IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 13, 2021, accepted July 19, 2021, date of publication July 26, 2021, date of current version August 2, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3099131*

# Optimal Control of Robotic Systems Using Finite Elements for Time Integration of Covariant Control Equations

**JUAN ANTONIO ROJAS-QUINTERO[1], JORGE VILLALOBOS-CHIN[2], AND VICTOR SANTIBANEZ[2], (Member, IEEE)**

[1]CONACYT/Tecnológico Nacional de México/I. T. Ensenada, Ensenada 22780, México
[2]Tecnológico Nacional de México/I. T. La Laguna, Torreón 27000, México

Corresponding author: Juan Antonio Rojas-Quintero (jarojas@conacyt.mx)

This work was supported by in part by the Secretaría de Educación Pública (SEP) and Consejo Nacional de Ciencia y Tecnología (CONACYT) under Grant A1-S-29824, and in part by the Tecnológico Nacional de México (TecNM) projects.

**ABSTRACT** We used an optimal control method involving covariant control equations as optimality conditions, to command the actuators of robot manipulators. These form a coupled system of second order nonlinear ordinary differential equations when associated with the robot motion equations. By solving this system, the control action required to take the robot from an initial to a final state is optimized in a prescribed time. However, the target set of equations exhibited stiffness. Therefore, an adequate solution could only be found for short trajectory durations with readily available numerical methods. We examined a time discretization procedure based on cubic and quintic Hermite finite elements which exhibited superconvergence properties for interpolation. This motivated us to develop a time integration algorithm based on Hermite's technique, where motion and control equations were perturbed to solve the optimal control problem. The optimal motion of a robotic manipulator was simulated using this algorithm. Our method was compared with a commercial differential equations solver on the basis of specific indicators. It outperformed the commercial solver by effectively solving the stiff set of equations for longer trajectory durations, with the cubic elements performing better than the quintic ones in this sense. The convergence analysis of our method confirmed that the quintic elements are more precise at the cost of increased computational burden, but converge at a lower rate than expected. Controlled motion experiments on a robotic manipulator validated our methodology. Trajectories were smoothly tracked and results exposed further methodology improvements.

**INDEX TERMS** Differential equations, finite element methods, nonlinear dynamical systems, numerical simulation, optimal control, robot control.

## I. INTRODUCTION

The finite element method (FEM) has become a favored technique for solving engineering problems relating to the estimation of stress, strain or wear in elastic materials (see references [1]–[5] for some examples). Its origins can be linked to the works of Richard Courant back in 1943 [6]. However, it has been advocated that the FEM can easily be used as a general approximation method for solving boundary-value problems that arise in science and engineering applications [7], [8]. The basic idea lies in subdividing a system, regarded as the domain, into smaller and geometrically simpler subdomains that are called finite elements to achieve spatial discretization. The solution of a particular differential equation is then approximated over each finite element, regarded as a domain for which boundary values should be met. The whole system is modeled by assembling all of the finite elements into a system of algebraic equations.

The FEM has proved to be computationally accurate, efficient and stable [7]–[11]. It is therefore natural that the FEM has found applications in the areas of systems control and robotics. In recent years, FEM has been widely used in the modeling and control of distributed parameter systems. The method became important in this field, where systems can be modeled using partial differential equations (PDE), and the FEM provides a numerical solution that is useful

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

for, among other things, evaluating the performance of the controllers. For example, the finite element (FE) formulation presented in [1], is used to achieve the active bending and torsional control of functionally gradient material plates that are subject to a thermal gradient. In this work, the bending and torsional vibration suppression of the plates are accomplished by using a feedback algorithm. Another good example may be [12], where an iterative procedure is proposed to update the FE model of a system. The poles and zeros of a frequency response reduced order model are determined from a measured frequency response function. The obtained model is applicable to the determination and control of unmeasurable performance variables, which is relevant in the context of model based control. The robust output regulation of linear boundary control systems is studied in [13]. Linear finite elements were used to approximate solutions to sets of PDE such as convection and beam equations. The study held in [14] presents a simulation framework that uses the output signals retrieved from a FEM-based solution of heat flow equations. This would allow for the testing of different control strategies on an additive manufacturing process in a virtual environment.

Soft robotics is a field where the FEM has found applications for the analysis and control of flexible link manipulators. For example, a numerical integration strategy based on a FEM with a numerical optimization based on Lagrange multipliers was conducted in [3]. This was done to obtain the forward and inverse kinematic models of soft manipulators in order to perform position control. The FEM proved useful since the formulation of the problem heavily relied in the use of continuum mechanics. A nonlinear finite element-based procedure was studied in [2] and applied to the control of a single-link flexible manipulator with hydraulic actuation. This was achieved by using what the authors called Virtual Decomposition Control which enabled the handling of the dynamics and control of the hydraulic actuator and the flexible link separately. As a more traditional application, the FEM was also used to assist the design of soft robots, for example, to model their dynamics and predict the controllability of the points of interest [15]. In contrast with the approach presented in this document, in all of these previous examples the FEM is employed to discretize space as opposed to time, using beam-like models to estimate link deformation during motion.

An early development involving the FEM for time discretization in the context of optimal control was proposed by Cavin and Tandon [16] where heat PDE subject to boundary conditions were solved using a space-time finite element formulation. Later, a space-time finite element approximation scheme for the optimal control problem (OCP) of systems governed by fractional order PDE was proposed in [17]. The approach considers piecewise constant discontinuous Galerkin time discretization and linear finite elements for spatial discretization of the state equation. Nonetheless, only a few works make use of the FEM to discretize time in the context of multibody systems dynamics. Control of general rigid multibody systems commonly involves nonlinear problem formulations. In this context, shooting procedures are instead commonly used [11], [18]. However, a finite element description has proven to be computationally advantageous for the direct integration of the dynamic equations governing multibody systems motions. For example, finite elements have previously been used in [11], [19] for the simulation of multibody systems motion, to solve nonlinear differential equations arising in biomechanics and robotics. It was shown that the required number of steps and CPU-time are lower for such a FEM-based algorithm when compared to popular Runge-Kutta (fourth order) and Newmark methods. One of the advantages of the FEM is that boundary configurations are exactly satisfied in the solution. This property is particularly relevant in robotics applications. These promising results led us to develop a time integration method based in the Hermite finite elements piecewise basis functions, to solve the OCP of a robot manipulator.

Optimal control methodologies are aimed at finding system controls that optimize a predefined performance index (usually also called objective function) according to specific restrictions. Traditionally speaking, solutions to OCP tend to fall into two main frameworks: Pontryagin's Maximum Principle (PMP) and the Bellman approach which led to Dynamic Programming (DP). Generally, PMP-related methods involve ordinary differential equations (ODE) whereas DP-based methods involve PDE. PMP-based methods are therefore at a numerical advantage and are often easier to use than DP-based methods in order to solve OCP [20]. With these two methodologies, it is possible to perform optimal path planning where the optimal controls, according to a specific performance index, are obtained. Alternatively, the Linear Quadratic Regulator (LQR), which defines a class of optimal control methods, operates differently by finding the controller parameters (usually gains) that will optimize a specified performance index [21]. It should be noted however that the LQR methodology is specifically designed for linear systems because optimal controller parameters are obtained through the solution of a linear Riccati equation. Therefore, a much more general approach can be found in what is known as State-Dependent Riccati Equation (SDRE) control. SDRE control also requires the solution of a linear Riccati equation to find optimal controller parameters. However, nonlinearities are fully captured and dealt with through extended linearization [22], [23].

In this work, we propose a FE-based method which can be applied to obtain the solutions to OCP in which stiff ODE arise, which is usually the case for PMP-related developments [24]. Due to the FE nature of our method, it is not excluded that our algorithm can be adapted to solve PDE that arise in DP. We will focus in the ODE case of the optimal control methodology described in [25] which involves covariant control equations. This methodology is similar to PMP, however, the resulting set of nonlinear ODE are of the second order instead, but also exhibit stiffness. The covariant control equations involved, act as control restrictions that

bound their intensities. It has been shown that these covariant control equations define adjoint states in the framework of PMP [26] thus rendering them compatible with a wide range of optimal control solution methods. Our proposed FE-based algorithm should also be of interest for solving ODE that arise in other optimal path planning and kinematics control strategies [27], [28].

The organization of the paper is the following. First, we will establish the selected optimal control methodology in section II. This methodology results in a two-point boundary-value problem (TPBVP) and involves covariant control equations which are conjugate to the robot motion equations, composing a system of second order ODE (see section II). The solution of this TPBVP provides the optimal path along with optimal torques required by such path. We will then present a time discretization procedure based on Hermite finite elements piecewise basis functions in section III. There, two simple examples will illustrate the technique benefits and accuracy for our purposes. A time integration procedure will also be proposed, followed by a one dimensional nonlinear example. Next, we carry out optimal motion simulations in section IV. For this purpose, we begin by proposing an optimization procedure in the form of an algorithm, which involves the perturbation of motion and control equations to find a suitable finite element-based approximation. Optimal motion simulation examples are conducted to compare our method with a commercial ODE solver. It is shown that our method outperforms this solver by being able to produce longer simulations. We conclude section IV by performing a convergence analysis of our proposed method. Our simulations are then followed by controlled motion experiments conducted on a robotic serial manipulator in section V. We present our control scheme in this section. Additionally, we carry out a robustness analysis of our control strategy. The conducted experiments verify the tractability of our approach and allow us to identify ways to improve our optimal control methodology.

## II. OPTIMAL CONTROL METHODOLOGY

We have followed the optimal control methodology presented in [25]. Therefore, this section summarizes the main ideas of the cited methodology. The reader is referred to the previously mentioned document for more details. First, the dynamics of a robotic manipulator are derived. The optimal control of such a system then consists in minimizing an integral functional of the joint torques. This procedure leads to the establishment of covariant control equations regarded as optimality conditions.

### A. SYSTEM DYNAMICS

We consider serial robotic manipulators with $n$ degrees of freedom (DOF) which are described by $n$ configuration parameters $q^i$, and operated by $n$ actuators placed between two consecutive segments. In the following, tensor notation and the Einstein summation convention on repeated indices [29] will be used to establish the robot dynamics.

The robot velocity field is a linear function of the variables $q^i$ time derivatives $\dot{q}^i$ (where a superposed dot shall denote a time differential). The kinetic energy function $K(q, \dot{q})$ is quadratic and strictly convex. It is defined with the positive definite Hessian $M(q)$, commonly called mass tensor, which is regarded as a Riemannian metric with components $M_{ij}$. It expresses as

$$K = \frac{1}{2} M_{ij} \dot{q}^i \dot{q}^j. \tag{1}$$

A gravitational potential $V(q)$, defined as the product between mass, local gravitational field intensity, and center of mass height models gravity actions. A torque $u_i$ is exerted between two consecutive links by the $i^{\text{th}}$ actuator. For any virtual variation $\delta q^i$, the actuators virtual work $u_i \delta q^i$ is invariant, therefore $u_i$ are the covariant components of a torque tensor.

Robot motion is governed by the $n$ Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial K}{\partial \dot{q}^i}\right) - \frac{\partial K}{\partial q^i} = u_i - \frac{\partial V}{\partial q^i},$$

which upon using (1), lead to the explicit expression

$$M_{ij}\ddot{q}^j + \Gamma_{ikl}\dot{q}^k\dot{q}^l + \nabla_i V = u_i. \tag{2}$$

where $\nabla_i V = \frac{\partial V}{\partial q^i}$ is the $i^{\text{th}}$ covariant derivative component of the potential $V$ and $\Gamma_{ikl}$ are Christoffel symbols of the first kind which derive from the $M_{ij}$ components by

$$\Gamma_{ikl} = \frac{1}{2}\left(\frac{\partial M_{ik}}{\partial q^l} + \frac{\partial M_{il}}{\partial q^k} - \frac{\partial M_{kl}}{\partial q^i}\right). \tag{3}$$

Let us introduce the coefficients of the inverse metric tensor $M^{ij}$, which lead to the Christoffel symbols of the second kind when multiplied by (3)

$$\Gamma^j_{kl} = M^{ji}\Gamma_{ikl}. \tag{4}$$

Invoking the above, equations (2) transform into

$$M_{ij}\left(\ddot{q}^j + \Gamma^j_{kl}\dot{q}^k\dot{q}^l + M^{jk}\nabla_k V\right) = u_i,$$

where the term $M^{jk}\nabla_k V = \nabla^j V$ is the $j^{\text{th}}$ contravariant derivative component of potential $V$. Upon renaming indices in the above equation, the contravariant torque tensor components reveal

$$\ddot{q}^i + \Gamma^i_{jk}\dot{q}^j\dot{q}^k + \nabla^i V = u^i. \tag{5}$$

Note that these contravariant torque components were retrieved by following the rules of tensor calculus in a Riemannian manifold, where $u^i = M^{ik}u_k$.

