IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 12, 2021, accepted May 20, 2021, date of publication June 2, 2021, date of current version June 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3085528

A Force/Motion Control Approach Based on Trajectory Planning for Industrial Robots With Closed Control Architecture

ALEJANDRO GUTIERREZ–GILES[1], LUIS U. EVANGELISTA–HERNANDEZ[1], MARCO A. ARTEAGA[2], CARLOS A. CRUZ–VILLAR[1], (Member, IEEE), AND ALEJANDRO RODRIGUEZ–ANGELES[1]

[1]Center for Research and Advanced Studies (CINVESTAV), Electrical Engineering Department, Mechatronics Section, Mexico City 07360, Mexico
[2]Electrical Engineering Division, Control and Robotics Department, National Autonomous University of Mexico (UNAM), Mexico City 04510, Mexico

Corresponding author: Alejandro Gutierrez–Giles (alejandro.giles@cinvestav.mx)

This work was supported in part by the Dirección General de Asuntos del Personal Académico (DGAPA-UNAM) under Grant IN117820, in part by the Mexican National Council for Science and Technology, (CONACyT) under Grant CB-2015-01/254329 and Grant CB2017-2018-A1-S-26123, and in part by the Scholarship under Grant CVU 334785.

ABSTRACT Most industrial robots are provided by the manufacturer with a controller that cannot be modified by the user (e.g. a standard PID). This arrangement is commonly referred to as *closed control architecture*, since it is not possible to program arbitrary control laws. For the implementation of novel algorithms, it is on the contrary necessary to employ an *open control architecture*, which allows programming any control scheme. For that reason, it is customary to have testbeds that are made up of robot manipulators specially designed for this goal. Another disadvantage of the closed control architecture is that the controller provided by the manufacturer usually does not include a force control term since it allows only to program desired motion trajectories. To overcome these drawbacks without physically modifying the closed control architecture, this contribution presents a novel approach to simultaneously follow position and force trajectories by employing only motion planning, i.e. only by choosing the desired position trajectory. The approach is especially well suited for DC motor actuators with large gear reduction ratios as those of many industrial robots. The convergence of the manipulator position and applied force depends exclusively on the performance of the controller provided by the manufacturer. The approach is tested on a dual-arm cooperative manipulation system made up of two *ABB IRB-2400* industrial robots with closed control architecture.

INDEX TERMS Force control, motion control, robot control, robotic assembly, manufacturing automation.

I. INTRODUCTION

Nowadays, as new and more complex tasks are performed by robots in the industry, they need to interact with human beings, other robots, or with the environment, and not only to move objects or tools. Such interaction requires also force controllers [1], as well as faster input/output interfaces, fundamental in cooperative manipulation or human/robot interaction systems [2]. However, most industrial robot manufacturers do not offer this as a standard option nor the possibility for the user to freely program his/her own control algorithm. This configuration is known as *closed control architecture* since the control law is already programed (e.g. a regular PID) and it is not meant to be modified at all. For the user to implement innovative control schemes it is necessary to have an *open control architecture*, which for industrial robots can usually be achieved by replacing the manufacturer's controller either with a higher (more expensive) option provided by the same company or by a "home-made" control device. This may allow the implementation of force controllers designed by the user, with external sensors and algorithms built at a high level [3]. This paper introduces a proposal to avoid changing the control module provided by the manufacturer just by cleverly choosing the desired position trajectory which includes a force control term. It is worthy to notice that the performance of the overall system is limited by sampling and bandwidth considerations [4].

A. RELATED WORKS

Force control can be achieved by employing several techniques. There are basically two main categories in which

The associate editor coordinating the review of this manuscript and approving it for publication was Shihong Ding.

force control can be divided namely, indirect and direct force control [1]. To the first category belong the impedance and admittance controllers [5]. These approaches are well suited for nonrigid environments, and they can achieve force control without the necessity of a force sensor, although the surface stiffness is still required to be known *a priori*, or it has to be estimated online as in [6]. To the second category belong the hybrid force/motion and the parallel position/force controllers. On one hand, the hybrid force/motion approach is well suited for rigid environments whose geometry is known in advance, or it has to be estimated online as well [7]. On the other hand, the parallel position/force approach is better suited for nonrigid and/or unknown environments. In this case, the force control is intended to dominate the motion part, therefore some position error at the constrained directions must be tolerated [8].

Implementation of direct force controllers requires a two-loop architecture: one in charge of Cartesian position tracking, and the other one to deal with the force control at the end-effector. The output of these two loops is passed to the low-level joint controller provided by the manufacturer. Besides these controllers, it is a common practice to have a supervisory controller to manage the single and coordinated tasks for the case of a cooperative system [9].

Dual-arm manipulation has become a common task in industry, either because the desired task cannot be performed by a single manipulator, or because of the size of the object to be manipulated, or due to flexibility and dexterity requirements. To accomplish a dual-arm manipulation task, coordinated or uncoordinated approaches can be considered [10]. In the former, both robots share the same task with independent but coordinated motion, whereas in the latter each robot moves independently [11]. Often, the number of available sensors and the type of communication between the robots determine the approach to follow.

It has been shown that when manipulating an object by a cooperative multi-robot system, a decomposition of generalized forces/wrenches is required. Moreover, load distribution is highly desirable among the manipulators. Force decomposition allows analyzing the repercussions of the end-effector forces on internal torques, which imposes kinematic constraints [12]. In turn, load distribution defines the object grasping internal stresses [13]. When the robots interact with an object, kinematic constraints at the end-effectors arise to establish a closed kinematic chain that has to be kept to ensure object grasping. These constraints define a Jacobian, which relates forces and torques, but relies on a *pseudoinverse* to compute the external/internal force decomposition. To avoid the Jacobian inversion, [14] proposes an optimization algorithm that is solved by differential evolution and tested on a dual-arm *KUKA Youbot* system. When decomposing the generalized forces applied to the object, its geometry and dynamic parameters are assumed to be available. However, it is often the case that such data is not available, for which some estimators have been designed, showing that by using force sensors it is possible to perform the intended manipulation task even with a poor knowledge of the object to be manipulated [15]. The object dynamics is an important consideration to achieve bounded internal forces when grasping it and simultaneously interacting with the environment, as shown in [16]. In the mentioned work, both impedance-based centralized and decentralized controllers are proposed and validated through experimental results with a pair of open control architecture industrial robots.

In [17], a comparison between two impedance control approaches, namely the Multiple Impedance Controller (MIC) and the Augmented Object Model (AOM), is presented for a dual-arm manipulation system. The task consists in inserting an object into a peg, for which the authors conclude that the MIC has smaller tracking errors and dissipates more energy than the AOM in presence of impacts between the manipulated object and the environment.

By taking into account the kinematic constraints imposed at the end-effectors, it is possible to consider a dual-arm system as a single kinematic chain through a relative Jacobian, mapping each manipulator joint velocities to the relative motion of the object and the end-effectors, as shown in [18], where an impedance controller based on a desired dynamics of the object is proposed. The authors validate the effectiveness of the approach by a set of experimental results with two *Faraman* robots.

Another approach that considers spatial and temporal constraints for dual-arm manipulation tasks is the so called *STAAMS (Simultaneous Task Allocations and Motion Scheduling)*, which focuses on planning and scheduling a set of tasks to achieve the manipulating goal. In [19], this approach is applied to an assembling task with a pair of industrial *KUKA LBR iiwa* manipulators.

To achieve high-performance in the control of cooperative robotic systems, it is necessary to take into account the dynamics of the manipulators themselves in the controller design. Several works propose adaptive control schemes to deal with this problem (e.g. [20]). However, as the number of degrees of freedom grows, these kinds of controllers become more complex and difficult to implement, for which some alternatives such as multiple-input/multiple-output fuzzy logic-based controllers [21] have been explored as well.

Dual-arm manipulation has been also investigated in the context of robotic teleoperation, with the increased complexity of the control objectives, namely stability in presence of time-delays and transparency of the teleoperator. Some recent solutions have been proposed based on second-order sliding modes [22] and adaptive sliding mode controllers based on neural networks [23].

To achieve high-performance manipulation by a dual-arm system, it is highly desirable to have access to the motor torque/current control, as well as force/torque sensors. However, most industrial robots have closed control architecture, thus allowing only kinematic control, i.e. only velocity and/or position references can be commanded, which in turn are passed to the built-in low-level controller.

Furthermore, this also limits the sampling time. Nevertheless, it has been shown that pure kinematic control laws at industrial robots are sufficient to perform most dual-arm manipulation and collaborative human-robot tasks [24]. As pointed out in this reference, low-level industrial controllers overcome coupled and nonlinear robot dynamics. Recent efforts have been made to improve human-robot safe interaction with closed control architecture by taking advantage of a force and torque sensor [25].

B. MOTIVATION AND CONTRIBUTIONS

The main motivation of this work is to propose an approach to introduce a force control term for an industrial robot with closed control architecture, without needing the implementation of a specialized force control module and without changing the original position control law provided by the manufacturer. The basic idea is therefore to propose the desired position trajectory which contains all the information necessary to achieve this goal and that can be given to the manufacturer's standard motion planner. To summarize, the main contributions are:

- The proposed scheme makes it possible to simultaneously follow trajectories in both position and force, by employing only desired positions and velocities commanded to the manufacturer's controller, which is assumed to be a standard motion planner.
- The proposed approach is supported by a mathematical analysis that guarantees closed loop convergence of the contact forces and position trajectories to the desired references, depending on the precision provided by the manufacturer's controller.
- The scheme is valid for both rigid and soft objects/environments, as well as for a single robot or for dual-arm cooperative manipulation systems.

C. OUTLINE OF THE PAPER

The remainder of the manuscript is organized as follows: in Section 2, a mathematical model of a robot interacting with its environment is presented, as well as some of its useful properties. In Section 3, the main contribution of the article is developed, i.e. the proposed control approach with the corresponding mathematical analysis for the case of rigid and nonrigid surfaces, as well as an extension to the case of dual-arm manipulation. In Section 4, some experimental results are presented to validate the proposed scheme. Finally, in Section 5 some conclusions and directions for future work are given.

