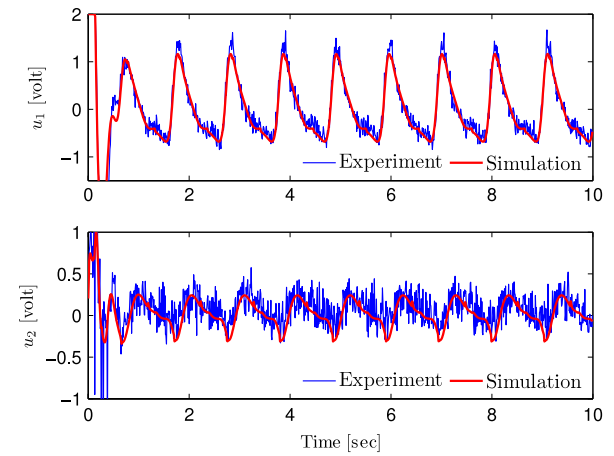


Fig. 8. New controller: comparison of the applied voltages $u_1(t)$ and $u_2(t)$ obtained by experiment and numerical simulation.

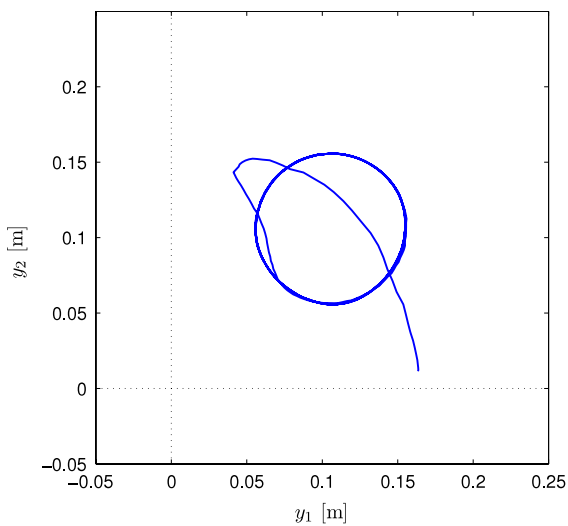


Fig. 6. New controller: robot path in Cartesian coordinates.

reasons are the 2000 [ppr] encoder resolution of the robot actuators, discrete implementation of the controller and filter, quantization errors, and PWM switching of the servo amplifiers. In spite of the high frequency contents in the voltage control signal,

we did not observe negative effects in the performance of the robot such as mechanical vibrations.

By using the robot model in (44)–(48), with the estimated parameters in Table 1, we have compared the performance of the implemented controllers with respect to numerical simulations, where non-quantized joint position measurements were assumed. Figs. 5 and 8 indicate that the experimental and simulation results are very similar. Additional numerical simulations showed that the high frequency contents of the control signals $u_1(t)$ and $u_2(t)$ were mainly due to the encoder resolution of the robot actuators and to the quantized joint position measurements.

Both cases show good tracking of the desired Cartesian trajectory as can be seen in Figs. 3 and 6. In order to evaluate the performance of the tested controllers we have used the contouring error, defined as [32]:

$$e_C(t) = r_0 - \sqrt{[y_{c1} - y_1(t)]^2 + [y_{c2} - y_2(t)]^2},$$

where $y_{c1} = 0.15$ [m] and $y_{c2} = 0.0$ [m] are the circle center coordinates and $r_0 = 0.05$ [m] is the corresponding radius. The contouring error for both controllers is shown in Fig. 9. The peak-to-peak value of the contouring errors were 0.0071 [m] for the inverse dynamics controller, and 0.0022 [m] for a new controller, that is based on synthetic joint velocity feedback. It is remarkable to see that by means of the new controller the performance was enhanced by 69% with respect to the inverse dynamics controller.

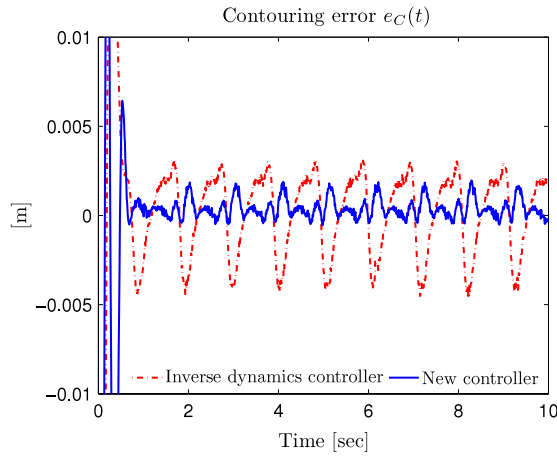


Fig. 9. Contouring error $e_c(t)$ for both controllers.