### B. OPTIMAL CONTROL

The required torque intensities to bring the robotic system from an initial state $x_0 = x(0)$, to a final state $x_1 = x(T)$ in a prescribed time $T$, can be constrained by minimizing an integral functional of the type:

$$J(u) = \int_0^T \gamma(u(t))\,\mathrm{d}t.$$

This integral functional is regarded as the performance index. The integrand $\gamma$ is commonly called cost function and is generally chosen to be convex so that the integral functional $J$ is reasonable to minimize. It has been noticed that $\gamma$ is often chosen so that

$$\gamma = \frac{1}{2} A_{ij} u_i u_j,$$

where $A_{ij}$ is a diagonal matrix with constant coefficients, usually unitary [11], [30]–[33]. However such a cost function is not invariant under a change of coordinates. Instead, as proposed in [25], we select the invariant cost function $\gamma = \frac{1}{2} u_i u^i = \frac{1}{2} M^{ij} u_i u_j$ so that the performance index becomes

$$J(u) = \int_0^T \frac{1}{2} u_i u^i \, \mathrm{d}t, \tag{6}$$

where $u_i$ and $u^i$ are respectively expressed by equations (2) and (5) and are functions of parameters $q^i$, $\dot{q}^i$ and $\ddot{q}^i$. Note that $\gamma$ is selected to be a convex and invariant cost function, involving the inverse mass tensor components $M^{ij}$, to which the Euler-Lagrange equations can be applied as

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\left(\frac{\partial \gamma}{\partial \ddot{q}^i}\right) - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \gamma}{\partial \dot{q}^i}\right) + \frac{\partial \gamma}{\partial q^i} = 0. \tag{7}$$

By applying the rules of tensor calculus, equations (7) transform into the covariant control equations

$$\frac{\hat{\mathrm{d}}^2 u_i}{\mathrm{d}t^2} + R^k_{jil} \dot{q}^j \dot{q}^l u_k + \nabla_i \nabla_l V \, u^l = 0. \tag{8}$$

The reader is referred to [25], where the complete analysis of how equations (7) lead to (8), is conducted. These control equations are nonlinear second order ODE that restrict the torques during motion. They are regarded as optimality conditions, and it has been shown that these equations define the adjoint states in the framework of PMP [26]. The left hand side of these equations involve: the second covariant time derivative of the covariant torque tensor components $u_i$ in the first term (see equation (2)); the Riemann-Christoffel curvature tensor in the second term; the second covariant derivative of potential $V$ in the third term; and contravariant torque tensor components $u^l$ in the third term (see equation (5)). Definitions for these objects can be found in [25], [29] and have the following expressions:

$$\begin{aligned}
\frac{\hat{\mathrm{d}}^2 u_i}{\mathrm{d}t^2} &= \ddot{u}_i - \left(\frac{\partial \Gamma^k_{ij}}{\partial q^l} - \Gamma^m_{ji}\Gamma^k_{ml}\right) u_k \dot{q}^j \dot{q}^l \\
&\quad - \Gamma^k_{ij}\left(u_k \ddot{q}^j + 2\dot{u}_k \dot{q}^j\right); \\
R^k_{jil} &= \frac{\partial \Gamma^k_{jl}}{\partial q^i} - \frac{\partial \Gamma^k_{ji}}{\partial q^l} + \Gamma^m_{jl}\Gamma^k_{mi} - \Gamma^m_{ji}\Gamma^k_{ml}; \\
\nabla_i \nabla_l V &= \frac{\partial^2 V}{\partial q^l \partial q^i} - \Gamma^m_{il}\frac{\partial V}{\partial q^m}.
\end{aligned} \tag{9}$$

Control equations (8) can be associated to the motion equations (2), thus providing a coupled system of $2n$ second order nonlinear ODE in the dual variables $(q^i, u_i)$ for resolving the optimal trajectories and torques:

$$M_{ij}\ddot{q}^j + \Gamma_{ikl}\dot{q}^k\dot{q}^l + \nabla_i V - u_i = 0 \tag{10a}$$

$$\frac{\hat{\mathrm{d}}^2 u_i}{\mathrm{d}t^2} + R^k_{jil}\dot{q}^j\dot{q}^l u_k + \nabla_i\nabla_l V \, u^l = 0. \tag{10b}$$

Solving the system of equations (10) minimizes the performance index (6). Note that $q^i$ and $u_i$ are independent variables and that $4n$ boundary conditions are required. These are generally known values of initial and final positions and velocities: $q^i(0)$, $q^i(T)$, $\dot{q}^i(0)$ and $\dot{q}^i(T)$, where $T$ denotes the prescribed trajectory duration. Using these boundary constraints to solve the ODE system (10), the initial OCP transforms into a TPBVP. Note also that no boundary conditions are imposed on $u_i$ parameters, which are regarded as conjugate parameters that directly provide motion torque requirements.

## III. TIME INTEGRATION METHOD

In order to solve the nonlinear ODE system (10) in an interval $[0, T]$, we perform a time finite element discretization. This means that the solution domain is discretized and represented as a mesh of time elements. The unknown variables (configurations and controls in our case) behavior is approximated over each element by continuous functions expressed in terms of the nodal values of the unknown and its time derivatives [16] (more on this along the section). Hermite piecewise functions [9] are defined over each time finite element. The collection of these interpolation functions for the whole domain provides a piecewise approximation to the sought variables.

### A. PIECEWISE HERMITE FINITE ELEMENTS INTERPOLATION

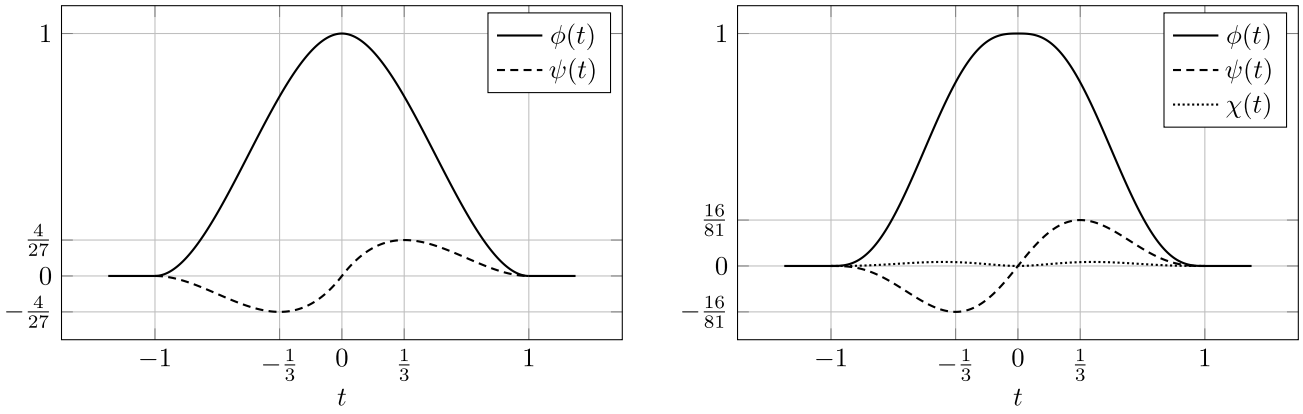
Hermite finite elements (HFE) are compact piecewise continuous functions that can be connected in a continuously differentiable way. Therefore, the values of their derivatives are also used to characterize the functions that they approximate [9]. This property is particularly useful for our approach because the targeted optimal control methodology requires a continuously differentiable solution. Cubic HFE (CHFE) are formed by functions $\phi$ and $\psi$, shown in Figure 1a, and defined by

$$\phi(t) = \begin{cases} (1-|t|)^2\,(1+2\,|t|) & \forall\, |t| \le 1 \\ 0 & \forall\, |t| > 1; \end{cases}$$

$$\psi(t) = \begin{cases} t\,(1-|t|)^2 & \forall\, |t| \le 1 \\ 0 & \forall\, |t| > 1. \end{cases}$$

One can construct higher order HFE functions by following prescriptions found in [10]. For example, quintic HFE (QHFE, using fifth degree polynomials) are formed by functions $\phi$, $\psi$ and $\chi$, shown in Figure 1b, and defined by

$$\phi(t) = \begin{cases} (1-|t|)^3\left(1+3|t|+6t^2\right) & \forall\, |t| \le 1 \\ 0 & \forall\, |t| > 1; \end{cases}$$




(a) Cubic (b) Quintic

**FIGURE 1.** Hermite finite elements basis functions: (a) cubic; (b) quintic.

$$\psi(t) = \begin{cases} t\,(1-|t|)^3\,(1+3|t|) & \forall\,|t| \le 1 \\ 0 & \forall\,|t| > 1; \end{cases}$$

$$\chi(t) = \begin{cases} \frac{1}{2}t^2\,(1-|t|)^3 & \forall\,|t| \le 1 \\ 0 & \forall\,|t| > 1. \end{cases}$$

Considering the prescribed time $T$ in the functional integral (6), let us construct a uniform time mesh by dividing the domain $[0, T]$ into $N$ equal pieces such that the size of each time finite element is of $h = T/N$, by the instances $t_p = p\,h$. We define the basis functions for $0 \le p \le N$ as

$$\begin{aligned} \phi_p(t) &= \phi\left(\frac{t}{h} - p\right) \\ \psi_p(t) &= \psi\left(\frac{t}{h} - p\right) \\ \chi_p(t) &= \chi\left(\frac{t}{h} - p\right), \end{aligned} \tag{11}$$

where $\chi_p(t)$ is used only for the QHFE case. The number of basis functions corresponds to $2N+2$ for the CHFE case, and to $3N+3$ for the QHFE case. Basis functions that overflow the domain $[0, T]$ must be truncated on the left for $p = 0$ or on the right for $p = N$. Each configuration parameter $q^i$ and each control parameter $u_i$ can then be approximated by the piecewise CHFE

$$\begin{aligned} q_h^i &= \sum_{p=0}^{N} \left(a_p^i \phi_p(t) + h\,b_p^i \psi_p(t)\right) \\ u_{h_i} &= \sum_{p=0}^{N} \left(x_p^i \phi_p(t) + h\,y_p^i \psi_p(t)\right), \end{aligned} \tag{12}$$

or alternatively by the QHFE

$$\begin{aligned} q_h^i &= \sum_{p=0}^{N} \left(a_p^i \phi_p(t) + h\,b_p^i \psi_p(t) + h^2\,c_p^i \chi_p(t)\right) \\ u_{h_i} &= \sum_{p=0}^{N} \left(x_p^i \phi_p(t) + h\,y_p^i \psi_p(t) + h^2\,z_p^i \chi_p(t)\right). \end{aligned} \tag{13}$$

Let us remark for now that coefficients $a_p^i$, $b_p^i$ and $c_p^i$ respectively correspond to the values of parameters $q^i$, their time derivatives $\dot{q}^i$, and their second time derivatives $\ddot{q}^i$ at time instance $t = p\,h$. Analogously, coefficients $x_p^i$, $y_p^i$ and $z_p^i$ correspond to the values of parameters $u_i$, their time derivatives $\dot{u}_i$, and their second time derivatives $\ddot{u}_i$ at time instance $t = p\,h$.