II. PRELIMINARIES

A. ROBOT's DYNAMICS

Consider a n–degrees of freedom rigid robot in contact with its environment, represented by a m–dimensional constraint. The dynamics of the system is then given by [26]:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{F}_{\mathrm{e}}, \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of generalized joint coordinates, $\boldsymbol{H}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal torques, $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal positive semidefinite matrix accounting for viscous friction, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the vector of gravitational torques, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of torques acting at the joints, $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the manipulator geometric Jacobian and $\boldsymbol{F}_{\mathrm{e}} \in \mathbb{R}^n$ is the vector of environmental forces. We use the notation

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}), \tag{2}$$

to denote the robot direct kinematics, while it is assumed that the relationship for the inverse kinematics can be denoted as

$$\boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of work–space coordinates. Its definition is sometimes better suited according to the following well–known relationship

$$\dot{\boldsymbol{x}} = \begin{bmatrix} {}^0\dot{\boldsymbol{p}}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where ${}^0\boldsymbol{\omega}_n \in \mathbb{R}^{\nu}$ is the angular velocity of the end–effector, while ${}^0\boldsymbol{p}_n \in \mathbb{R}^{(n-\nu)}$ is the end–effector position. Usually $n = 6$ and $\nu = 3$. Then, $\boldsymbol{x}$ is given in this case by

$$\boldsymbol{x} = \begin{bmatrix} {}^0\boldsymbol{p}_n \\ {}^0\boldsymbol{\phi}_n \end{bmatrix} = \begin{bmatrix} {}^0\boldsymbol{p}_n \\ \int_0^t {}^0\boldsymbol{\omega}_n \mathrm{d}\vartheta \end{bmatrix}. \tag{5}$$

We can further set[1]
$\boldsymbol{x} = \begin{bmatrix} x & y & z & \phi_{\mathrm{x}} & \phi_{\mathrm{y}} & \phi_{\mathrm{z}} \end{bmatrix}^{\mathrm{T}}$.

As explained in full detail in [26], ${}^0\boldsymbol{\phi}_n$ in (5) does not have any physical meaning. If it is desired to have a vector ${}^0\boldsymbol{\phi}_n$ with a physical meaning, e.g. Euler angles, the analytical Jacobian $\boldsymbol{J}_{\mathrm{A}}(\boldsymbol{q})$ should be used instead. For that case, however, $\frac{\mathrm{d}}{\mathrm{d}t}{}^0\boldsymbol{\phi}_n$ purely represents the derivative of the orientation representation given by ${}^0\boldsymbol{\phi}_n$. Whenever the robot is not in a singularity, the following relationship holds

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}. \tag{6}$$

B. FORCE DESCRIPTION

A key point in force control is the description of the force interacting with the environment. Assume that the robot gets in contact with a surface or object, whose undeformed shape can be described by the following relationship

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{0}, \tag{7}$$

where $\boldsymbol{\varphi}(\boldsymbol{x}) \in \mathbb{R}^m$. Furthermore, any workspace configuration satisfying (7) belongs to a set $\mathcal{X}$, i.e. if $\boldsymbol{x} \in \mathcal{X}$ then (7) holds. For simplicity, we use the notation $\boldsymbol{x}_0$ for any $\boldsymbol{x} \in \mathcal{X}$.

[1] Using $\boldsymbol{x}$ as vector and x as scalar (to designate the x–coordinate) should not cause any confusion.

1) RIGID SURFACES: PRINCIPLE OF ORTHOGONALIZATION

Assume that the force $\boldsymbol{F}_e$ arises at the contact with a rigid object or frictionless surface. For that case the Principle of Orthogonalization can be employed [27]:

Property 1: The vector $\dot{\boldsymbol{x}}$ satisfies

$$\dot{\boldsymbol{x}} = \boldsymbol{Q}_x(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{P}_x(\boldsymbol{x})\dot{\boldsymbol{x}} = \boldsymbol{Q}_x(\boldsymbol{x})\dot{\boldsymbol{x}}, \tag{8}$$

where $\boldsymbol{Q}_x(\boldsymbol{x}) \triangleq (\boldsymbol{I}_{n\times n} - \boldsymbol{P}_x(\boldsymbol{x}))$, $\boldsymbol{P}_x(\boldsymbol{x}) \triangleq \boldsymbol{J}_{\varphi x}^{+}\boldsymbol{J}_{\varphi x}$, and $\boldsymbol{J}_{\varphi x}^{+} \triangleq \boldsymbol{J}_{\varphi x}^{T}\left(\boldsymbol{J}_{\varphi x}\boldsymbol{J}_{\varphi x}^{T}\right)^{-1}$ stands for the Penrose's pseudoinverse, and $\boldsymbol{Q}_x \in \mathbb{R}^{n\times n}$ satisfies rank $(\boldsymbol{Q}_x) = n - m$. These two matrices are orthogonal, i.e. $\boldsymbol{Q}_x\boldsymbol{P}_x = \boldsymbol{O}$ (and in fact $\boldsymbol{Q}_x\boldsymbol{J}_{\varphi x}^{T} = \boldsymbol{O}$ and $\boldsymbol{J}_{\varphi x}\boldsymbol{Q}_x = \boldsymbol{O}$). Note that the last equality in (8) is due to the fact that $\dot{\boldsymbol{\varphi}}(\boldsymbol{x}) = \boldsymbol{J}_{\varphi x}\dot{\boldsymbol{x}} = \boldsymbol{0}$ in view of constraint (7). □

2) DEFORMABLE SURFACES

In case the contact surface is deformable a more proper way to describe the contact force is given by [1]:

$$\boldsymbol{F}_e = \boldsymbol{K}_f(\boldsymbol{x} - \boldsymbol{x}_0), \tag{9}$$

where $\boldsymbol{K}_f \in \mathbb{R}^{n\times n}$ is the contact stiffness matrix, which we assume to be diagonal and positive semidefinite.

C. ACTUATORS' DYNAMICS

Equation (1) describes the dynamics of a rigid robot manipulator with an external force without taking into account the actuators' dynamics, typically hydraulic, pneumatic, or electrical devices. A case of particular interest due to its wide use in industrial robots are DC motors, for which it is possible to get the following relationship between the generalized input torques and the input voltages $\boldsymbol{v} \in \mathbb{R}^n$ applied to the motors' armature [28]

$$\boldsymbol{D}_j\ddot{\boldsymbol{q}} + \boldsymbol{D}_f\dot{\boldsymbol{q}} + \boldsymbol{D}_r\boldsymbol{\tau} = \boldsymbol{D}_K\boldsymbol{v}, \tag{10}$$

where $\boldsymbol{D}_r, \boldsymbol{D}_j, \boldsymbol{D}_f, \boldsymbol{D}_K \in \mathbb{R}^{n\times n}$ are all diagonal matrices given by

$$\boldsymbol{D}_j = \text{diag}\left\{\begin{bmatrix} J_{m1} & \cdots & J_{m6}\end{bmatrix}\right\} \tag{11}$$

$$\boldsymbol{D}_f = \text{diag}\left\{\begin{bmatrix} f_{m1} + \frac{K_{a1}K_{b1}}{R_{a1}} & \cdots & f_{m6} + \frac{K_{a6}K_{b6}}{R_{a6}}\end{bmatrix}\right\} \tag{12}$$

$$\boldsymbol{D}_r = \text{diag}\left\{\begin{bmatrix} \frac{1}{r_1^2} & \cdots & \frac{1}{r_6^2}\end{bmatrix}\right\} \tag{13}$$

$$\boldsymbol{D}_K = \text{diag}\left\{\begin{bmatrix} \frac{K_{a1}}{R_{a1}r_1} & \cdots & \frac{K_{a6}}{R_{a6}r_6}\end{bmatrix}\right\}. \tag{14}$$

For $i = 1, \ldots, 6$ J_{mi} [kg m^2] is the rotor inertia, f_{mi} [N m] is the rotor friction coefficient, K_{ai} [N m/A] is the motor–torque constant, R_{ai} [Ω] is the armature resistance and K_{bi} [V s/rad] is the back emf constant. Finally, r_i is the gear reduction ratio usually chosen so that $r_i \gg 1$. In this work this case is assumed.

Substituting (10) in (1) yields

$$\left(\boldsymbol{H}(\boldsymbol{q}) + \boldsymbol{D}_r^{-1}\boldsymbol{D}_j\right)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \left(\boldsymbol{D} + \boldsymbol{D}_r^{-1}\boldsymbol{D}_f\right)\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{D}_r^{-1}\boldsymbol{D}_K\boldsymbol{v} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F}_e. \tag{15}$$

III. MAIN RESULT

Most industrial robots do not allow the user to program his/her own control algorithm, but only a desired trajectory $\boldsymbol{x}_d \in \mathbb{R}^6$ containing both the information of position and orientation of the manipulator end-effector. Once inverse kinematics is applied, the corresponding desired trajectory in joint coordinates becomes $\boldsymbol{q}_d \in \mathbb{R}^n$, where usually it is assumed that at least its first and second time-derivatives are bounded. Then, the position tracking error is defined as

$$\boldsymbol{e}_p = \boldsymbol{q} - \boldsymbol{q}_d, \tag{16}$$

for joint coordinates, and as

$$\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_d, \tag{17}$$

for workspace coordinates. Probably the most employed control law in industrial robots is a regular PID [29], defined as

$$\boldsymbol{v}_{pid} = -\boldsymbol{K}_{mp}\boldsymbol{e}_p - \boldsymbol{K}_{mi}\int_0^t \boldsymbol{e}_p dt - \boldsymbol{K}_{md}\dot{\boldsymbol{e}}_p, \tag{18}$$

where $\boldsymbol{K}_{mp}, \boldsymbol{K}_{mi}, \boldsymbol{K}_{md} \in \mathbb{R}^{n\times n}$ are diagonal positive definite matrices.