The reason for the new controller (57)–(58), used with filter (15)–(16) and kinematic controller (17)–(18), behaves better is its nested two-loop structure, which increases the robustness to disturbances. High gains in the velocity loop reject torque disturbances and make that the closed-loop approaches a first order system in the operational space error. However, the study on how robust is the controller, that is, the derivation of the relationship between the gains and the bounding of the disturbances is left as further research.

In conclusion, the experimental case studies showed that better performance is obtained with the proposed joint velocity controller (57)–(58), which is based on synthetic velocity $\vartheta(t)$ obtained from (15)–(16), and the kinematic controllers (17)–(18).

5. Concluding remarks

This paper dealt with the trajectory tracking control of manipulators in operational space. The main contribution has been the introduction of a controller for pose trajectory tracking control that makes use of only joint position measurements. Theory of singularly perturbed systems was crucial to show a rigorous analysis of the closed-loop system. An experimental study was carried out on a horizontal two degrees-of-freedom direct-drive robot. The inverse dynamics controller and the proposed scheme were tested. Matching between the numerical simulation and the real-time execution of the controllers was appreciated, which shows the nonlinear dynamics of the experimental system. The synthetic velocity-based hierarchical controller showed better performance than the classical inverse dynamics joint trajectory tracking controller.

Appendix. Stability of singularly perturbed systems

The symbol B_r denotes the set given by the ball $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$.

Theorem 1. Consider the singularly perturbed system

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}, \epsilon), \quad \mathbf{x} \in \mathbb{R}^{n_1} \quad (59)$$

$$\epsilon \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{x}, \mathbf{z}, \epsilon), \quad \mathbf{z} \in \mathbb{R}^{n_2}, \quad (60)$$

and $\epsilon > 0$. Assume that the following assumptions are satisfied for all $(t, \mathbf{x}, \epsilon) \in [0, \infty) \times B_r \times [0, \epsilon_0]$

$$1. \mathbf{f}(t, \mathbf{0}, \mathbf{0}, \epsilon) = \mathbf{0} \text{ and } \mathbf{g}(t, \mathbf{0}, \mathbf{0}, \epsilon) = \mathbf{0}.$$

2. The equation

$$\mathbf{0} = \mathbf{g}(t, \mathbf{x}, \mathbf{z}, \mathbf{0})$$

has an isolated root $\mathbf{z} = \mathbf{h}(t, \mathbf{x})$ such that $\mathbf{h}(t, \mathbf{0}) = \mathbf{0}$.

3. The functions \mathbf{f} , \mathbf{g} , and \mathbf{h} and their partial derivatives up to order 2 are bounded for $\mathbf{z} - \mathbf{h}(t, \mathbf{x}) \in B_\rho$.

4. The origin of the reduced system

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{h}(t, \mathbf{x}), \mathbf{0}) \quad (61)$$

is exponentially stable.

5. The origin of the boundary-layer system

$$\frac{d\mathbf{p}}{d\sigma} = \mathbf{g}(t, \mathbf{x}, \mathbf{p} + \mathbf{h}(t, \mathbf{x}), \mathbf{0}), \quad (62)$$

$\sigma = t/\epsilon$, $\mathbf{p} = \mathbf{z} - \mathbf{h}(t, \mathbf{x})$, is exponentially stable, uniformly in (t, \mathbf{x}) .

Then, there exists $\epsilon^* > 0$ such that, for all $\epsilon < \epsilon^*$, the origin of (59)–(60) is exponentially stable.