### B. INTERPOLATIONS ACCURACY

Let us now consider a one dimensional example in the approximation of the circular configuration parameter $q(t) = \sin t$. After following the piecewise HFE interpolation procedure presented in III-A, the error $e(t) = q_h(t) - \sin t$ of the interpolations and their convergence rates are measured as suggested by [9]. Let us introduce the norms

$$\|e\|_3 = \sqrt{\int_0^T \left[(e(t))^2 + h^2(\dot{e}(t))^2\right] \mathrm{d}t}$$

for the CHFE case, and

$$\|e\|_5 = \sqrt{\int_0^T \left[(e(t))^2 + h^2(\dot{e}(t))^2 + h^4(\ddot{e}(t))^2\right] \mathrm{d}t}$$

for the QHFE case. Errors for varied values of $h$ are plotted in the logarithmic scale. This allows to visualize convergence rates which correspond to the slope of the error logarithm, as a function of the logarithm of $h$. As shown by Figure 2a, the norm of the error $e(t) = q_h(t) - \sin t$ reveals to be in $h^4$ for the CHFE case and in $h^6$ for the QHFE interpolation case. These convergence rates are one degree higher than the order of the chosen HFE interpolation and is a phenomenon known as superconvergence. This is due to the mesh uniformity and the choice of the error norm [9].

### C. TIME INTEGRATION PROCEDURE

When the torque history $u_i(t)$ is known, the system state vector $x^i(t) = \left(q^i(t), \dot{q}^i(t)\right)$ can be predicted at any time instance $t$ after its initial value $x^i(0) = \left(q^i(0), \dot{q}^i(0)\right)$ by

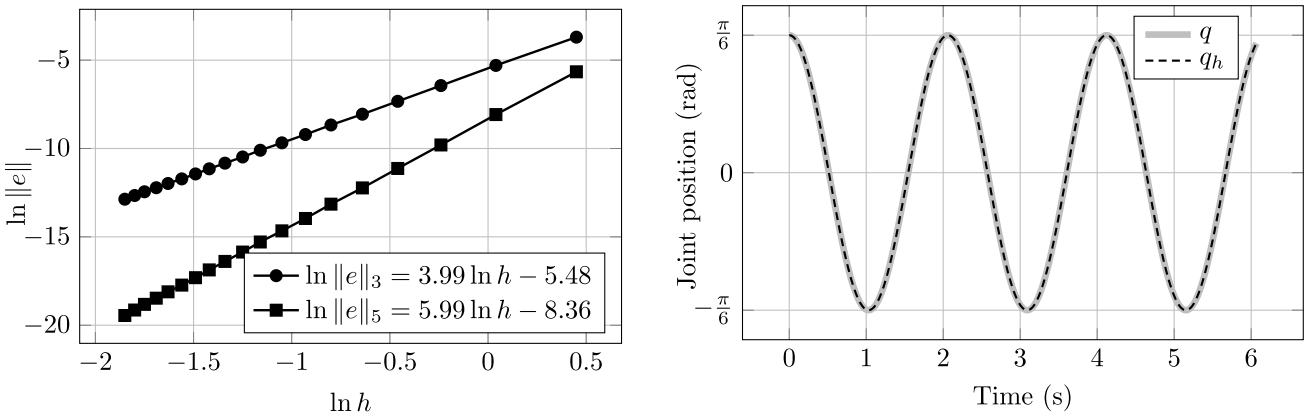



**FIGURE 2.** Accuracy of Hermite finite elements interpolations. (a) Interpolations superconvergence: cubic elements convergence is in $h^4$ and quintic elements convergence is in $h^6$, where $h$ is the step size. (b) Analytical solution $q$ versus CHFE approximation $q_h$ of the nonlinear pendulum motion: the CHFE approximation is as accurate as the analytical solution.

numerically solving the system motion equation (2). However, when an optimal trajectory is desired, the ODE system (10) may be solved instead between boundary values $x^i(0)$ and $x^i(T)$. Motion simulations may be conducted by following the procedure:

(i) approximate each configuration parameter $q^i$ by its interpolation $q_h^i$, and each control parameter $u_i$ by its interpolation $u_{h_i}$ (12) for the CHFE case or (13) for the QHFE case;

(ii) express motion and control equations (10) at $2n(N+1)$ time instances for the CHFE case or $3n(N+1)$ time instances for the QHFE case ($n$ is the number of system DOF and $N$ is the desired nodes quantity):

$$M_{ij}\ddot{q}_h^j + \Gamma_{ikl}\dot{q}_h^k\dot{q}_h^l + \nabla_i V - u_{h_i} = 0$$
$$\frac{\hat{\mathrm{d}}^2\, u_{h_i}}{\mathrm{d}t^2} + R^k_{jil}\dot{q}_h^j\dot{q}_h^l u_{h_k} + \nabla_i\nabla_l V\, u_h^l = 0;$$

after some development, the previous system becomes an algebraic system with unknowns $a_p^i$, $b_p^i$, $x_p^i$, $y_p^i$ (for CHFE discretization), $c_p^i$ and $z_p^i$ (for QHFE discretization)

$$\mathbf{K}\mathbf{v} = \mathbf{r}$$

where $\mathbf{K}$ is a square band matrix containing the basis functions (11) evaluated at each node and scaled by the system parameters, $\mathbf{v}$ is the vector of unknowns, and $\mathbf{r}$ contains the residual terms given by the boundary conditions;

(iii) solve the algebraic system $\mathbf{K}\mathbf{v} = \mathbf{r}$ using generalized matrix inversion techniques;

(iv) build the approximations $q_h^i$ and $u_{h_i}$ ((12) for CHFE discretization or (13) for QHFE discretization).

In the above procedure, steps (ii) and (iii) are inspired from the inertial parameters identification method presented in [34]. Steps (ii) and (iii) show that our FE-based procedure determines the whole trajectory at once by solving $\mathbf{K}\mathbf{v} = \mathbf{r}$. Steps (i) and (ii) of the above procedure constitute the HFE time discretization Algorithm 1 shown below. Note that the above procedure works, as presented, with linear models. In order to account for nonlinearities, a perturbation method will be detailed in the next subsection III-D.

Recall that $\mathbf{v}$ contains the values of parameters $(q^i, \dot{q}^i, \ddot{q}^i, u_i, \dot{u}_i, \ddot{u}_i)$ at time instances $t = p\,h$ (last paragraph of III-A). Because of this important property, reconstructions (12) or (13) may not be needed if only the values of these parameters, at nodes, suffice. As a consequence, other interpolations may be carried out in step (iv) if desired.

In the examples of IV-B and IV-C, boundary conditions provide with the values of $a_0^i = q^i(0)$, $b_0^i = \dot{q}^i(0)$, $a_N^i = q^i(T)$, and $b_N^i = \dot{q}^i(T)$.

### D. ONE DIMENSIONAL NONLINEAR EXAMPLE

In order to verify the accuracy of our simulation method, we now study the periodicity and the precision over the period of the classical one-dimensional nonlinear pendulum. However, this system motion equation is nonlinear, which is also generally true for robotic systems. Therefore, a perturbation method [35] is conducted in order to implement the above time integration algorithm. This is an iterative process where increments $\Delta q$ and $\Delta u$ are respectively added to approximate solutions $q$ and $u$ at each step. Upon perturbation, motion equations (2) develop in:

$$\begin{aligned} M_{ij}\left(\ddot{q}^j + \Delta\ddot{q}^j\right) &+ \frac{\partial M_{ij}}{\partial q^k}\Delta q^k\ddot{q}^j \\ &+ \Gamma_{ikl}\left(\dot{q}^k\dot{q}^l + \dot{q}^l\Delta\dot{q}^k + \dot{q}^k\Delta\dot{q}^l\right) \\ &+ \frac{\partial\Gamma_{ikl}}{\partial q^m}\dot{q}^k\dot{q}^l\Delta q^m \\ &+ \nabla_i V + \frac{\partial^2 V}{\partial q^j\partial q^i}\Delta q^j = u_i + \Delta u_i, \end{aligned} \tag{14}$$

where squared "$\Delta$" terms vanish.

**Algorithm 1** HFE Time Discretization Algorithm

**procedure** HFE($q$, $u$)
  **for** $i \leftarrow 1, n$ **do** ▷ $n$ is the system number of DOF
    **if** CHFE **then**
      **for** $m \leftarrow 0, 2N$ **do** ▷ $N$ is the desired nodes quantity
        $q^i(t_m) \leftarrow q_h^i(t_m) = \sum_{p=0}^{N} \left( a_p^i \phi_p(t_m) + h\, b_p^i \psi_p(t_m) \right)$
        $u_i(t_m) \leftarrow u_{h_i}(t_m) = \sum_{p=0}^{N} \left( x_p^j \phi_p(t_m) + h\, y_p^j \psi_p(t_m) \right)$ ▷ Using equation (12)
        Insert $\left(q_h^i(t_m), u_{h_i}(t_m)\right)$ into the target ODE
      **end for**
    **else if** QHFE **then**
      **for** $m \leftarrow 0, 3N$ **do** ▷ $N$ is the desired nodes quantity
        $q^i(t_m) \leftarrow q_h^i(t_m) = \sum_{p=0}^{N} \left( a_p^i \phi_p(t_m) + h\, b_p^i \psi_p(t_m) + h^2\, c_p^i \chi_p(t_m) \right)$
        $u_i(t_m) \leftarrow u_{h_j}(t_m) = \sum_{p=0}^{N} \left( x_p^i \phi_p(t_m) + h\, y_p^i \psi_p(t_m) + h^2\, z_p^i \chi_p(t_m) \right)$ ▷ Using equation (13)
        Insert $\left(q_h^i(t_m), u_{h_i}(t_m)\right)$ into the target ODE
      **end for**
    **end if**
  **end for**
  Establish the resulting algebraic system $\mathbf{Kv} = \mathbf{r}$ subject to the required boundary values
**end procedure**

For the present case of a simple pendulum, the motion equation is $\ddot{q} + \omega^2 \sin q = 0$ where only initial values for $q(0) = \frac{\pi}{6}$ rad and $\dot{q}(0) = 0\ \mathrm{rad\,s^{-1}}$ are given as boundary conditions. Note that no torques $u_i$ appear in this equation and thus, no $\Delta u$ increment is introduced because it vanishes. Note also that the initial position implies a $\frac{\pi}{3}$ rad amplitude motion (see Figure 2b). For the present one-dimensional case, the mass tensor has a unique constant component $M$, embedded in the $\omega$ parameter. Pendulum length and mass are such that $\omega = 3.102\,\mathrm{s^{-1}}$.

For this example, CHFE discretization is used (see III-A) in the time integration procedure proposed in III-C with $N = 100$ interpolation nodes. The implemented perturbation method then leads to the following procedure:

(1) approximate the linear pendulum motion by solving the equation $\ddot{q} + \omega^2 q = 0$ with the same initial conditions as the nonlinear pendulum, and let that solution be called $q_1$;
(2) according to the perturbed motion equation (14) approximate the solution $\Delta q_1$ of the linear ODE $\Delta\ddot{q}_1 + \omega^2 \cos(q_1)\Delta q_1 = -\ddot{q}_1 - \omega^2 \sin(q_1)$ with vanishing initial conditions;
(3) improve the approximation $q_1$ by setting $q_2 = q_1 + \Delta q_1$;
(4) iterate this process until $\|\Delta q_i\|_3 < 10^{-5}$ rad.

A periodic solution $q_h$ was found after three iterations and is shown by Figure 2b. This numerical solution can be compared to the analytical solution $q$ expressed with Jacobi's elliptic function *sn* [36], [37], also shown by Figure 2b. With our time methodology, the identified oscillations period was of 2.061 08 s, and is $-3 \times 10^{-7}$ s away from the period obtained with the incomplete elliptic integral of the first kind $F$ [38],

$$\tau = \frac{4}{\omega} \int_0^{\frac{\pi}{2}} \frac{\mathrm{d}\alpha}{\sqrt{1 - \left(\sin^2 \frac{\pi}{12}\right) \sin^2 \alpha}} = \frac{4}{\omega} F\left(\frac{\pi}{2}, \sin^2 \frac{\pi}{12}\right).$$

Naturally, the above reported error can be decreased by further increasing the number of interpolation nodes or by using QHFE instead. The norm $\| \ \|_3$ of the difference between the theoretical solution and our numerical solution reveals to be in $h^4$, again illustrating the superconvergence of the method (the resulting graph of $\ln \|e\|_3$ versus $\ln \|h\|$ is similar to the one shown by Figure 2a). This superconvergence is a consequence of Hermite's technique and will also occur for more complex single ODE models [9]. This example validates our approach for solving second order nonlinear ODE.