Assumption 1: The controller gains $\boldsymbol{K}_p, \boldsymbol{K}_i, \boldsymbol{K}_d$ have been properly tuned by the robot manufacturer to achieve $\boldsymbol{e}_p \approx \boldsymbol{0}$ and $\dot{\boldsymbol{e}}_p \approx \boldsymbol{0}$ for time varying trajectories $\boldsymbol{q}_d(t)$, and $\boldsymbol{e}_p = \dot{\boldsymbol{e}}_p = \boldsymbol{0}$ for a constant $\boldsymbol{q}_d$. Therefore, the PID control law achieves $\Delta\boldsymbol{x} \approx \boldsymbol{0}$ and $\Delta\dot{\boldsymbol{x}} \approx \boldsymbol{0}$ for time varying $\boldsymbol{x}_d(t)$, and $\Delta\boldsymbol{x} = \Delta\dot{\boldsymbol{x}} = \boldsymbol{0}$ for a constant $\boldsymbol{x}_d$. □

Remark 1: Note that the proposed approach assumes that the controller gains $\boldsymbol{K}_{mp}$, $\boldsymbol{K}_{mi}$, and $\boldsymbol{K}_{md}$ in (18) are set by the manufacturer and the user has no way to retune them, given the closed control architecture considered in this work. Assumption 1 also implies that the controller has some degree of robustness to external disturbances. The following analysis is carried out assuming the motion controller provided by the manufacturer is a standard PID, but similar analyses can be derived for PD and PD plus gravity compensation motion controllers. □

Consider now rewriting (15) as

$$\left(\boldsymbol{D}_r\boldsymbol{H}(\boldsymbol{q}) + \boldsymbol{D}_j\right)\ddot{\boldsymbol{q}} + \boldsymbol{D}_r\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \left(\boldsymbol{D}_r\boldsymbol{D} + \boldsymbol{D}_f\right)\dot{\boldsymbol{q}} + \boldsymbol{D}_r\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{D}_K\boldsymbol{v} - \boldsymbol{D}_r\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F}_e. \tag{19}$$

By assumption it is $r_i \gg 1\ \forall\ i$, so that from (13) it holds $\boldsymbol{D}_r \approx \boldsymbol{O}$, meaning that (19) can be approximated by

$$\boldsymbol{D}_j\ddot{\boldsymbol{q}} + \boldsymbol{D}_f\dot{\boldsymbol{q}} = \boldsymbol{D}_K\boldsymbol{v}. \tag{20}$$

By substituting (18) in (20) one gets

$$\boldsymbol{D}_j\ddot{\boldsymbol{q}} + \boldsymbol{D}_f\dot{\boldsymbol{q}} = -\boldsymbol{K}_{dmp}\boldsymbol{e}_p - \boldsymbol{K}_{dmi}\int_0^t \boldsymbol{e}_p dt - \boldsymbol{K}_{dmd}\dot{\boldsymbol{e}}_p, \tag{21}$$

where $\boldsymbol{K}_{dmp} = \boldsymbol{D}_K\boldsymbol{K}_{mp}, \boldsymbol{K}_{dmi} = \boldsymbol{D}_K\boldsymbol{K}_{mi}, \boldsymbol{K}_{dmd} = \boldsymbol{D}_K\boldsymbol{K}_{md} \in \mathbb{R}^{n\times n}$ are diagonal positive matrices.

Remark 2: Equation (21) shows that for large gear reduction ratios the highly nonlinear dynamics of a robot manipulator becomes not only approximately linear, but also decoupled. However, CARE MUST BE TAKEN since in reality the position vector $\boldsymbol{q}$ in (21) is not free, but constrained by the force $\boldsymbol{F}_e$, even though it does not explicitly appear. The usefulness of (21) relies on the fact that it allows implementing a force control action indirectly, just by properly choosing the desired position trajectories, as it will be done next. □

In the following paragraphs, force tracking error is analyzed by employing the standard bounded-input bounded-output (BIBO) stability of linear first-order filters [30] to establish practical convergence of both position and forces to the desired references, for both rigid and nonrigid surfaces.

A. RIGID SURFACES

Suppose it is wished that the robot manipulator tracks a trajectory $\boldsymbol{x}_0 \in \mathcal{X}$ for a nondeformable surface so that a force $\boldsymbol{F}_e$ due to the contact with the environment arises. To avoid damages to the contact surface and to have an appropriate force performance, consider the following desired trajectory

$$\boldsymbol{x}_d = \boldsymbol{x}_0 + \boldsymbol{x}_f, \tag{22}$$

where

$$\boldsymbol{x}_f = \boldsymbol{K}_p \boldsymbol{e}_F + \boldsymbol{K}_i \int_0^t \boldsymbol{e}_F dt, \tag{23}$$

with $\boldsymbol{K}_p, \boldsymbol{K}_i \in \mathbb{R}^{n\times n}$ positive definite matrices of constant gains. The force error is defined as

$$\boldsymbol{e}_F = \boldsymbol{F}_d - \boldsymbol{F}_e, \tag{24}$$

where $\boldsymbol{F}_d \in \mathbb{R}^n$ is a desired (possibly time-varying) contact force. Now, the next proposition is stated regarding the force/position control over rigid surfaces.

Proposition 1: Let $\boldsymbol{x}_0 \in \mathcal{X}$ be the desired position trajectory satisfying the holonomic constraint (7) and let $\boldsymbol{F}_d \in \mathbb{R}^n$ be the desired contact force, with the corresponding force error defined by (24). Let the reference trajectory for the closed-loop architecture controller be computed by (22). Then, a combination of gains $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ can always be found to drive both position and force errors arbitrarily close to zero, as long as Assumption 1 holds.

Proof: In what follows, in order to simplify the forthcoming analysis and without loss of generality, these matrices are assumed to have the form $\boldsymbol{K}_p = k_p \boldsymbol{I}$ and $\boldsymbol{K}_i = k_i \boldsymbol{I}$, where $\boldsymbol{I}$ is the $n \times n$ identity matrix. According to Assumption 1, it is $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_d \approx \boldsymbol{0}$, which means that

$$\boldsymbol{x} - \boldsymbol{x}_0 - \boldsymbol{x}_f \approx \boldsymbol{0}. \tag{25}$$

Note that necessarily $\boldsymbol{x} \in \mathcal{X}$, while by definition $\boldsymbol{x}_0 \in \mathcal{X}$. Fig. 1 depicts the cases for small and large errors. Assumption 1 guarantees a small error, while Property 1 implies that the force is perpendicular to the surface, i.e. $\boldsymbol{F}_e = \boldsymbol{P}(\boldsymbol{x})\boldsymbol{F}_e$ and for small errors $\boldsymbol{F}_d \approx \boldsymbol{P}(\boldsymbol{x})\boldsymbol{F}_d$, so that

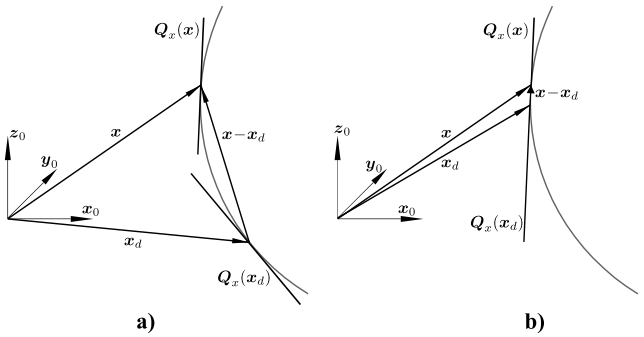

FIGURE 1. Position errors: a) Large error. b) Small error.

also $\boldsymbol{e}_F \approx \boldsymbol{P}(\boldsymbol{x})\boldsymbol{e}_F$. However, this is not necessarily true for the integral term $\int_0^t \boldsymbol{e}_F dt$, for which we consider two cases:

a) $\boldsymbol{P}(\boldsymbol{x})$ is constant. For that case it holds

$$\int_0^t \boldsymbol{P}(\boldsymbol{x})\boldsymbol{e}_F dt = \boldsymbol{P}(\boldsymbol{x}) \int_0^t \boldsymbol{e}_F dt, \tag{26}$$

which means that $\boldsymbol{x}_f = \boldsymbol{P}(\boldsymbol{x})\boldsymbol{x}_f$. Since $\boldsymbol{P}(\boldsymbol{x})$ spans the space orthogonal to the surface, it does not affect the robot's end–effector position. Thus, necessarily $\boldsymbol{x} \approx \boldsymbol{x}_0$ (or $\boldsymbol{x} = \boldsymbol{x}_0$ for a constant $\boldsymbol{x}_0$). For that reason one gets

$$\boldsymbol{x}_f = k_p \boldsymbol{e}_F + k_i \int_0^t \boldsymbol{e}_F dt \approx \boldsymbol{0}, \tag{27}$$

or

$$\boldsymbol{e}_F \approx -\frac{k_i}{k_p} \int_0^t \boldsymbol{e}_F dt. \tag{28}$$

This last equation represents a stable first order autonomous linear system, so that in the end both $\boldsymbol{e}_F$ and $\int_0^t \boldsymbol{e}_F dt$ tend (approximately) to $\boldsymbol{0}$ or exactly for a constant $\boldsymbol{x}_0$. A constant $\boldsymbol{P}(\boldsymbol{x})$ arises for flat surfaces, but if $\boldsymbol{x}_0$ is chosen constant, all errors, position and force, will tend to zero even for a non-flat surface.

b) $\boldsymbol{P}(\boldsymbol{x})$ is not constant and varies depending on $\boldsymbol{x}$. For that case the relationship (26) does not hold and instead one has

$$\begin{aligned} \boldsymbol{x}_f &= \boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_f + \boldsymbol{P}(\boldsymbol{x})\boldsymbol{x}_f \\ &= k_p \boldsymbol{e}_F + k_i \boldsymbol{Q}(\boldsymbol{x}) \int_0^t \boldsymbol{e}_F dt + k_i \boldsymbol{P}(\boldsymbol{x}) \int_0^t \boldsymbol{e}_F dt. \end{aligned} \tag{29}$$

Thus, instead of (28) it holds

$$\boldsymbol{e}_F \approx -\frac{k_i}{k_p} \boldsymbol{P}(\boldsymbol{x}) \int_0^t \boldsymbol{e}_F dt. \tag{30}$$

But it must also hold

$$\boldsymbol{x} \approx \boldsymbol{x}_0 + \boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_\mathrm{f}. \tag{31}$$

However, according to Fig. 1, $\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_\mathrm{f}$ cannot be too large simply because $\boldsymbol{x}_0 \in \mathcal{X}$ and a large $\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_\mathrm{f}$ would cause $\boldsymbol{x}_0 + \boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_\mathrm{f} \notin \mathcal{X}$ while $\boldsymbol{x} \in \mathcal{X}$, which represents a contradiction. Note that the exception would be a flat surface, but that is precisely Case a). Therefore, since $\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}_\mathrm{f} \approx \boldsymbol{0}$, then necessarily $\int_0^t \boldsymbol{e}_\mathrm{F}\mathrm{d}t \approx \boldsymbol{P}(\boldsymbol{x})\int_0^t \boldsymbol{e}_\mathrm{F}\mathrm{d}t$. This means that in the end the system behavior tends to be the same as before, i.e. $\boldsymbol{e}_\mathrm{F} \approx \boldsymbol{0}$ and $\boldsymbol{x} \approx \boldsymbol{x}_0$.