Proof. See [24], page 380. \square

References

- [1] Khatib O. A unified approach to motion and force control of robot manipulators: the operational space formulation. *IEEE Journal on Robotics and Automation* 1987;3(1):43–53.
- [2] Natale C. Interaction control of robot manipulators: six degrees-of-freedom tasks. Germany: Springer; 2003.
- [3] Lizarralde F, Wen J. Attitude control without angular velocity measurement: a passivity approach. *IEEE Transactions on Automatic Control* 1996;41(3):468–72.
- [4] Tsiotras P. Further passivity results for the attitude control problem. *IEEE Transactions Automatic Control* 1998;43(11):1597–600.
- [5] Wong H, de Queiroz MS, Kapila V. Adaptive tracking control using synthesized velocity from attitude measurements. *Automatica* 2001;37(6):947–53.
- [6] Caccavale F, Natale C, Villani L. Task-space tracking control without velocity measurements. In: Proc. of the 1999 IEEE international conference on robotics and automation. 1999. p. 512–17.
- [7] Xian B, de Queiroz MS, Dawson D, Walker I. Task-space tracking control of robot manipulators via quaternion feedback. *IEEE Transactions on Robotics and Automation* 2004;20(1):160–7.
- [8] Masson GS, Stone LS. From following edges to pursuing objects. *Journal of Neurophysiology* 2002;88:2869–73.
- [9] Corke P. The unimation puma servo system, Report MTM–226, CSIRO Division of Manufacturing Technology, Australia, July, 1994.
- [10] Nilsson K. Industrial robot programming, Ph.D. thesis, Lund Institute of Technology, Dept. of Automatic Control, Sweden, 1996.
- [11] Kelly R, Moreno J. Manipulator motion control in operational space using joint velocity inner loops. *Automatica* 2005;41(8):1423–32.
- [12] Aicardi M, Caiti A, Cannata G, Casalino G. Stability and robustness analysis of a two layered hierarchical architecture for the closed loop control of robots in the operational space. In: Proc. of the 1995 IEEE international conference on robotics and automation. Japan: Nagoya; 1995. p. 2771–8.
- [13] Yu K, Kieffer J. Robotic force/velocity control for following unknown contours of granular materials. *Control Engineering Practice* 1999;7:1249–56.
- [14] Roy J, Whitcomb LL. Adaptive force control of position/velocity controlled robots: theory and experiments. *IEEE Transactions on Robotics and Automation* 2002;18(2):121–37.
- [15] Jatta F, Legnani G, Visioli A. Friction compensation in hybrid force/velocity control of industrial manipulators. *IEEE Transactions on Industrial Electronics* 2006;53(2):604–13.
- [16] Moreno-Valenzuela J. Velocity field control of robot manipulators by using only position measurements. *Journal of the Franklin Institute* 2007;344:1021–38.
- [17] Herman P. A quasi-velocity-based nonlinear controller for rigid manipulators. *Mechanics Research Communications* 2009;36(7):859–66.
- [18] Vuong ND, Ang MH, Lim TM, Lim SY. Multi-rate operational space control of compliant motion in robotic manipulators. In: Proc. of the 2009 IEEE international conference on systems, man and cybernetics. 2009. p. 3175–80.
- [19] Wang L, Chai T, Fang Z. Neural-network-based two-loop control of robotic manipulators including actuator dynamics in task space. *Journal of Control Theory and Applications* 2009;7(2):112–8.
- [20] Beji L, Abichou A. A singular perturbation approach for tracking control of a parallel robot including motor dynamics. *International Journal of Control* 1997;68(4):689–708.
- [21] Heredia JA, Yu W. A high-gain observer-based PD manipulator control for robot. In: Proc. of the proceedings of the American control conference. 2000. p. 2218–522.

- [22] Subudhi B, Morris AS. Singular perturbation approach to trajectory tracking of flexible robot with joint elasticity. *International Journal of Systems Science* 2003;34(3):167–79.
- [23] Tavasoli A, Eghtesad M, Jafarian H. Two-time scale control and observer design for trajectory tracking of two cooperating robot manipulators moving a flexible beam. *Robotics and Autonomous Systems* 2009;57(2):212–21.
- [24] Khalil H. *Nonlinear systems*. Upper Saddle River: Prentice-Hall; 1996.
- [25] Moreno J, Kelly R. On motor velocity control by using only position measurements: two case studies. *International Journal of Electrical Engineering Education* 2002;39(2):118–27.
- [26] Sciavicco L, Siciliano B. *Modelling and control of robot manipulators*. London: Springer; 2000.
- [27] Kelly R, Santibáñez V, Loría A. *Control of robot manipulators in joint space*. New York: Springer; 2005.
- [28] Canudas de Wit C, Siciliano B, Bastin G, editors. *Theory of robot control*. London: Springer-Verlag; 1996.
- [29] Belanger PR. Estimation of angular velocity and acceleration from shaft encoder measurements. In: *Proc. of the 1992 IEEE international conference on robotics and automation*. p. 585–92.
- [30] Siciliano B. Kinematic control of redundant robot manipulators. *Journal of Intelligent and Robotic Systems* 1990;(3):201–12.
- [31] Gautier M, Poignet Ph. Extended kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice* 2001;9(12):1361–72.
- [32] Chiu GT-C, Tomizuka M. Contouring control of machine tool feed drive systems: a task coordinate frame approach. *IEEE Transactions on Control Systems Technology* 2000;9(1):130–1.