## IV. OPTIMAL CONTROL OF ROBOTIC MANIPULATORS

### A. OPTIMIZATION PROCEDURE

Let us consider the nonlinear ODE system (10). Its solution provides optimal torques and trajectories. Recall that $4n$ boundary conditions $q^i(0)$, $q^i(T)$, $\dot{q}^i(0)$ and $\dot{q}^i(T)$ are usually given, thus establishing a TPBVP. Instead of solving the $2n$ second order ODE (10), we can insert the CHFE or QHFE approximations (respectively given by equations (12) or (13)) of $q$ and $u$ in (10), and conduct the time integration procedure

proposed in III-C with the perturbation method introduced in III-D. Recall that the motion equation (10a) is perturbed as in equation (14). Analogously, control equation (10b) is perturbed according to equation (15):

$$\begin{aligned}
&\frac{\hat{\mathrm{d}}^2 u_i}{\mathrm{d}t^2} + \frac{\hat{\mathrm{d}}^2 \Delta u_i}{\mathrm{d}t^2} \\
&\quad - \left( \frac{\partial \Gamma^k_{ij}}{\partial q^l} - \Gamma^m_{ji}\Gamma^k_{ml} \right) \left( \Delta\dot{q}^j \dot{q}^l u_k + \dot{q}^j \Delta\dot{q}^l u_k \right) \\
&\quad - \left( \frac{\partial^2 \Gamma^k_{ij}}{\partial q^p \partial q^l} - \frac{\partial \Gamma^m_{ji}}{\partial q^p}\Gamma^k_{ml} - \Gamma^m_{ji}\frac{\partial \Gamma^k_{ml}}{\partial q^p} \right) \dot{q}^j \dot{q}^l u_k \Delta q^p \\
&\quad - \Gamma^k_{ij} \left( \Delta\ddot{q}^j u_k + 2\Delta\dot{q}^j \dot{u}_k \right) - \frac{\partial \Gamma^k_{ij}}{\partial q^l} \Delta q^l \left( \ddot{q}^j u_k + 2\dot{q}^j \dot{u}_k \right) \\
&\quad + R^k_{jil} \left( \dot{q}^j \dot{q}^l u_k + \Delta\dot{q}^j \dot{q}^l u_k + \dot{q}^j \Delta\dot{q}^l u_k + \dot{q}^j \dot{q}^l \Delta u_k \right) \\
&\quad + \frac{\partial R^k_{jil}}{\partial q^p} \dot{q}^j \dot{q}^l u_k \Delta q^p + \nabla_i \nabla_l V (u^l + \Delta u^l) \\
&\quad + \left( \frac{\partial^3 V}{\partial q^p \partial q^l \partial q^i} - \frac{\partial \Gamma^m_{il}}{\partial q^p} \frac{\partial V}{\partial q^m} - \Gamma^m_{il} \frac{\partial^2 V}{\partial q^p \partial q^m} \right) u^l \Delta q^p \\
&\quad + \nabla_i \nabla_l V \frac{\partial M^{ls}}{\partial q^p} u_s \Delta q^p = 0, \qquad (15)
\end{aligned}$$

where squared "$\Delta$" terms vanish.

Therefore, our optimal control procedure consists in adapting the perturbation method proposed in section III-D, to the solution of the nonlinear ODE system (10):

(1) approximate the solution of the linearized system (10) with the same boundary values as for the nonlinear case, and let the solutions be called $(q^i_1, u^1_i)$;
(2) approximate the solution of the ODE system composed of the perturbed motion equations (14) and the perturbed control equations (15) with vanishing boundary values, and let the solutions be called $(\Delta q^i_1, \Delta u^1_i)$;
(3) improve the approximations $(q^i_1, u^1_i)$ by setting $q^i_2 = q^i_1 + \Delta q^i_1$ and $u^2_i = u^1_i + \Delta u^1_i$;
(4) iterate this process until $\|\Delta q_i\|_3 < \epsilon$ (for CHFE discretization) or $\|\Delta q_i\|_5 < \epsilon$ (for QHFE discretization), where $\epsilon$ is chosen accordingly to the desired accuracy ($10^{-3}$ is a reasonable value).

We will refer to this general procedure as HFE/Perturbation, which denotes two variants: CHFE/Perturbation for CHFE time discretization, or QHFE/Perturbation for QHFE time discretization. Note that in step (3), covariant torque components $u_i$ are being calculated instead of the contravariant components $u^i$. Let us emphasize that when conducting motion control simulations and experiments with HFE discretization, the ODE system (10) now depends solely on coefficients $a^i_p$, $b^i_p$, $x^i_p$ and $y^i_p$ for the CHFE case, or $a^i_p$, $b^i_p$, $c^i_p$, $x^i_p$, $y^i_p$ and $z^i_p$ for the QHFE case. Either way, recall that $a^i_p$ and $b^i_p$ are known for $p = 0$ and $p = N$:

$$a^i_0 = q^i(0),\ b^i_0 = \dot{q}^i(0),\ a^i_N = q^i(T),\ \text{and } b^i_N = \dot{q}^i(T).$$

Upon obtaining an approximation of $q^i(t)$, recall that approximations of $\dot{q}^i(t)$ and $\ddot{q}^i(t)$ are also automatically being obtained (see last comment in III-A). The same applies for optimal torques $u_i(t)$ and their first and second time derivatives. Algorithm 2 presents our HFE/Perturbation method that combines the HFE time discretization procedure (Algorithm 1) with the perturbation method presented above.

**Algorithm 2** Time Integration Algorithm Using HFE Time Discretization and Perturbation Method

**procedure** HFE/Perturbation($q$, $u$)
  **Execute** procedure HFE($q$, $u$) to discretize the linearized system (10), with the same boundary values as for the nonlinear system
  **Solve** the resulting algebraic system $\mathbf{Kv} = \mathbf{r}$ to obtain solutions $(q^i_1, u^1_i)$
  **for** $s \leftarrow 1, w$ **do** ▷ $w$ is sufficiently large
    **Insert** $(q^i_s, u^s_i)$ into the perturbed motion equations (14) and perturbed control equations (15)
    **Execute** procedure HFE($\Delta q^i_s$, $\Delta u^s_i$) to discretize the perturbed motion equations (14) and perturbed control equations (15) with null boundary values
    **Solve** the resulting algebraic system $\mathbf{Kv} = \mathbf{r}$ to obtain solutions $(\Delta q^i_s, \Delta u^s_i)$
    $q^i_{s+1} \leftarrow q^i_s + \Delta q^i_s$
    $u^{s+1}_i \leftarrow u^s_i + \Delta u^s_i$
    **if** $\|\Delta q_s\| \leq \epsilon$ **then**
      **Break**
    **end if**
  **end for** ▷ Where $\epsilon$ is chosen accordingly to the desired accuracy
**end procedure**

### B. OPTIMAL CONTROL OF A 1-DOF ROBOTIC MANIPULATOR

In order to illustrate and validate our optimization procedure, we shall consider a simple robotic manipulator with one DOF. To resolve optimal trajectories and torques, the solution of the governing motion and control equations must be obtained:

$$\begin{cases} M(\ddot{q} + \omega^2 \sin(q)) = u \\ \ddot{u} + \omega^2 u \cos(q) = 0, \end{cases} \qquad (16)$$

where $M = 1.384\,\mathrm{kg\,m^2}$ is the unique component of the mass tensor; system link length and mass are such that $\omega = 2.609\,\mathrm{s^{-1}}$. The prescribed trajectory duration is set to $T = 1.0\,\mathrm{s}$. Boundary values are given by

$$q(0) = 0,\ \dot{q}(0) = 0,\ q(T) = \frac{\pi}{3},\quad \dot{q}(T) = 0. \qquad (17)$$

According to the optimization procedure proposed in the previous section, the solution to the above second order nonlinear ODE system must first be approximated by solving the following linear ODE system (step (1),

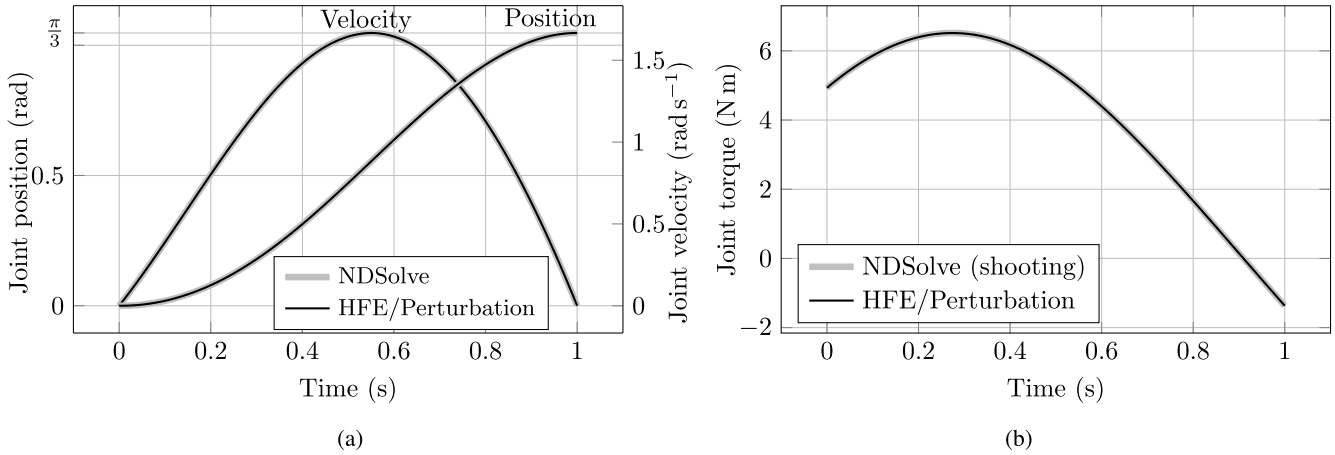



**FIGURE 3.** Optimal trajectory provided by Mathematica's NDSolve and the proposed HFE/Perturbation algorithm approximation, for the optimal motion simulation of a 1-DOF robotic manipulator: (a) position and velocity; (b) torque.

lines 2 and 3 of Algorithm 2)

$$\begin{cases} M(\ddot{q} + \omega^2 q) - u = 0 \\ \ddot{u} + \omega^2 u = 0, \end{cases}$$

subject to the above mentioned boundary conditions, where CHFE approximations of $q$ and $u$ are inserted. This provides with solutions $(q_1, u^1)$ that are used as a starting point to further solve the perturbation of (16). This perturbation is composed of equations (14) and (15) (step (2)), which for instance reduce to

$$\begin{cases} M(\Delta\ddot{q}_1 + \omega^2 \cos(q_1)\Delta q_1) - \Delta u^1 \\ + M(\ddot{q}_1 + \omega^2 \sin(q_1)) - u^1 = 0 \\ \Delta\ddot{u}^1 + \omega^2 \cos(q_1)\Delta u^1 - \omega^2 u^1 \sin(q_1)\Delta q_1 \\ + \ddot{u}^1 + \omega^2 \cos(q_1) u^1 = 0. \end{cases}$$

Following steps (3) and (4) of the optimization procedure proposed in IV-A (lines 5 to 9 of Algorithm 2) leads to approximate the solution of the initial nonlinear problem (16). Note that Algorithm 1 implies an approximation of the above systems of ODE with rectangular matrix equations. The latter is solved by using generalized matrix inversion. Note also that this procedure minimizes the performance index $J(u_h)$ (see equation (6)).

Algorithms 1 and 2 were implemented in Wolfram Mathematica® using $N = 50$ nodes for both the CHFE/Perturbation and QHFE/Perturbation time integration algorithms, and led to an acceptable solution after three iterations of Algorithm 2. For reference, the nonlinear ODE system (16) was also solved using the built-in NDSolve ODE solver, set up to operate with shooting methods. Solutions obtained with our algorithm and solutions obtained with NDSolve are displayed in Figures 3a and 3b.

The optimal trajectory positions and velocities provided by our HFE/Perturbation solvers and NDSolve are indistinguishable on Figures 3a and 3b, thus indicating that our method provides reasonable results. Note that both solutions successfully satisfy position and velocity boundary values that lead to the desired motion (Figure 3a).