□

B. DEFORMABLE SURFACES

Suppose now that the surface is deformable so that the contact force is described by (9). The following result can be stated.

Proposition 2: Let $\boldsymbol{x}_0 \in \mathcal{X}$ be the desired position trajectory, designed to keep contact with the surface under no deformation, and let $\boldsymbol{F}_\mathrm{d} \in \mathbb{R}^n$ be the desired contact force, with the corresponding force error defined by (24). Let the reference trajectory for the closed-loop architecture controller be computed by (22). Then, a combination of gains $\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{i}$ can always be found to drive the force error arbitrarily close to zero while keeping the position error bounded, as long as Assumption 1 holds.

Proof: From (23) and (25), one has

$$\boldsymbol{x} - \boldsymbol{x}_0 \approx k_\mathrm{p}\boldsymbol{e}_\mathrm{F} + k_\mathrm{i}\int_0^t \boldsymbol{e}_\mathrm{F}\mathrm{d}t. \tag{32}$$

Multiplying both sides of (32) by $\boldsymbol{K}_\mathrm{f}$ one gets

$$\boldsymbol{F}_\mathrm{e} = \boldsymbol{K}_\mathrm{f}(\boldsymbol{x} - \boldsymbol{x}_0) \approx \boldsymbol{K}_\mathrm{f}\left(k_\mathrm{p}\boldsymbol{e}_\mathrm{F} + k_\mathrm{i}\int_0^t \boldsymbol{e}_\mathrm{F}\mathrm{d}t\right). \tag{33}$$

Alternatively using $\boldsymbol{F}_\mathrm{d}$ it is

$$\boldsymbol{e}_\mathrm{F} = \boldsymbol{F}_\mathrm{d} - \boldsymbol{F}_\mathrm{e} \approx \boldsymbol{F}_\mathrm{d} - k_\mathrm{p}\boldsymbol{K}_\mathrm{f}\boldsymbol{e}_\mathrm{F} - k_\mathrm{i}\boldsymbol{K}_\mathrm{f}\int_0^t \boldsymbol{e}_\mathrm{F}\mathrm{d}t, \tag{34}$$

or

$$(\boldsymbol{I} + k_\mathrm{p}\boldsymbol{K}_\mathrm{f})\dot{\boldsymbol{e}}_\mathrm{F} \approx -k_\mathrm{i}\boldsymbol{K}_\mathrm{f}\boldsymbol{e}_\mathrm{F} + \dot{\boldsymbol{F}}_\mathrm{d}. \tag{35}$$

Since $\boldsymbol{K}_\mathrm{f}$ is diagonal, then one can get n equations of the form

$$(1 + k_\mathrm{p}k_{\mathrm{f}i})\dot{e}_{\mathrm{F}i} \approx -k_\mathrm{i}k_{\mathrm{f}i}e_{\mathrm{F}i} + \dot{F}_{\mathrm{d}i}, \tag{36}$$

where for $i = 1, \ldots, n$, $\dot{e}_{\mathrm{F}i}, e_{\mathrm{F}i}$ and $\dot{F}_{\mathrm{d}i}$ denotes the i-th element of $\dot{\boldsymbol{e}}_\mathrm{F}, \boldsymbol{e}_\mathrm{F}$ and $\dot{\boldsymbol{F}}_\mathrm{d}$, respectively, while $k_{\mathrm{f}i}$ is the i-th element of the diagonal of $\boldsymbol{K}_\mathrm{f}$. Note that if $k_{\mathrm{f}i} \equiv 0$, then both $F_{\mathrm{d}i}$ and $F_{\mathrm{e}i}$ are zero, and (35) becomes trivially $0 = 0$. If $k_{\mathrm{f}i} > 0$, we distinguish two cases. If $F_{\mathrm{d}i}(t)$ is constant, then $\dot{F}_{\mathrm{d}i} \equiv 0$, and one gets

$$\dot{e}_{\mathrm{F}i} \approx -\frac{k_\mathrm{i}k_{\mathrm{f}i}}{(1 + k_\mathrm{p}k_{\mathrm{f}i})}e_{\mathrm{F}i}, \tag{37}$$

which is a stable autonomous linear filter, meaning that $e_{\mathrm{F}i} \to 0$ approximately. On the other hand, if $F_{\mathrm{d}i}(t)$ is time varying and thus $F_{\mathrm{d}i} \neq 0$. By applying the Laplace operator one gets

$$\left((1 + k_\mathrm{p}k_{\mathrm{f}i})s + k_\mathrm{i}k_{\mathrm{f}i}\right)E_{\mathrm{F}i}(s) \approx \dot{F}_{\mathrm{d}i}(s). \tag{38}$$

Note that in (38) it has been chosen to compute directly the Laplace transform of the derivative of $F_{\mathrm{d}i}$ as $\dot{F}_{\mathrm{d}i}(s)$ instead of using the well–known relationship $sF_{\mathrm{d}i}(s)$. For this reason (38) becomes

$$E_{\mathrm{F}i}(s) \approx \frac{1/(1 + k_\mathrm{p}k_{\mathrm{f}i})}{s + k_\mathrm{i}k_{\mathrm{f}i}/(1 + k_\mathrm{p}k_{\mathrm{f}i})}\dot{F}_{\mathrm{d}i}(s). \tag{39}$$

Clearly, the smaller the magnitude of $\dot{F}_{\mathrm{d}i}(t)$ the smaller the force error will be. Alternatively, the error can be made smaller by setting k_i larger since this makes the filter gain in (38) smaller. Note that the same result is gotten for a very large $k_{\mathrm{f}i}$, which represents a rigid surface. This is in good agreement with the result obtained using the Principle of Orthogonalization. □

Remark 3: Contrarily to the case of rigid surfaces, both position and force errors cannot be made arbitrarily close to zero for nonrigid surfaces by employing the proposed algorithm. This can be seen by looking at equations (32) and (39), where the larger the gains k_p and k_i, the smaller the force error, but at the expense of making the position error larger. Therefore, a trade-off between force and position tracking performance must be made for deformable surfaces. □

C. EXTENSION TO COOPERATIVE DUAL-ARM MANIPULATION

The extension of the proposed approach to the case of cooperative dual-arm manipulation is straightforward by employing the *cooperative task space* formulation proposed in [31]. In such formulation the desired motion of the object is defined as an *absolute* position and orientation given by

$$\boldsymbol{x}_{\mathrm{a}0} = \frac{1}{2}(\boldsymbol{x}_{10} + \boldsymbol{x}_{20}) \tag{40}$$

$$\boldsymbol{R}_{\mathrm{a}0} = {}^0\boldsymbol{R}_{10}\,{}^1\boldsymbol{R}_\mathrm{k}({}^1\boldsymbol{k}_{21}, \vartheta_{21}/2), \tag{41}$$

where $\boldsymbol{x}_{10}$ and $\boldsymbol{x}_{20}$ are the desired Cartesian positions for the end-effectors, ${}^0\boldsymbol{R}_{10}$ is the rotation matrix of one of the end-effector frames, ${}^1\boldsymbol{k}_{21}$ and $\vartheta_{21}/2$ are the axis/angle parametrization of the rotation ${}^1\boldsymbol{R}_2$, and ${}^j\boldsymbol{R}_\mathrm{k}({}^j\boldsymbol{k}, \vartheta)$ is the rotation matrix by ϑ over the axis ${}^j\boldsymbol{k}$. For planar rotations, the absolute rotation angle is simply given by $\vartheta_\mathrm{a} = \frac{1}{2}(\vartheta_1 + \vartheta_2)$, where ϑ_1 and ϑ_2 are the rotation angles of the end-effector frames for the first and second manipulators, respectively.

In turn, the desired *relative* position and orientation of the end-effectors are given by

$$\boldsymbol{x}_{\mathrm{r}0} = \boldsymbol{x}_{20} - \boldsymbol{x}_{10} \tag{42}$$

$$\boldsymbol{R}_{\mathrm{r}0} = {}^{0}\boldsymbol{R}_{10}^{\mathrm{T}}\,{}^{1}\boldsymbol{R}_{20}. \tag{43}$$

For planar rotations the relative orientation is simply $\vartheta_{\mathrm{r}} = \vartheta_2 - \vartheta_1$. In a similar fashion, the desired absolute linear and angular velocities are

$$\dot{\boldsymbol{x}}_{\mathrm{a}0} = \frac{1}{2}(\dot{\boldsymbol{x}}_{10} + \dot{\boldsymbol{x}}_{20}) \tag{44}$$

$$\boldsymbol{\omega}_{\mathrm{a}0} = \frac{1}{2}(\boldsymbol{\omega}_{10} + \boldsymbol{\omega}_{20}), \tag{45}$$

whereas the desired relative linear and angular velocities are given by

$$\dot{\boldsymbol{x}}_{\mathrm{r}0} = \dot{\boldsymbol{x}}_{20} - \dot{\boldsymbol{x}}_{10} \tag{46}$$

$$\boldsymbol{\omega}_{\mathrm{r}0} = \boldsymbol{\omega}_{20} - \boldsymbol{\omega}_{10}. \tag{47}$$

The mentioned quantities are shown in Fig. 2. For rigid objects' manipulation, the desired relative position and orientation must be constant. Thus, the desired positions and orientations for both end-effectors are easily computed once defined the desired absolute position and orientation for the object.

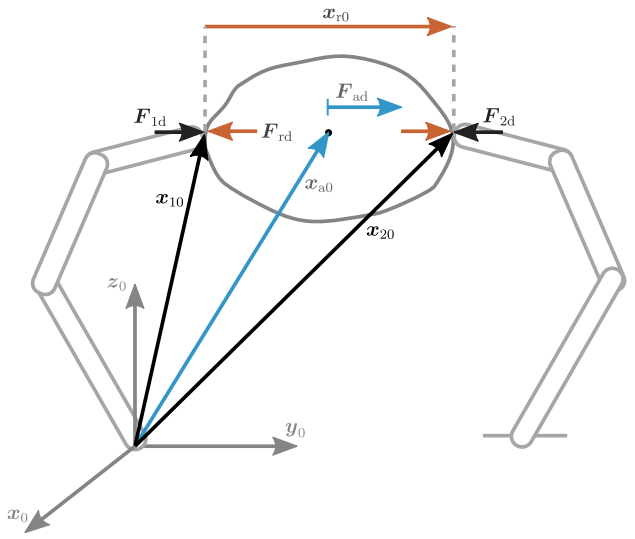

FIGURE 2. Cooperative task space formulation.

For the desired forces and moments, an analogous definition as that for the motion part is employed. The desired absolute forces and moments, $\boldsymbol{F}_{\mathrm{ad}}$ and $\boldsymbol{n}_{\mathrm{ad}}$, in terms of the corresponding end-effectors forces and moments, $\boldsymbol{F}_{id}$ and $\boldsymbol{n}_{iad}$, $i = 1, 2$ are

$$\boldsymbol{F}_{\mathrm{ad}} = \boldsymbol{F}_{1\,\mathrm{d}} + \boldsymbol{F}_{2\,\mathrm{d}} \tag{48}$$

$$\boldsymbol{n}_{\mathrm{ad}} = \boldsymbol{n}_{1\,\mathrm{d}} + \boldsymbol{n}_{2\mathrm{d}}, \tag{49}$$

and the desired relative forces and moments are given by

$$\boldsymbol{F}_{\mathrm{rd}} = \frac{1}{2}(\boldsymbol{F}_{2\,\mathrm{d}} - \boldsymbol{F}_{1\,\mathrm{d}}) \tag{50}$$

$$\boldsymbol{n}_{\mathrm{rd}} = \frac{1}{2}(\boldsymbol{n}_{2\,\mathrm{d}} - \boldsymbol{n}_{1\mathrm{d}}). \tag{51}$$

On the one hand, to obtain a decoupled motion/force controller as the one proposed in Section III-A, the desired absolute forces and moments must be zero, i.e. $\boldsymbol{F}_{\mathrm{ad}} = \boldsymbol{0}$ and $\boldsymbol{n}_{\mathrm{ad}} = \boldsymbol{0}$. On the other hand, the relative forces and moments represent the internal (grasping) forces on the object. Therefore, given the desired relative forces and torques, the desired ones for each end-effector are well defined.

IV. EXPERIMENTS

In this section, three experiments are presented to validate the results given in Section III. The first experiment consists in a force/motion trajectory tracking over a rigid surface. The second experiment is a cooperative manipulation assembling task, which consists in inserting a rigid part into another rigid one, for which both external and internal forces are required to be controlled, as well as the motion of the manipulated part. The third experiment consists in the manipulation of a deformable object, i.e. a basketball, using the dual-arm system of the second experiment.

A. EXPERIMENT 1: FORCE/MOTION CONTROL ON A RIGID SURFACE

The first experiment consists of a single *ABB IRB-2400* manipulator in contact with a rigid surface, as shown in Fig. 3. The task consists in following a circle over the surface, whose time-parametrized coordinates are given by

$$\boldsymbol{x}_0 = \begin{bmatrix} x_0 + r_0 \sin(2\pi t/T_0) \\ y_0 + r_0 \cos\phi_0 \cos(2\pi t/T_0) \\ z_0 + r_0 \sin\phi_0 \cos(2\pi t/T_0) \end{bmatrix},$$

where $r_0 = 40$ [mm] is the circle radius, $T_0 = 60$ [s] is the time required to complete one circle and $\phi_0 = 43.5$ [°] is the inclination of the surface with respect to the horizontal plane. At the same time, it is intended to track a desired force signal, defined by

$$F_{\mathrm{d}} = 1 + 9\left(1 - e^{-0.1\,t}\right) \text{ [N]}.$$

The gains in (23) were chosen as $\boldsymbol{K}_{\mathrm{p}} = 0.0002\boldsymbol{I}$ and $\boldsymbol{K}_{\mathrm{i}} = 0.03\boldsymbol{I}$. The sampling frequency for the control loop is 25 [Hz], constrained by the closed architecture nature of the system.

The position of the end-effector tip in the 3D Cartesian space is shown in Fig. 4. The time evolution of each of the tip Cartesian coordinates is shown in Fig. 5, while the corresponding position errors are displayed in Fig. 6. In these figures, a very good tracking can be appreciated, since the tracking error is maintained around 1 [mm], taking into account the relatively large sample time for the control loop, i.e. $T = 40$ [ms]. The force tracking and force tracking error time evolution plots are shown in Fig. 7. In this figure, the effectiveness of the approach in tracking the desired force can be appreciated.

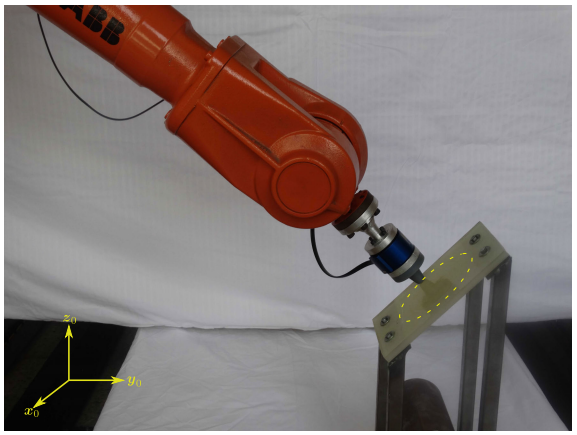

FIGURE 3. Setup for Experiment 1.

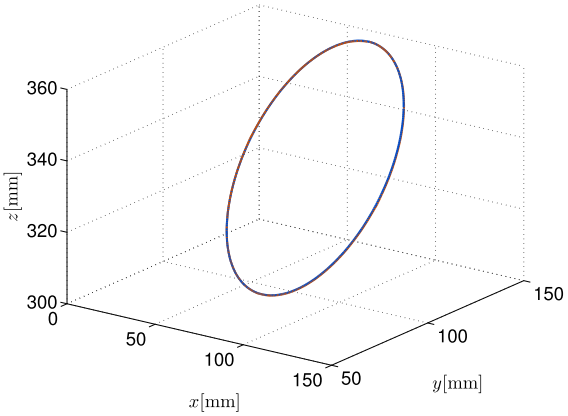

FIGURE 4. Experiment 1. Cartesian end-effector position in 3D: desired (- - -), measured (—).

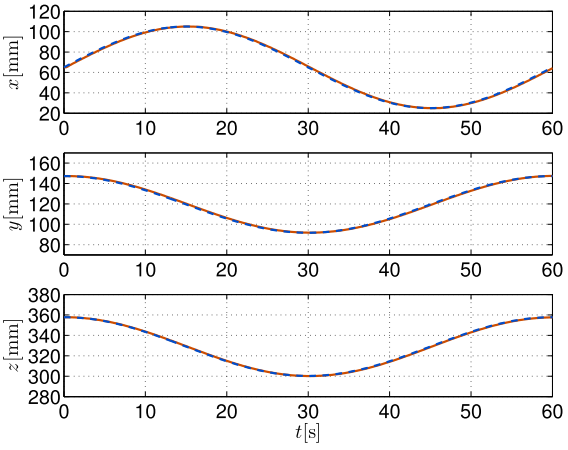

FIGURE 5. Experiment 1. Cartesian coordinates time evolution: desired (- - -), measured (—).

B. EXPERIMENT 2: COOPERATIVE ASSEMBLING TASK

The second experiment consists in a standard assembling task, where one (mobile) rigid part is manipulated to drive it near another (fixed) rigid part and push the mobile part to insert it firmly into the fixed one, as described in Fig. 8.

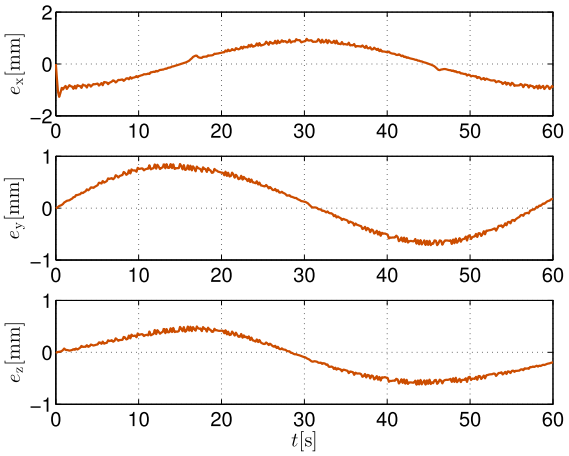

FIGURE 6. Experiment 1. Position error.

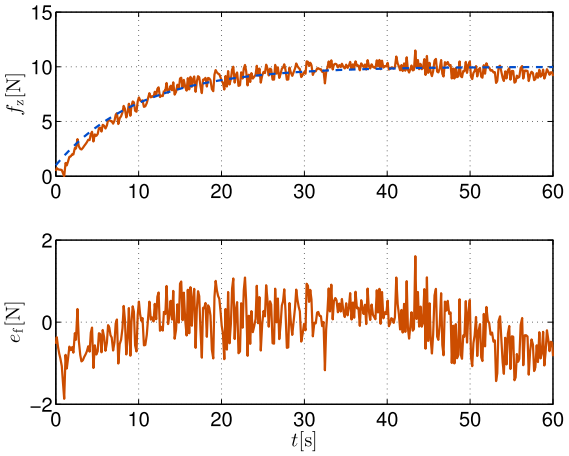

FIGURE 7. Experiment 1. Force tracking (up): desired (- - -), measured (—).Force error (down).