**TABLE 1.** Solver performance for the solution of equations (16) for the optimal motion simulation of a 1-DOF robotic manipulator subject to boundary conditions (17) where $T = 1$ s. Lower values of performance index (PI, equation (6)) and error norms indicate better performance.

| Solver | PI | $\|f^m\|$ | $\|f^c\|$ |
|---|---|---|---|
| NDSolve (Shooting) | 8.339 | 0.090 | 0.612 |
| NDSolve (RK) | 8.339 | $2.5 \times 10^{-5}$ | $2.5 \times 10^{-4}$ |
| CHFE/Perturbation | 8.339 | $7.4 \times 10^{-5}$ | $5.5 \times 10^{-4}$ |
| QHFE/Perturbation | 8.339 | $1.5 \times 10^{-6}$ | $2.0 \times 10^{-5}$ |

Recalling that the initial goal is to minimize the performance index (6), by solving the set of equations (10) for robotic manipulators, the performance of our algorithm can be assessed by evaluating the performance index value for the calculated trajectory. Additionally, let us define $\|f^m\|$ as the error norm of the left hand side of equation (10a):

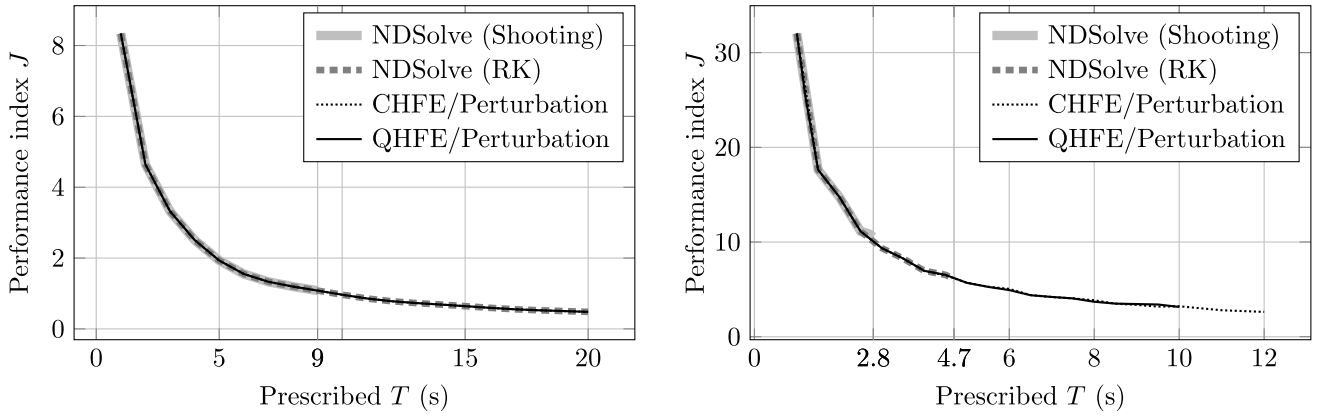
$$\|f^m\| = \sqrt{\int_0^T \left(M_{ij}\ddot{q}^j + \Gamma_{ikl}\dot{q}^k\dot{q}^l + \nabla_i V - u_i\right)^2 \mathrm{d}t}. \quad (18)$$

Analogously, let us define $\|f^c\|$ as the error norm of the left hand side of equation (10b):

$$\|f^c\| = \sqrt{\int_0^T \left(\frac{\hat{\mathrm{d}}^2 u_i}{\mathrm{d}t^2} + R^k_{jil}\dot{q}^j\dot{q}^l u_k + \nabla_i\nabla_l V\, u^l\right)^2 \mathrm{d}t}. \quad (19)$$

These error norms evaluate the accuracy of our proposed method to solve the ODE system (10). In the following, our algorithm will be compared with Mathematica's NDSolve solver on these terms.

As shown by Table 1, when NDSolve is set up to operate with shooting methods, error norms $\|f^m\|$ and $\|f^c\|$ evaluate to fairly large values as compared to the other methods. Therefore, it was alternatively set up to use a Runge-Kutta (RK) method for time integration in order to increase its accuracy. Note that both our HFE/Perturbation algorithms lead to the same performance index value as both NDSolve solutions,




(a) 1-DOF manipulator

(b) 2-DOF manipulator

**FIGURE 4. Performance index evolution for growing values of trajectory duration $T$ for the optimal motion simulation robotic manipulators: (a) 1-DOF subject to boundary conditions (17); (b) 2-DOF subject to boundary conditions (20). Comparison of four solvers. NDSolve (Shooting) is unable to produce results for $T > 9$ s for the 1-DOF manipulator, and for $T > 4.7$ s for the 2-DOF manipulator. HFE/Perturbation is able to produce results for longer trajectory durations than NDSolve.**

thus exhibiting adequate performance. QHFE/Perturbation does however display lower error norm values, further verifying the accuracy of our method. This occurs because QHFE/Perturbation directly computes position and torque, along with their first and second derivatives at each node, which are then used to calculate norms $\|f^m\|$ and $\|f^c\|$. NDSolve provides with $(q, u)$ as an output, so these have to be differentiated afterwards in order to compute $\|f^m\|$ and $\|f^c\|$. It is important to remark that the precision of our methods can be augmented by increasing the total number of interpolation nodes (which was of $N = 50$ for this example).

Additionally, it is possible to verify the stability over the trajectory duration of each solver by conducting simulations with increasing values of $T$, where boundary conditions on position and velocity are kept equal. Naturally, solvers struggle more with longer simulation periods. We have therefore conducted the following test: equations (16), subject to boundary conditions (17), were solved using the four solvers NDSolve (Shooting), NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation, for increasing values of $T$ going from 1 s to 20 s. It resulted that three solvers were equally stable: NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation all were able to give a solution up to $T = 20$ s. However, NDSolve (Shooting) was only able to provide an adequate solution up to $T = 9$ s. For every value of $T > 9$ s, NDSolve (Shooting) was unable to provide a solution to (16) that meets boundary conditions (17). This is shown by Figure 4a where the curve for NDSolve (Shooting) truncates at $T = 9$ s. The other three solvers returned practically the same performance index values for this particular example for $T > 9$ s.

Interestingly, we could also verify that as $T$ grows, the performance index tends to zero, which is in accordance with our optimization goal, i.e., the minimization of (6) (see Figure 4a). For example, at $T = 20$ s, $J = 0.481$ and the RMS torque for the trajectory evaluates to 0.258 N m. Upon extending duration up to $T = 60$ s, the performance index goes down to $J = 0.158$ and the RMS torque for the trajectory drops to 0.085 N m, and at $T = 100$ s, we obtain $J = 0.095$ and an RMS torque of 0.051 N m thus confirming that both RMS torque and performance index follow a similarly decreasing pattern.

### C. OPTIMAL CONTROL OF A 2-DOF ROBOTIC MANIPULATOR

The previous results indicate that our HFE/Perturbation algorithms can be successfully applied for the optimal path planning of robotic manipulators, and also illustrate how our method is applied. However, the previous 1-DOF example has a single constant mass tensor component, and such a robotic manipulator does not exhibit centrifugal and Coriolis effects embedded in the second term of motion equation (2).

We therefore now apply our method to the optimal path planning of a 2-DOF robotic manipulator with revolute joints which exhibits most complexities and nonlinearities that are characteristic of realistic robotic systems such as: configuration-dependent mass tensor components, centrifugal and Coriolis effects, and inertia effects mutually induced by each link. We conducted our simulations on a 2-DOF robotic manipulator platform for which the corresponding parameters are reported in [39] (see Figure 7). For this robot, the system of $2n$ second order nonlinear ODE (10) translates into a TPBVP involving a set of 4 ODE subject to 8 boundary values.

In this case, our simulation exercise was similar to the one we held in the previous section. It consisted in taking the robotic manipulator from an initial state, to a final state with increasing prescribed time $T$ values. Boundary conditions were set to

$$\begin{cases} \left(q^1(0), q^2(0)\right) = (0, 0) \text{ rad} \\ \left(\dot{q}^1(0), \dot{q}^2(0)\right) = (0, 0) \text{ rad s}^{-1} \\ \left(q^1(T), q^2(T)\right) = (0.8, 1.0) \text{ rad} \\ \left(\dot{q}^1(T), \dot{q}^2(T)\right) = (0, 0) \text{ rad s}^{-1}. \end{cases} \tag{20}$$

These simulations were conducted with Mathematica's NDSolve ODE solver set up to operate with shooting methods first, and then set up to operate with a RK time integration algorithm to obtain alternative results. Simulations were then conducted with both our HFE/Perturbation methods with a fixed number of nodes set to $N = 60$ for each case. Recall that QHFE provide with positions, velocities and accelerations at each node and show higher precision than CHFE (see section III-B and Table 1). These features might be desirable depending on the experimental platform requirements and thus QHFE might be favored over CHFE.

Figure 4b shows the performance index evolution for increasing values of $T$ for the optimal motion simulation of the 2-DOF manipulator subject to boundary values (20). It compares the performance index values obtained with four solvers and is the analog of Figure 4a but for the 2-DOF robot case. As opposed to the 1-DOF case, we now see that neither NDSolve (Shooting) nor NDSolve (RK) are able to complete the required trajectory past $T = 4.7$ s in Figure 4b. This situation occurs because the system of equations (10) is more complicated to solve for a 2-DOF robot than it is for a 1-DOF robot. Therefore, this example shows that our proposed HFE/Perturbation algorithms are more stable with respect to an increasing trajectory duration.

In fact, for $T > 2.8$ s, no performance index values are obtained with NDSolve (Shooting). This is because with this method, position or velocity boundary values could not be met (depending on the case) and therefore the desired trajectory could not be completed. Using the RK option for this solver, ensures that boundary values are met for higher values of $T$. However, for prescribed times greater than 4.7 s, this is no longer the case because the solver indicates that the system is suspected to be stiff. It has been noticed that optimal control methodologies often lead to complex numerical problems [40]. Stiffness is a common numerical issue that arises in the control of nonlinear systems [24]. We therefore set up NDSolve to account for stiffness but the outcome did not change.

As opposed to these methods, our HFE/Perturbation successfully achieves a trajectory that meets boundary conditions (20) even for specified times $T \geq 5.0$ s with acceptable accuracy. Interestingly, performance index values tend to be slightly lower with CHFE/Perturbation. However, as $T$ grows, its accuracy diminishes indicating that more nodes may be required. Additionally, CHFE/Perturbation usually requires more iterations of Algorithm 2 than QHFE/Perturbation, which is in accordance with Figure 2a. Nevertheless, the precision of our HFE/Perturbation methods increases by raising the number of nodes as demonstrated in section III-B (see Figure 2a).

Figure 4b also shows that no performance index values are obtained with the QHFE/Perturbation for $T > 10$ s. Interestingly this method struggles for trajectory durations that approach $T = 10$ s for the 2-DOF robot manipulator. Afterwards, the solutions obtained with this solver do meet boundary values, because of the nature of the FEM, but $\|\Delta q_s\|$ increases instead of decreasing to a value lower than $\epsilon$ in Algorithm 2. On the contrary, CHFE/Perturbation is able to provide solutions up to $T = 12$ s implying that the CHFE/Perturbation solver is more stable with respect to trajectory duration.

Interestingly, since our HFE/Perturbation algorithms allow to calculate optimal trajectories with durations of $T \geq 5$ s, it is now possible to analyze the behavior of our system as $T$ grows. Figures 5a and 5c show the optimal positions $(q_h^1, q_h^2)$ for three different values of $T$. Note how for $T = 1$ s, the joints reach the goal with one swing, whereas it requires more swings as $T$ grows. This behavior was observed for both robot joints in our simulations. A tendency for the optimal position can thus be appreciated: as $T$ grows, the robot links will oscillate ever approaching the desired goal value.

Now, consider Figures 5b and 5d which shows the optimal torques $(u_{1h}, u_{2h})$ for three different values of $T$. Note how as $T$ grows, torque values tend to oscillate around zero. Again, this behavior was observed for both joints in our simulations. This occurs because our optimization method focuses in minimizing a performance index (6) for which the integrand is a function of the joint torques. It is therefore natural that torques oscillate around the zero value.

### D. HFE/PERTURBATION METHOD CONVERGENCE ANALYSIS

The results presented in the previous section indicate that our HFE/Perturbation algorithms are more stable with respect to the trajectory duration $T$ than the NDSolve solver. However, the convergence of both our proposed HFE/Perturbation solvers shall be analyzed for our practical case. Indeed, a superconvergence of the HFE interpolations was demonstrated in section III-B. However, solving equations (10) is more complicated than a simple interpolation.

In order to determine the convergence of our algorithms to solve equations (10), we follow the methodology proposed in [9], which for our case articulates in the following steps.

1) Establish the *most converged* solutions to equations (10) subject to fixed boundary values following Algorithm 2 with a large number of nodes $N$. Solutions are denoted $(q_*, u_*)$ and serve as a reference against which other "less converged" solutions are compared with.
2) Obtain other "less converged" solutions to equations (10) subject to the same boundary values as for step 1) with a growing number of nodes $N$. In this step, the maximum number of nodes $N$ should be chosen such that both $\|q_* - q_h\| \leq \zeta$ and $\|u_* - u_h\| \leq \zeta$. In our case, we have settled with $\zeta = 10^{-5}$.
3) For each case of $N$, compute the logarithm of $\|q_* - q_h\|$, $\|u_* - u_h\|$, $\|f^m\|$ and $\|f^c\|$. Establish curves of these values against the corresponding value of $\log(h)$, where $h$ is the step size.
4) Fit the resulting curves with a first degree polynomial to obtain the convergence rate.

These steps were followed for the optimal motion simulation of both a 1-DOF robotic manipulator and a 2-DOF

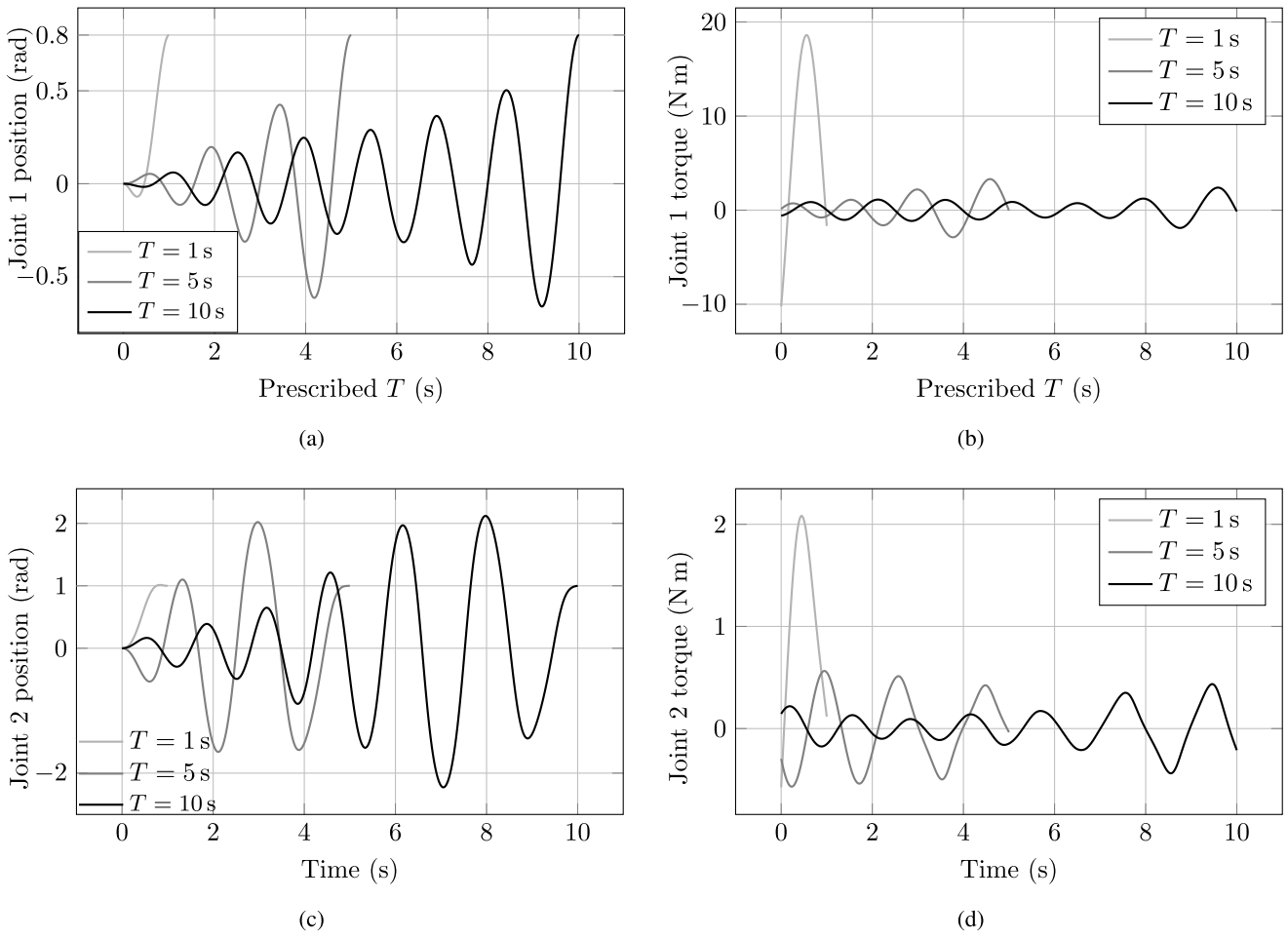


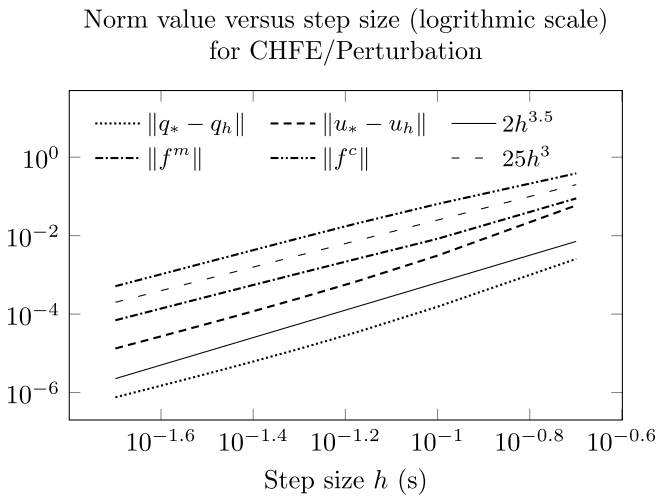

**FIGURE 5.** Optimal motion simulations of a 2-DOF robotic manipulator with QHFE/Perturbation for three different values of $T$. Figures show the tendency as $T$ increases for: (a) optimal configurations $q_h^1$; (b) optimal controls $u_{1h}$; (c) optimal configurations $q_h^2$; (d) optimal controls $u_{2h}$.

robotic manipulator. For both robots, we performed a trajectory where $T = 1\,\mathrm{s}$. Boundary values (17) were taken for the case of the 1-DOF robotic manipulator, and (20) for the case of the 2-DOF robotic manipulator. For the 1-DOF robot, we set $N = 1500$ to calculate $(q_*, u_*)$ (step 1) above) with both CHFE/Perturbation and QHFE/Perturbation. For the 2-DOF, we set $N = 1000$ to calculate $(q_*, u_*)$ with both HFE/Perturbation algorithms.

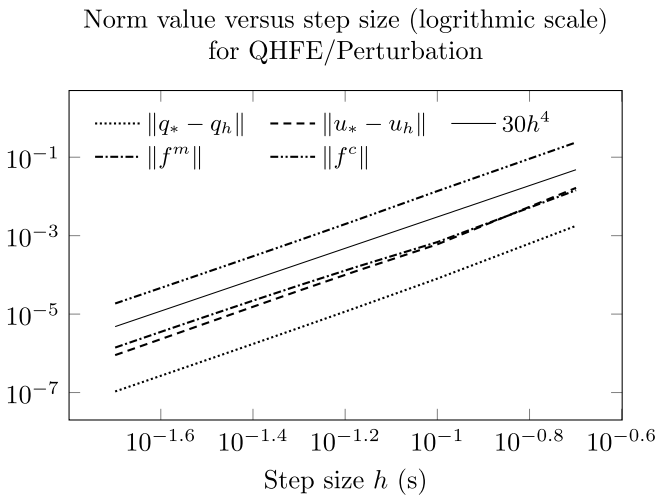
Starting with the 1-DOF robotic manipulator, Figure 6a shows the various norm values plotted against the step size in the logarithmic scale for CHFE/Perturbation. This figure shows a curve with equation $25h^3$ located between norms $\|f^m\|$ and $\|f^c\|$ for reference. Convergence of these norms resulted to be in $h^3$. The same figure also shows a curve with equation $2h^{3.5}$ located between norms $\|q_* - q_h\|$ and $\|u_* - u_h\|$ for reference. Convergence of these norms resulted to be in $h^{3.5}$. Analogously, Figure 6b displays the same norms plotted against the step size also in the logarithmic scale for QHFE/Perturbation. This figure shows a curve with equation $30h^4$ located between the norms $\|f^m\|$ and $\|f^c\|$ for reference. Convergence of all norms resulted to be in $h^4$ for QHFE/Perturbation.

In the case of the 2-DOF robotic manipulator, Figure 6c shows the various norm values plotted against the step size in the logarithmic scale for CHFE/Perturbation. This figure shows a curve with equation $200h^3$ located between norms $\|f^m\|$ and $\|f^c\|$ for reference. Convergence of these norms resulted to be in $h^3$. The same figure also shows a curve with equation $25h^{3.5}$ located between norms $\|q_* - q_h\|$ and $\|u_* - u_h\|$ for reference. Convergence of these norms resulted to be in $h^{3.5}$. Analogously, Figure 6d displays the same norms plotted against the step size also in the logarithmic scale for QHFE/Perturbation. This figure shows a curve with equation $30h^4$ located between norms $\|q_* - q_h\|$ and $\|u_* - u_h\|$ for reference. Convergence of all norms resulted to be in $h^4$ for QHFE/Perturbation.

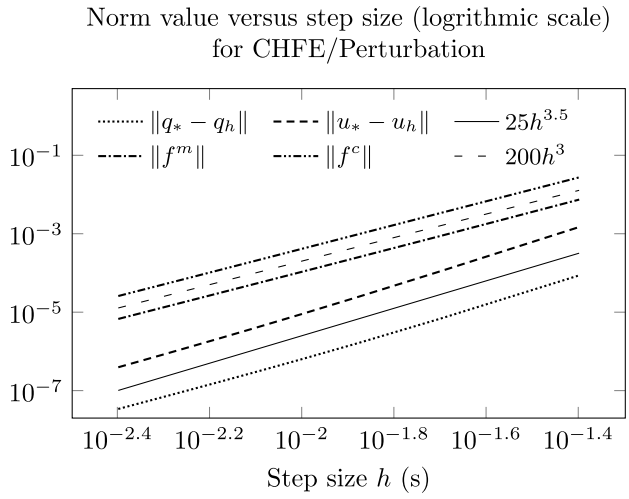
Let us remark that convergence rates were the same for both manipulators. This is significant because it indicates that our HFE/Perturbation method convergence rates are not system dependent. Instead, these rates depend on the target ODE



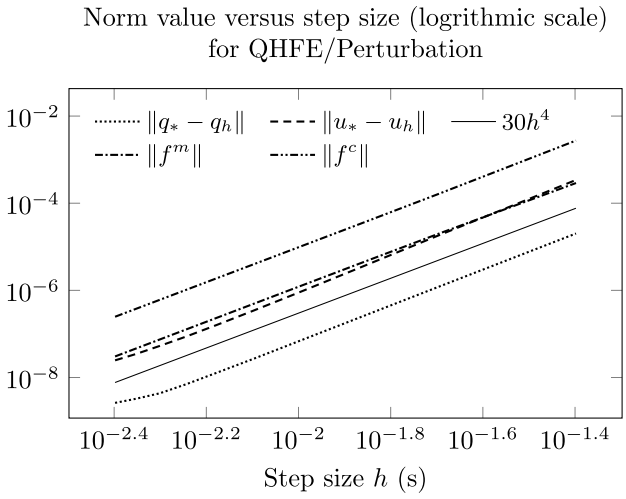

(a) 1-DOF




(b) 1-DOF




(c) 2-DOF




(d) 2-DOF

**FIGURE 6.** **Convergence of HFE/Perturbation for the optimal control of robotic manipulators: (a) CHFE/Perturbation for a 1-DOF robot; (b) QHFE/Perturbation for a 1-DOF robot; (c) CHFE/Perturbation for a 2-DOF robot; (d) QHFE/Perturbation for a 2-DOF robot. Convergence rates are equal for both manipulators.**

structure. As a summary, Table 2 presents the convergence rates for both CHFE/Perturbation and QHFE/Perturbation.