The experimental setup is shown in Fig. 9. In order to do the assembling task, the cooperative system must control the force necessary to grasp the object without damaging it, and simultaneously the part motion. Finally, grasping, motion, and interaction force with the fixed part must be controlled to ensure smooth assembling. To achieve this objective, a decentralized scheme is employed for the cooperative system control, i.e. the desired absolute position trajectory and relative forces of the object are designed and then the desired contact forces and motions for each manipulator are obtained through the relations explained in Section III-C.

The gains in (23) for this experiment were chosen as $\boldsymbol{K}_{\mathrm{p}} = 0.015\boldsymbol{I}$ and $\boldsymbol{K}_{\mathrm{i}} = \boldsymbol{I}$. The sample time of the force/motion control loop is $T = 40\,[\mathrm{ms}]$. The experiment is divided into three parts:

Stage 1 Grasping without movement, where the desired relative (grasping) force along the object z-axis is proposed to follow the profile

$$F_{\mathrm{rzd}} = 5 + 30\left(1 - e^{-0.5(t-t_0)}\right)\ [\mathrm{N}],$$

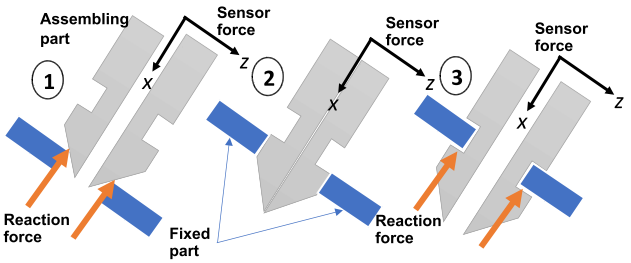

FIGURE 8. Description of the assembling task for Experiment 2.

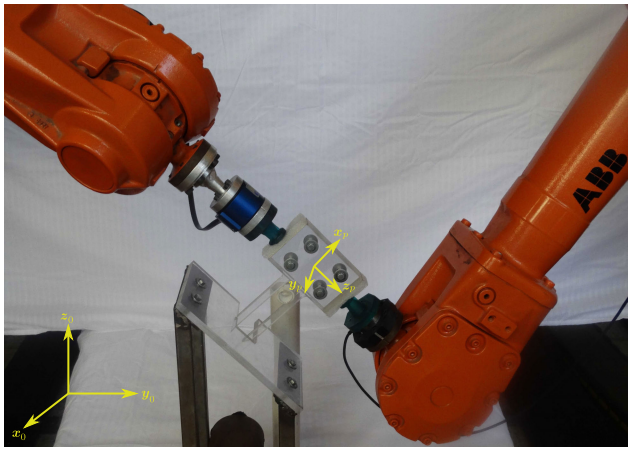

FIGURE 9. Setup for Experiment 2.

where $t_0 = 0[s]$ is the starting time of this phase. This profile starts with a relatively low force (5 [N]) and grows exponentially to its final value (35 [N]) to avoid the overshoot from the PI force control.

Stage 2 Mobile part force/motion control. The desired grasping force is maintained constant, i.e. $F_{\text{rzd}} = 35$ [N] in this stage, while the desired position trajectory is computed offline to take out the mobile part from its initial holder, then to rotate it, and finally to take it closer to the fixed part. This stage takes 120 [s], i.e. from $t_1 = 20$ [s] to $t_2 = 140$ [s].

Stage 3 Insertion. In this stage, the grasping force is still maintained constant at $F_{\text{rzd}} = 35$ [N]. There is no explicit desired motion. Instead, a desired relative force in the object x-axis direction, which is normal to the fixed part surface, is defined as

$$F_{\text{rxd}} = -4 - 16\,(t - t_2)\,/T_{\text{a}}\,[\text{N}],$$

where $T_{\text{a}} = 40$ [s] is the desired assembling task time. This profile starts with a relatively low insertion force (4 [N]) and then it is linearly increased to perform the assembly task. After the assembly, the desired force continues to increase until its maximum value of 20 [N].

A visualization of this experiment is provided in the accompanying video.

The desired and measured grasping forces for stage 1 are displayed in Fig. 10. There is a slow settling time, mainly due to the slow sampling frequency of 25 [Hz], which in turn is a consequence of the system's closed architecture. Nevertheless, the force tracking error in steady state is very good. For stage 2, the desired grasping force is maintained at $F_{\text{rzd}} = 35$ [N], while the mobile part is moved over all three Cartesian axes and rotated over the x-axis. The position tracking and the position tracking error for the center of the object are shown in Figs. 11 and 12, respectively. It must be noticed that the error in all coordinates is maintained below 1 [mm] during all the duration of this stage. Finally, in Fig. 13, the desired and real insertion force are shown. This stage starts at $t = t_2 = 140$ [s], and the pieces are sufficiently close, or even in contact, to initiate the assembling task. In Fig. 13, there are three parts: i) First, for $t < 158$ [s] the fixed part opposes the movement of the mobile part, and force tracking is achieved. ii) A sliding of the mobile part into the fixed one starts, and the contact force is drastically diminished, while the motion of the mobile part is permitted. iii) Once the mobile part has reached its final position, the fixed part opposes the movement, and force tracking is achieved again. Notice that force tracking in this component is also very good, both before and after the insertion of the mobile piece. Furthermore, the assembling task is successfully carried out as can be seen in the accompanying video.

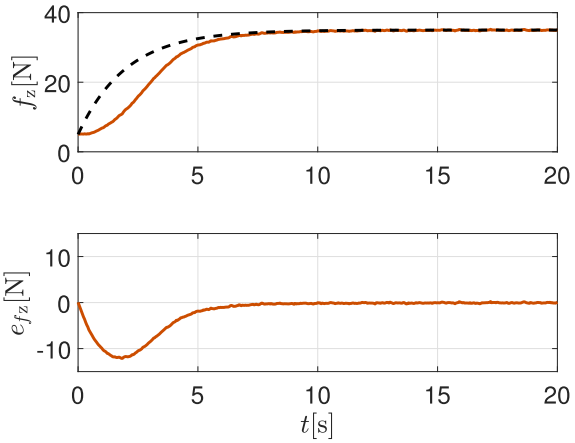

FIGURE 10. Experiment 2, stage 1. Grasping force (up): desired (- - -), measured (—). Grasping force error (down).

C. EXPERIMENT 3: SOFT ENVIRONMENT

The third experiment consists in a manipulation of a soft object, in this case, a basketball, as shown in Figure 14. The gains in (23) for this experiment were chosen as $\boldsymbol{K}_{\text{p}} = 0.015\boldsymbol{I}$ and $\boldsymbol{K}_{\text{i}} = \boldsymbol{I}$. The sampling time of the force/motion control loop is $T = 40$ [ms]. This experiment is also carried out in three stages:

Stage 1 Grasping without movement, where the desired relative (grasping) force is proposed to follow the profile

$$F_{\text{rzd}} = 35\left(1 - e^{-0.5(t-t_0)}\right)\,[\text{N}],$$

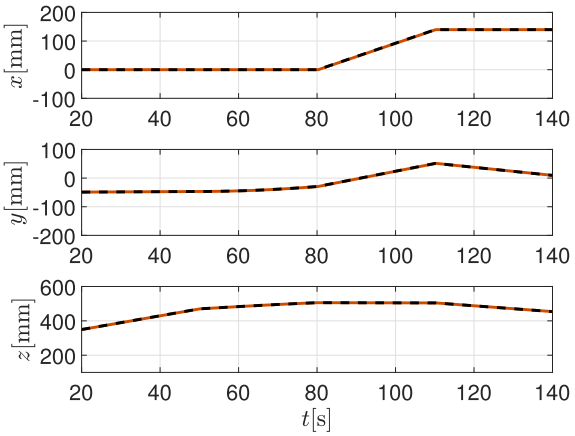

FIGURE 11. Experiment 2, stage 2. Position tracking: desired (- - -), measured (——).

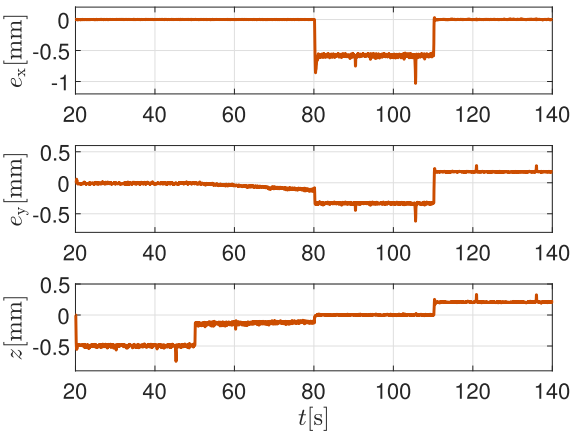

FIGURE 12. Experiment 2, stage 2. Position tracking error.

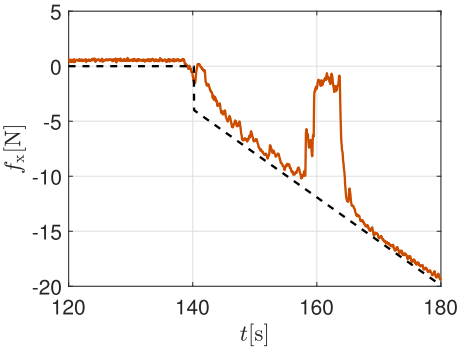

FIGURE 13. Experiment 2, stage 3. Insertion force: desired (- - -), measured (——).

where $t_0 = 0\,[\mathrm{s}]$ is the starting time for this phase. This profile starts with a zero contact force and grows exponentially to its final value (35 [N]) to avoid the overshoot arising from the PI force control. This stage takes 42 [s] to complete. The desired force value remains constant for the rest of the experiment.

FIGURE 14. Setup for Experiment 3.