Table 2 reports two main facts:

1) that the superconvergence of HFE interpolations is not inherited to the case of solving equations (10);
2) that with QHFE/Perturbation, error norms did converge at a higher rate than with CHFE/Perturbation, but at a lower rate than expected.

Regarding the first issue, superconvergence was established for a simple interpolation case in section III-B. Admittedly, this motivated us to develop the presented HFE/Perturbation method. However, we must emphasize that the set of ODE (10) is not as simple. These target equations are second order nonlinear ODE. Nevertheless, we must also emphasize that for CHFE/Perturbation, the convergence rate of the evaluated norms is at least in the order of the chosen cubic interpolations. In fact, error norms $\|q_* - q_h\|$ and $\|u_* - u_h\|$ converge at a rate which is half a degree higher than the order of the cubic interpolations. Regarding the second issue, surprisingly, the evaluated error norms revealed to

**TABLE 2.** **Norm convergence rates with the proposed HFE/Perturbation method upon solving the second order nonlinear set of equations (10) subject to boundary values for both a 1-DOF and a 2-DOF robotic manipulators. Scalar *h* denotes the step size.**

| | Method convergence rate | |
|---|---|---|
| Norm | CHFE/Perturbation | QHFE/Perturbation |
| $\|q_* - q_h\|$ | $h^{3.5}$ | $h^4$ |
| $\|u_* - u_h\|$ | $h^{3.5}$ | $h^4$ |
| $\|f^m\|$ | $h^3$ | $h^4$ |
| $\|f^c\|$ | $h^3$ | $h^4$ |

converge at a rate which is one degree lower than the order of the chosen quintic interpolations with the QHFE/Perturbation method.

Recall that QHFE require one more basis function than CHFE (see section III-A). This means that to solve equations (10), QHFE/Perturbation computes $2nN$ more unknowns than CHFE/Perturbation according to Algorithm 1. This is beneficial when second time derivatives of $(q, u)$ are required at each node. However, CHFE/Perturbation reveals to be more cost effective (computationally speaking) whenever first time derivatives of $(q, u)$, at each node, suffice.

These results illustrate the performance and accuracy of our approach for the optimal path planning of robotic manipulators, and encourage us to perform motion control experiments to validate our proposal.

## V. EXPERIMENTAL VALIDATION

In order to validate our approach, we conducted controlled motion experiments on a 2-DOF robotic manipulator platform presented in [39], [41] and located at the Instituto Tecnológico de La Laguna, Mexico, previously used in [25]. The robotic manipulator is actuated by direct-drive brushless servo motors operated in torque mode. These act as torque sources, receiving analog voltages as torque reference signals. This property is ideal because it is thus possible to directly feed the optimal torques obtained with our HFE/Perturbation method as an input reference signal. Optimal positions and velocities are compared to the positions and velocities fed back from the robot. A 32-bit DSP microprocessor receives joint positions obtained via incremental encoders. The control algorithm is executed in a sampling period of 2.5 milliseconds on a host computer running the WinMechLab environment presented in [42].

### A. CONTROL SCHEME

The control law used to implement the optimal trajectories may be divided into two operation modes, an optimal reaching phase, which is active if $0 \leq t \leq T$, and a holding phase that is active if $t > T$, with $T$ given as in (20).

The control scheme used to implement the optimal reaching phase follows the structure of a Proportional-Derivative (PD) control plus a feed-forward term. Its purpose is to make the system follow a prescribed optimal trajectory which attains a specific value on a fixed prescribed time. The control law during this phase is described by the equation

$$\tau = \mathbf{u}_h + K_p\tilde{\mathbf{q}} + K_v\dot{\tilde{\mathbf{q}}},$$

where $\tau \in \mathbb{R}^n$ is comprised of the torques that are applied on each link of the $n$-DOF robot, and $\tilde{\mathbf{q}} = \mathbf{q}_h - \mathbf{q}$ represents the position error with $\mathbf{q}_h$ as the desired optimal position of the links. The matrices $K_p$ and $K_v$ are diagonal positive definite matrices of dimension $n \times n$. A schematic representation of this section of the control law may be seen in Figure 8.

The feed-forward term is denoted as $\mathbf{u}_h$ and its value is the optimal torque obtained from the optimization method

**FIGURE 7.** Robotic manipulator platform used for the experimental validation. It is a 2-DOF manipulator actuated by direct-drive brushless servo motors operated in torque mode.

previously described. In the absence of external disturbances and parametric uncertainties, and if the initial conditions at time $t = 0$ are precisely the ones that are used as boundary conditions, the output of the control law is precisely the feed-forward term, i.e., the optimal torque.

In practice, it may be that the initial conditions for the system deviate by a small amount due to measuring error. The PD part of the controller corrects this deviation. A deviation from initial conditions that is big in magnitude may be corrected by selecting appropriate values of the initial conditions (20). The PD term also adds robustness in the case where vanishing disturbances arise during operation or when manipulator parameters are not properly identified.

After the prescribed time $T$, the holding phase is active. During this phase, the control law becomes a PD plus gravity compensation controller. The equation that describes the control law can be written as

$$\tau = K_p\tilde{\mathbf{q}} - K_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}),$$

where $\mathbf{g}(\mathbf{q})$ is the gravitational torque vector of the robot, and $\tilde{\mathbf{q}} = \mathbf{q}_h - \mathbf{q}$, with $\mathbf{q}_h$ a constant vector that contains the values of the boundary conditions at $t = T$. On this phase the PD term serves the same purpose as in the previous phase and gravity compensation is used to keep the manipulator in place.

In summary, the overall control law may be written as

$$\tau = \begin{cases} \mathbf{u}_h + K_p\tilde{\mathbf{q}} + K_v\dot{\tilde{\mathbf{q}}} & \text{if } 0 \leq t \leq T \\ K_p\tilde{\mathbf{q}} - K_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) & \text{if } t > T \end{cases}$$

Friction compensation was also used during each phase to cancel the effect of kinetic friction. On the prototype, kinetic

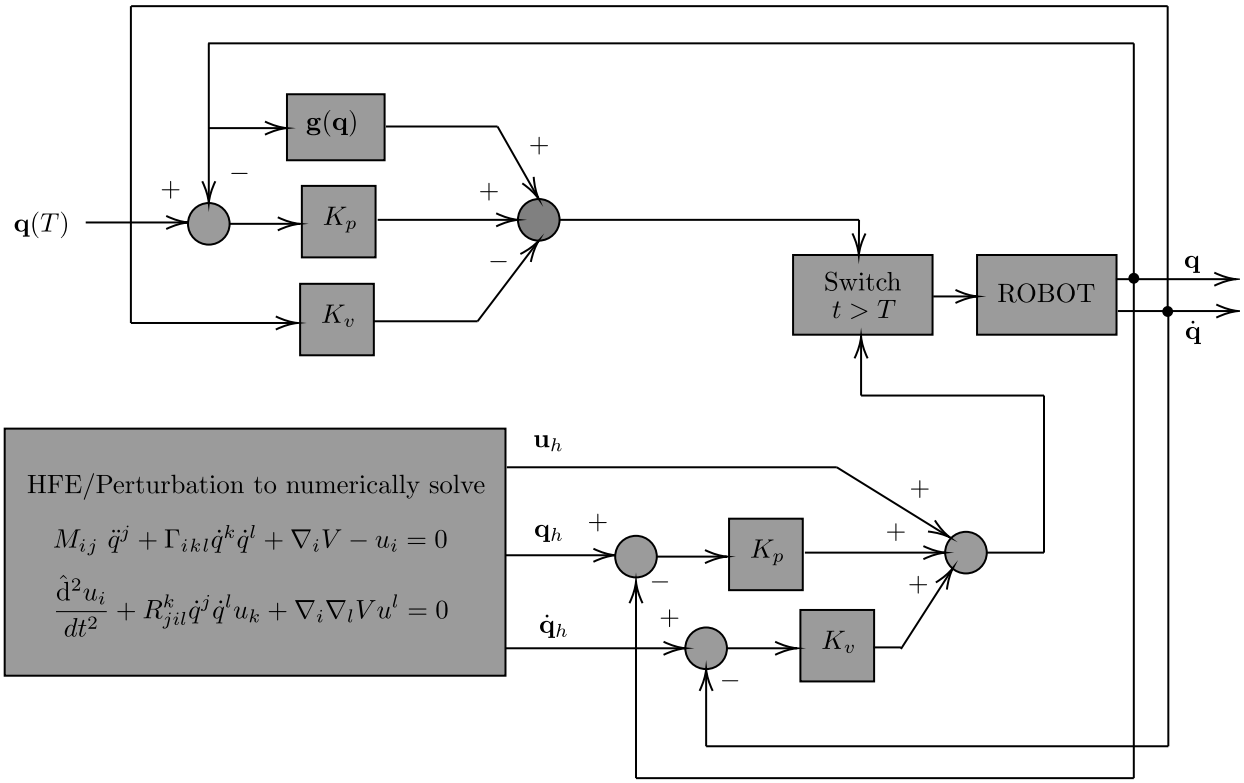



**FIGURE 8.** Schematic representation of the controller.

friction was modeled as Coulomb friction (as described in [43]) and the compensation that was used is as follows. Let $\tau_{exp}$ denote the torque that is actually applied to the manipulator during the experiment. The applied control torque, in the experiment, may be expressed as

$$\tau_{exp} = \tau + F_C \mathbf{sign}(\dot{\mathbf{q}})$$

where $\tau$ denotes the torque produced by either the optimal or reaching phase control laws, and $F_C$ is a diagonal matrix containing the coulomb friction coefficients. Since friction coefficients were previously identified, this strategy effectively cancels out the effects of the most predominant nonconservative forces affecting our prototype.

We used the same control scheme as in [25], where its stability was formally analyzed.

### B. THE EFFECT OF EXTERNAL DISTURBANCES

We now analyze the effect of external disturbances on the closed loop system. To quantify the effect of external disturbances, it is possible to consider each phase independently (optimal reaching and holding), as it is done to show the stability of the closed loop in [25]. The overall disturbance effect depends on the controller gains and the disturbance's upper bound.

As it was shown in [25], it is possible to express the solution $\mathbf{u}_h$ of equation (10) as a function of $\mathbf{q}_h$ in the following way:

$$\mathbf{u}_h(t) = M(\mathbf{q}_h)\ddot{\mathbf{q}}_h + C(\mathbf{q}_h, \dot{\mathbf{q}}_h)\dot{\mathbf{q}}_h + \mathbf{g}(\mathbf{q}_h)$$

where $M$ is the inertia matrix of the manipulator, $C$ is a matrix that captures the effect of centrifugal and Coriolis forces and $\mathbf{g}$ is a vector that contains the torques on the system that are due to the effects of gravity. Further details on this representation of the dynamical model may be found in [44].

Throughout this section, the external disturbance is assumed to be additive, that is, the motion equations of the mechanical system may be expressed as

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau(t) + \mathbf{d}(t)$$

where $\tau$ is the torque applied to each link (which is the sum of a PD term and $\mathbf{u}_h$), and $\mathbf{d}$ is the external disturbance on each link. It is also assumed that the norm of the external disturbance can be bounded by a positive constant $\delta_d$ such that $\|\mathbf{d}(t)\| \leq \delta_d$ for all $t \geq 0$.