Stage 2 Object rotation. The desired grasping force is kept constant, i.e. $F_{\mathrm{rzd}} = 35\,[\mathrm{N}]$ in this stage, while the object is rotated about the x_0-axis. The desired rotation angle is in turn given in two parts:

a) A linear change of the desired angle for $t_1 < t \leq t_2$

$$\vartheta_{\mathrm{d}} = -a_{\vartheta_1}(t - t_1). \tag{52}$$

b) A sinusoidal desired angle for $t_2 < t \leq t_3$

$$\vartheta_{\mathrm{d}} = -a_{\vartheta_2} + b_{\vartheta_2}\Big(1 - \cos(\omega_{\vartheta_2}(t - t_2))\Big), \tag{53}$$

where $t_1 = 42\,[\mathrm{s}]$, $t_2 = 57\,[\mathrm{s}]$, $t_3 = 102\,[\mathrm{s}]$, $a_{\vartheta_1} = 0.3\,[^\circ]$, $a_{\vartheta_2} = 4.5\,[^\circ]$, $b_{\vartheta_2} = 6\,[^\circ]$, and $\omega_{\vartheta_2} = 0.135\,[\mathrm{rad/s}]$. This stage takes 80 [s], i.e. from $t_1 = 42\,[\mathrm{s}]$ to $t_3 = 102\,[\mathrm{s}]$.

Stage 3 Object motion. In this stage, the grasping force is maintained constant at $F_{\mathrm{rzd}} = 35\,[\mathrm{N}]$, while the absolute desired position is given by a sinusoidal over the y_0-axis, given by

$$y_{\mathrm{a0}} = y_{\mathrm{ini}} + a_{\mathrm{y}}\Big(1 - cos(\omega_{\mathrm{y}}(t - t_4))\Big), \tag{54}$$

where $t_4 = 108\,[\mathrm{s}]$, $y_{\mathrm{ini}} = y(t_4)$, $a_{\mathrm{y}} = 50\,[\mathrm{mm}]$, and $\omega_{\mathrm{y}} = 0.0785\,[\mathrm{rad/s}]$. This stage takes 78 [s], i.e. from $t_1 = 108\,[\mathrm{s}]$ to $t_3 = 186\,[\mathrm{s}]$.

A visualization of this experiment is provided in the accompanying video.

The desired and measured grasping force are shown in Figure 15, as well as the corresponding error. It can be seen that for the first stage ($t \leq 42\,[\mathrm{s}]$), the force tracking is as good as for rigid surfaces. However, for stages 2 and 3, the force tracking error grows. This is due to the fact that for nonrigid surfaces position and force tracking cannot be achieved independently, as stated in Section I-A. The rotation tracking corresponding to the second stage is displayed in Figure 16, along with the rotation error. After finishing the second stage, the object rotation is no longer a control objective and it remains constant for the rest of the experiment. The position tracking corresponding to the third stage is displayed in Figure 17 along with the position tracking error.

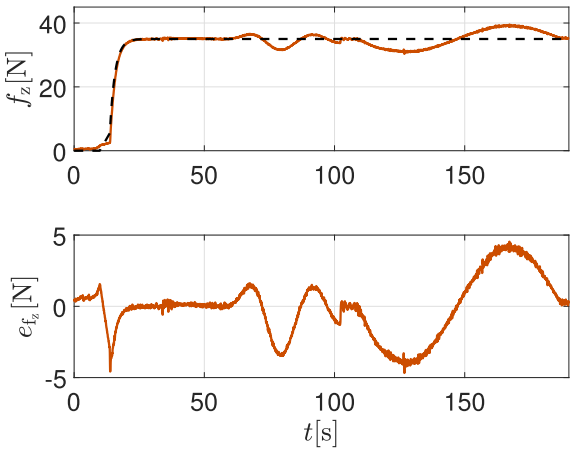

FIGURE 15. Experiment 3, stages 1-3. Grasping force (up): desired (- - -), measured (——). Force tracking error (down).

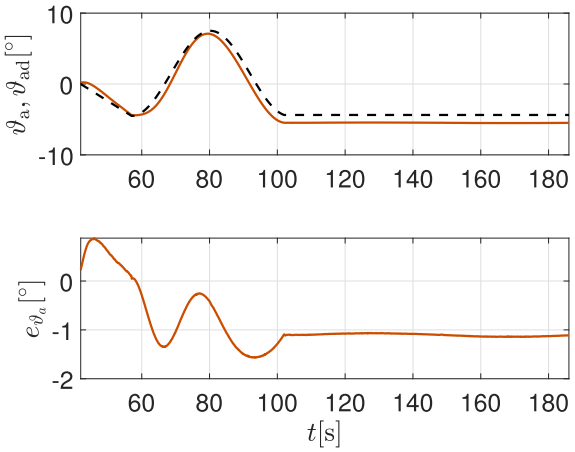

FIGURE 16. Experiment 3, stage 2. Rotation angle ϑ (up): desired (- - -), measured (——). Rotation error (down).

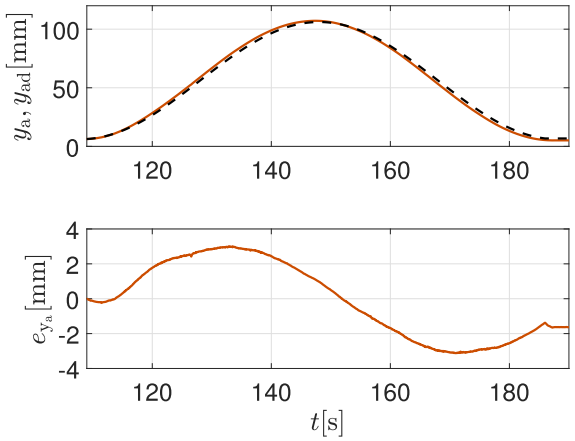

FIGURE 17. Experiment 3, stage 3. Position control along the y_0-axis (up): desired (- - -), measured (——). Position tracking error (down).

It can be noticed that the tracking error is larger than those present in the experiments with rigid surfaces, but still under 4 [mm].

D. PERFORMANCE INDEXES

In this section, a quantitative measurement of the controller performance is given by employing some well known norms, listed as follows:

- Root-Mean-Square Error (RMSE), defined as

$$\|\boldsymbol{e}_k\|_{RMSE} = \sqrt{\frac{\sum_i e_{ki}^2}{n_s}},$$

  where n_s is the total number of samples of a signal $\boldsymbol{e}_k$ and $\bar{e}_k$ its mean value.
- Mean Absolute Error (MAE), defined as

$$\|\boldsymbol{e}_k\|_{MAE} = \frac{\sum_i |e_{ki}|}{n_s}.$$

- $\mathcal{L}_\infty$-norm, defined as

$$\|\boldsymbol{e}_k\|_{\mathcal{L}_\infty} = \max_i\{|e_{ki}|\}.$$

These indexes are displayed in Table 1 for the error signals of all three experiments. Moreover, the percentages of the performance indexes with respect to the maximum change of the reference signal are given in Table 2. The position errors are smaller for the experiments with rigid surfaces (experiments 1 and 2), and larger for the nonrigid object (experiment 3), which is in good accordance with the stated in Propositions 1 and 2.

TABLE 1. Performance indexes for the position and force errors.

Experiment	Error	RMSE	MAE	$\mathcal{L}_\infty$	Units
1	$\boldsymbol{e}$	0.9040	0.9010	1.2708	mm
	e_{f_z}	0.5855	0.4789	1.8679	N
2	$\boldsymbol{e}$	0.4540	0.3978	1.2021	mm
	e_{f_z}	4.1275	2.0606	12.1473	N
	e_{f_x}	3.3133	1.7894	12.5179	N
3	e_{y_a}	2.1529	1.9017	3.1248	mm
	e_ϑ	1.0498	0.9965	1.5648	deg
	e_{f_z}	2.0499	1.4998	4.6786	N

TABLE 2. Performance indexes as a percentage of the maximum change in reference signals.

Experiment	Error	RMSE	MAE	$\mathcal{L}_\infty$	Units
1	$\boldsymbol{e}$	0.7990	0.7964	1.1233	%
	e_{f_z}	6.5215	5.3338	20.8053	%
2	$\boldsymbol{e}$	0.1948	0.1707	0.5157	%
	e_{f_z}	13.7591	6.8690	40.4930	%
	e_{f_x}	16.5770	8.9524	62.6283	%
3	e_{y_a}	1.7940	1.5847	2.6040	%
	e_ϑ	8.7481	8.3040	13.0404	%
	e_{f_z}	5.8570	4.2851	13.3674	%

V. DISCUSSION

As validated through the above experiments, the proposed approach makes it possible to follow references in both force and position for industrial robots interacting with rigid and soft environments. The control objective is reached by employing only position and velocity references, which are in

turn managed by the motion planner. Nevertheless, the standard motion planners are intended for offline designed paths. This means that although the inner control loop is running at a fast sample rate, the outer loop runs at a slower rate. In the present case, the outer loop, in which the force controller is implemented, is running at 25[Hz]. Such a slow sample rate limits the motion velocity of the overall system, which results in the relatively slow motions of the experiments.

The proposed approach assumes that the environment is completely known, i.e. the geometry of the parts to be manipulated is well characterized and the kinematics of the manipulators is well-calibrated. Although the second is a realistic assumption, the first one limits possible applications. Online estimation of the environment shape has been investigated for rigid surfaces [32], [33] by employing force measurements, and for nonrigid surfaces [6] by estimating the environment stiffness in an indirect force control scheme. Computer vision-based solutions have also been explored for both rigid [34] and nonrigid [35] environments as well. In the context of the present work, it is not clear if the limitations of the closed-loop architecture could obviate the implementation of such solutions and it is left as future research.

It is well known that force overshoots can appear when interacting with rigid and nonrigid surfaces. For the latter, the compliance nature of indirect force controllers can be of help to reduce such overshoots with the aid of force sensor measurements as in [36]. For rigid surfaces, the problem is more involved and it is an active topic of research. As a matter of fact, to avoid unstable behaviors when interacting with rigid environments, the robot end-effector must contact the surface with zero velocity [37], which requires a well calibrated structured environment. In the experiments presented in Section IV, force overshoots were avoided because of: i) good setting of the control gains and good knowledge of the environment kinematics, ii) slow-motion velocities before contact, and iii) reference forces are exponential signals starting at low values.

The experimental results of Section IV are in accordance with the theory presented in Section III. As can be seen in the graphics as well as in the performance indexes displayed in Tables 1–2, the best performance in position control is obtained for rigid surfaces, whereas a trade-off between force and position performance must be made for deformable surfaces. The relatively low performance of force control for all experiments is not uncommon for these kind of controllers and it is due to several factors such as sensor dynamics and delays introduced by the data acquisition, among others.

VI. CONCLUSION

The approach proposed in this work makes it possible to follow trajectories in both position and force for industrial robots with closed control architecture. It exclusively employs the standard built-in motion controller, which can help to reduce costs for tasks requiring force control. The proposed is suitable for both single and dual-arm manipulation systems with rigid and nonrigid surfaces. Furthermore, the convergence of the force and position signals to their reference values is guaranteed by the mathematical analysis developed in this work, while its precision depends on that of the controller provided by the manufacturer. As future work, an online identification algorithm of the geometry for uncalibrated environments will be pursued. The attenuation of oscillations and force overshoots arising from interaction control will be studied as well. Finally, a modification of the proposed scheme for the case of master-slave teleoperation will be proposed.

ACKNOWLEDGMENT

The authors would like to thank to Jesús Bañuelos and Andrés González for their help in constructing the experimental setup.

REFERENCES

[1] B. Siciliano and L. Villani, *Robot Force Control*. Dordrecht, The Netherlands: Kluwer, 1999.

[2] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Heidelberg, Germany: Springer, 2016.

[3] H. Wang, W. Ren, C. C. Cheah, Y. Xie, and S. Lyu, "Dynamic modularity approach to adaptive control of robotic systems with closed architecture," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2760–2767, Jun. 2020.

[4] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and J. Wang, "Extending an industrial robot controller: Implementation and applications of a fast open sensor interface," *IEEE Robot. Autom. Mag.*, vol. 12, no. 3, pp. 85–94, Sep. 2005.

[5] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2010, pp. 554–561.

[6] L. Roveda, A. A. Shahid, N. Iannacci, and D. Piga, "Sensorless optimal interaction control exploiting environment stiffness estimation," *IEEE Trans. Control Syst. Technol.*, early access, Mar. 5, 2021, doi: 10.1109/TCST.2021.3061091.

[7] A. Gutiérrez-Giles and M. Arteaga-Pérez, "Output feedback hybrid force/motion control for robotic manipulators interacting with unknown rigid surfaces," *Robotica*, vol. 38, no. 1, pp. 136–158, Jan. 2020.

[8] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. Robot. Autom.*, vol. 9, no. 4, pp. 361–373, Aug. 1993.

[9] G. P. Incremona, G. De Felici, A. Ferrara, and E. Bassi, "A supervisory sliding mode control approach for cooperative robotic system of systems," *IEEE Syst. J.*, vol. 9, no. 1, pp. 263–272, Mar. 2015.

[10] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—A survey," *Robot. Auton. Syst.*, vol. 60, no. 10, pp. 1340–1353, 2012.

[11] C.-Y. Weng, W. C. Tan, and I.-M. Chen, "A survey of dual-arm robotic issues on assembly tasks," in *ROMANSY 22—Robot Design, Dynamics and Control* (CISM International Centre for Mechanical Sciences), vol. 584, V. Arakelian and P. Wenger, Eds. Cham, Switzerland: Springer, 2019, pp. 474–480.

[12] S. Erhart and S. Hirche, "Internal force analysis and load distribution for cooperative multi-robot manipulation," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1238–1243, Oct. 2015.

[13] I. D. Walker, R. A. Freeman, and S. I. Marcus, "Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 396–409, Aug. 1991.

[14] J. Hernandez-Barragan, C. Lopez-Franco, A. Y. Alanis, N. Arana-Daniel, and M. Lopez-Franco, "Dual-arm cooperative manipulation based on differential evolution," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 1, pp. 1–20, 2019.

[15] R. Monfaredi, S. M. Rezaei, and A. Talebi, "A new observer-based adaptive controller for cooperative handling of an unknown object," *Robotica*, vol. 34, no. 7, pp. 1–27, 2014.

[16] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-DOF impedance control of dual-arm cooperative manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 5, pp. 576–586, Oct. 2008.
[17] H. Farivarnejad and S. A. A. Moosavian, "Multiple impedance control for object manipulation by a dual arm underwater vehicle–manipulator system," *Ocean Eng.*, vol. 89, pp. 82–98, Oct. 2014.
[18] J. Lee, P. H. Chang, and R. S. Jamisola, "Relative impedance control for dual-arm robots performing asymmetric bimanual tasks," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3786–3796, Jul. 2014.
[19] J. K. Behrens, R. Lange, and M. Mansouri, "A constraint programming approach to simultaneous task allocation and motion scheduling for industrial dual-arm manipulation tasks," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 8711–8795.
[20] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 498–510, Aug. 2002.
[21] W. Gueaieb, F. Karray, and S. Al-Sharhan, "A robust hybrid intelligent position/force control scheme for cooperative manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 109–125, Apr. 2007.
[22] J. Pliego-Jiménez, M. A. Arteaga-Pérez, and C. Cruz-Hernández, "Dexterous remote manipulation by means of a teleoperation system," *Robotica*, vol. 37, no. 8, pp. 1457–1476, Aug. 2019.
[23] Z. Chen, F. Huang, W. Chen, J. Zhang, W. Sun, J. Chen, J. Gu, and S. Zhu, "RBFNN-based adaptive sliding mode control design for delayed nonlinear multilateral telerobotic system with cooperative manipulation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1236–1247, Feb. 2020.
[24] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Karlsruhe, Germany, May 2013, pp. 4000–4007.
[25] E. Mariotti, E. Magrini, and A. D. Luca, "Admittance control for human-robot interaction using an industrial robot equipped with a F/T sensor," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 6130–6136.
[26] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, U.K.: Springer-Verlag, 2010.
[27] Y. H. Liu and S. Arimoto, "Implicit and explicit force controllers for Rheo–holonomically constrained manipulators and their extension to distributed cooperation control," in *Proc. IFAC 13th Triennial World Congr.*, San Francisco, CA, USA, 1996, pp. 618–623.
[28] R. Kelly, V. Santibáñez, and A. Loria, *Control of Robot Manipulators in Joint Space* (Advanced Textbooks in Control and Signal Processing). London, U.K.: Springer-Verlag, 2005.
[29] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 268–286.
[30] C.-T. Chen and C.-T. Chen, *Linear System Theory and Design*, vol. 301. New York, NY, USA: Holt, Rinehart and Winston, 1984.
[31] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, no. 6, pp. 879–887, Jun. 2000.
[32] Z. Doulgeri and Y. Karayiannidis, "Force/position regulation for a robot in compliant contact using adaptive surface slope identification," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2116–2122, Oct. 2008.
[33] J. Pliego-Jiménez and M. A. Arteaga-Pérez, "Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters," *Eur. J. Control*, vol. 22, pp. 1–12, Mar. 2015.
[34] A. C. Leite, F. Lizarralde, and L. Hsu, "Hybrid adaptive vision/force control for robot manipulators interacting with unknown surfaces," *Int. J. Robot. Res.*, vol. 28, no. 7, pp. 911–926, 2009.
[35] V. Lippiello, B. Siciliano, and L. Villani, "A position-based visual impedance control for robot manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2007, pp. 2068–2073.
[36] L. Roveda, N. Pedrocchi, M. Beschi, and L. M. Tosatti, "High-accuracy robotized industrial assembly task control schema with force overshoots avoidance," *Control Eng. Pract.*, vol. 71, pp. 142–153, Feb. 2018.
[37] S. S. M. Salehian and A. Billard, "A dynamical-system-based approach for controlling robotic manipulators during noncontact/contact transitions," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2738–2745, Oct. 2018.

ALEJANDRO GUTIERREZ–GILES received the Ph.D. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), in 2016. He was a Visiting Researcher with the Prisma Laboratory, University of Naples Federico II, Naples, Italy, from 2016 to 2018. Since August 2018, he has been a Visiting Researcher with the Center of Research and Advanced Studies, National Polytechnic Institute (Cinvestav–IPN), Mexico City, Mexico. He is currently a part of the National Researchers System of Mexico (SNI). His research interests include robotics, nonlinear control, observers, and electromechanical systems.

LUIS U. EVANGELISTA–HERNANDEZ received the B.E. degree in mechanical engineering from the Technological Institute of Orizaba, Mexico, in 2015, and the M.Sc. degree in electrical engineering from the Center for Research and Advanced Studies (CINVESTAV), Mexico, in 2019.

MARCO A. ARTEAGA was born in Huejutla de Reyes, Hidalgo, Mexico, in September 1967. He received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), in 1991 and 1993, respectively, and the Dr.-Ing. degree from Gerhard-Mercator University, Duisburg, Germany. Since 1998, he has been a Professor with the Department of Electrical Engineering, School of Engineering, National University of Mexico. His main research interests include robotics and control.

CARLOS A. CRUZ–VILLAR (Member, IEEE) received the B.E. degree in electronics engineering from the National Polytechnic Institute (IPN), Mexico, in 1996, and the Dr.Sc. degree from the Center for Research and Advanced Studies (CINVESTAV) Mexico, in 2001. In 2000, he was as a Lecturer and the Head of the Department of Basic Sciences, UPIITA-IPN. Since 2001, he has been a Full Professor with the Mechatronics Section, Department of Electrical Engineering, CINVESTAV. His research interests include the optimal design of mechatronic systems and optimal control.

ALEJANDRO RODRIGUEZ–ANGELES received the B.Sc. degree in communications and electronics from the National Polytechnic Institute (IPN), Mexico, in 1995, the M.Sc. degree in electrical engineering from the Center for Research and Advanced Studies (CINVESTAV-IPN), Mexico, in 1997, and the Ph.D. degree in mechanical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2002. From 1998 to 2000, he was a Research Assistant with the Department of Applied Mathematics, University of Twente, Enschede, The Netherlands. From 2000 to 2002, he was a Research Assistant with the Eindhoven University of Technology. From 2003 to 2005, he was with the Mexican Institute of Petroleum, Mexico. Since 2005, he has been a Researcher with the Mechatronics Group, CINVESTAV-IPN. His research interests include synchronization, nonlinear control, model-based control, optimization, robotic systems, and supply chain systems.