With this in mind, it is possible to state that the closed loop system takes the following form during the optimal reaching phase:

$$\frac{d}{dt}\begin{bmatrix}\tilde{\mathbf{q}}\\ \dot{\tilde{\mathbf{q}}}\end{bmatrix} = \begin{bmatrix}\dot{\tilde{\mathbf{q}}}\\ M(\mathbf{q})^{-1}\left[-K_p\tilde{\mathbf{q}} - K_v\dot{\tilde{\mathbf{q}}} + \mathbf{z}_1(t)\right]\end{bmatrix}, \quad (21)$$

where

$$\mathbf{z}_1(t) = -C(\mathbf{q}, \dot{\mathbf{q}})\dot{\tilde{\mathbf{q}}} - \mathbf{h}(t, \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) - \mathbf{d}(t),$$

and $\mathbf{h}$ denotes the residual dynamics given by

$$\mathbf{h}(t, \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = [M(\mathbf{q}_h) - M(\mathbf{q}_h - \tilde{\mathbf{q}})]\ddot{\mathbf{q}}_h$$

$$+[C(\mathbf{q}_h,\dot{\mathbf{q}}_h)-C(\mathbf{q}_h-\tilde{\mathbf{q}},\dot{\mathbf{q}}_h-\dot{\tilde{\mathbf{q}}}_h)]\dot{\mathbf{q}}_h$$
$$+\mathbf{g}(\mathbf{q}_h)-\mathbf{g}(\mathbf{q}_h-\tilde{\mathbf{q}}).$$

Consider now the function

$$V(t,\tilde{\mathbf{q}},\dot{\tilde{\mathbf{q}}})=\frac{1}{2}\dot{\tilde{\mathbf{q}}}^T M(\mathbf{q})\dot{\tilde{\mathbf{q}}} + \frac{1}{2}\tilde{\mathbf{q}}^T K_p\tilde{\mathbf{q}}+\gamma\, tanh(\tilde{\mathbf{q}})^T M(\mathbf{q})\dot{\tilde{\mathbf{q}}}, \quad (22)$$

which satisfies the bounds $\alpha_1\|\mathbf{x}\|^2 \leq V(t,\tilde{\mathbf{q}},\dot{\tilde{\mathbf{q}}}) \leq \alpha_2\|\mathbf{x}\|^2$ with $x=[\tilde{\mathbf{q}}^T \;\; \dot{\tilde{\mathbf{q}}}^T]^T$ and constants $\alpha_1,\alpha_2>0$, whose existence is ensured by the tuning conditions presented in [25], given by

$$\alpha_1=\lambda_{\min}\{P_1\},$$
$$\alpha_2=\lambda_{\text{Max}}\{P_2\},$$

where $\lambda_{\min}\{P_1\}$ and $\lambda_{\text{Max}}\{P_2\}$ denote the smallest and largest eigenvalues, respectively, of matrices $P_1$ and $P_2$ that are defined as

$$P_1=\begin{bmatrix} K_{p_m} & -\gamma K_{M_M} \\ -\gamma K_{M_M} & K_{M_m}\end{bmatrix}$$

and

$$P_2=\begin{bmatrix} K_{p_M} & \gamma K_{M_M} \\ \gamma K_{M_M} & K_{M_M}\end{bmatrix}.$$

Constants $K_{p_m}, K_{v_m}, K_{M_m}$ denote lower bounds on the eigenvalues of matrices $K_p$, $K_v$ and $M(\mathbf{q})$, respectively. Constants $K_{p_M}, K_{v_M}, K_{M_M}$ denote upper bounds of the eigenvalues.

The time derivative of (22) along the trajectories of (21) satisfies

$$\dot{V}\leq -\gamma\begin{bmatrix}\|tanh(\tilde{\mathbf{q}})\| \\ \|\dot{\tilde{\mathbf{q}}}\|\end{bmatrix}^T Q_1\begin{bmatrix}\|tanh(\tilde{\mathbf{q}})\| \\ \|\dot{\tilde{\mathbf{q}}}\|\end{bmatrix} + \delta_d\|\dot{\tilde{\mathbf{q}}}+\gamma\, tanh(\tilde{\mathbf{q}})\| \quad (23)$$

with

$$Q_1=\begin{bmatrix} K_{p_m}-k_{h2} & -a-\frac{1}{\gamma}\frac{k_{h2}}{2} \\ -a-\frac{1}{\gamma}\frac{k_{h2}}{2} & \frac{1}{\gamma}\left[K_{v_m}-k_{h1}\right]-b\end{bmatrix}.$$

Now, consider a domain $\mathcal{D}_r=\{\mathbf{x}\in\mathbb{R}^{2n}:\|\mathbf{x}\|\leq r\}$, where $\mathbf{x}=[\|\tilde{\mathbf{q}}\| \quad \|\dot{\tilde{\mathbf{q}}}\|]^T$. On $\mathcal{D}_r$, by defining $\varepsilon=\frac{\tanh(r)}{r}$, a positive constant that depends on the size of $\mathcal{D}_r$, it is possible to state that $\|\tilde{\mathbf{q}}\|\geq\|\mathbf{tanh}(\tilde{\mathbf{q}})\|\geq\varepsilon\|\tilde{\mathbf{q}}\|$ on $\mathcal{D}_r$. Therefore, on $\mathcal{D}_r$ it is true that

$$\dot{V}\leq -\gamma\lambda_{\text{Max}}\{Q_{1D}\}\|\mathbf{x}\|^2+\delta_d\max\{\gamma,1\}\|\mathbf{x}\| \quad (24)$$

where

$$Q_{1D}=\begin{bmatrix} \varepsilon^2(K_{p_m}-k_{h2}) & -a-\frac{1}{\gamma}\frac{k_{h2}}{2} \\ -a-\frac{1}{\gamma}\frac{k_{h2}}{2} & \frac{1}{\gamma}\left[K_{v_m}-k_{h1}\right]-b\end{bmatrix}.$$

Notice that $\dot{V}\leq 0$ if $\mu_1\leq\|\mathbf{x}\|\leq r$, where $\mu_1$ is given by

$$\mu_1=\frac{\delta_d\max\{\gamma,1\}}{\gamma\lambda_{\text{Max}}\{Q_{1D}\}}.$$

To ensure that function (22) is positive definite and its time derivative is non-positive on a subset of the domain $\mathcal{D}_r$, the same tuning conditions that were presented in [25] should be met along with an additional one given by

$$K_{p_m}>\frac{[2\gamma a+k_{h2}]^2}{4\varepsilon^2\gamma\left[K_{v_m}-k_{h1}-\gamma b\right]}+k_{h2}$$

where $k_{h1}$ and $k_{h2}$ are defined in [44], and constants $a$ and $b$ are given by

$$a=\frac{1}{2}\left[K_{v_M}+k_{C_1}\|\dot{\mathbf{q}}_h\|_{\text{M}}+k_{h1}\right],$$
$$b=K_{M_M}+\sqrt{n}k_{C_1}.$$

Under these given conditions and invoking Theorem 4.18 from [45], it is possible to state that that there exists a time $T_C$ such that every solution with initial conditions

$$\|\mathbf{x}_0\|\leq r_0\sqrt{\frac{\alpha_1}{\alpha_2}}$$

where $r_0<r$, satisfies

$$\|\mathbf{x}(t)\|\leq\mu_1\sqrt{\frac{\alpha_2}{\alpha_1}},\quad \forall t\geq t+T_C. \quad (25)$$

In other words, during the optimal reaching phase, if an external disturbance is present, solutions enter a set that can be made arbitrarily small with appropriate tuning as a function of the upper bound on the disturbance.

If the switching time $T$ is greater than $T_C$, that is $T>T_C$, then solutions will converge and remain inside the set described by (25) until the switching time. After switching occurs, another transient state might ensue depending on the used gains during the holding phase. Therefore, to completely describe the effects of the external disturbance, the behavior during the holding phase must first be analyzed. To this end, notice that during the holding phase, the closed loop system may be expressed as

$$\frac{d}{dt}\begin{bmatrix}\tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}}\end{bmatrix}=\begin{bmatrix}\dot{\tilde{\mathbf{q}}} \\ M(\mathbf{q})^{-1}\left[-K_p\tilde{\mathbf{q}}-K_v\dot{\tilde{\mathbf{q}}}+\mathbf{z}_2(t)\right]\end{bmatrix}, \quad (26)$$

where

$$\mathbf{z}_2(t)=-C(\mathbf{q},\dot{\mathbf{q}})\dot{\tilde{\mathbf{q}}}-\mathbf{d}(t).$$

Notice also that after the switch, solutions at time $t=T$ take the value of the initial conditions for the system evolution (26).

By using the same function (22), assertions can be made regarding the stability of the solutions to (26). It can be shown that the time derivative of $V$ along the solutions of (26) satisfies

$$\dot{V}\leq -\gamma\lambda_{\text{Max}}\{Q_{2D}\}\|\mathbf{x}\|^2+\delta_d\max\{\gamma,1\}\|\mathbf{x}\| \quad (27)$$

on $\mathcal{D}_r$ with $Q_{2D}$ given by

$$Q_{2D} = \begin{bmatrix} \varepsilon^2 K_{p_m} & -\frac{1}{2}K_{v_M} \\ -\frac{1}{2}K_{v_M} & \frac{1}{\gamma}K_{v_m} - \sqrt{n}k_{C1} - K_{M_M} \end{bmatrix}.$$

This ensures that function (22) has a non-positive time derivative on a subset of the domain $\mathcal{D}_r$ given by $\mu_2 \leq \|\mathbf{x}\| \leq r$, where $\mu_2$ is defined as

$$\mu_2 = \frac{\delta_d \max\{\gamma, 1\}}{\gamma \lambda_{\text{Max}}\{Q_{2D}\}},$$

under the same tuning conditions that were presented in [25] with an additional one given by

$$\frac{4\varepsilon^2 K_{p_m} K_{v_m}}{K_{v_M}^2 + 4\varepsilon^2 K_{p_m}\left[\sqrt{n}k_{C_1} + K_{M_M}\right]} > \gamma.$$

Just as for the optimal reaching phase, under these given conditions and invoking Theorem 4.18 of [45], it is possible to state that there exists a time $T_F$ such that every solution with initial conditions

$$\|\mathbf{x}_0\| \leq r_0\sqrt{\frac{\alpha_1}{\alpha_2}}$$

where $r_0 < r$, satisfies

$$\|\mathbf{x}(t)\| \leq \mu_2\sqrt{\frac{\alpha_2}{\alpha_1}}, \quad \forall t \geq t + T_F. \tag{28}$$

This means that if an external disturbance is present, solutions during the holding phase also enter a set that can be made arbitrarily small with appropriate tuning.

Controller gains during the holding phase can be different from the ones used during the optimal reaching phase. Consequently, values for $\alpha_1$, $\alpha_2$ may be different for the holding phase. This also implies that if, for the optimal reaching phase, $T_C < T$ and the region of convergence is smaller for the holding phase than for the optimal reaching phase, another small transient might be present. On the other hand, if $T_C > T$, then the solutions can still be expected to fulfill (28).

In summary, if a bounded external disturbance is present it is possible to establish that there will be a deviation from the optimal trajectory that may be corrected up to a certain degree depending on the controller gain values.

### C. EXPERIMENTAL RESULTS

We have conducted three experiments with different prescribed trajectory durations.

- **Experiment 1**: duration $T = 1$ s and boundary values (20) were taken.
- **Experiment 2**: duration $T = 3$ s; initial positions and initial and final velocities of (20) were taken but $q(T) = (1.2, 1.1)^T$ rad instead.
- **Experiment 3**: duration $T = 5$ s and boundary values (20) were taken.

Note that the only difference between experiments 1 and 3 is the trajectory duration.

**TABLE 3. Performance index values for the controlled motion experiments held on a 2-DOF robotic manipulator subject to boundary conditions.**

| $T$ | Final positions $q(T)$ | PI with friction compensation | PI without friction compensation | PI for input signal $u_h$ |
|---|---|---|---|---|
| 1 s | $\begin{pmatrix}0.8\\1\end{pmatrix}$ rad | 174.264 | 53.888 | 36.421 |
| 3 s | $\begin{pmatrix}1.1\\1.2\end{pmatrix}$ rad | 368.568 | 75.124 | 16.856 |
| 5 s | $\begin{pmatrix}0.8\\1\end{pmatrix}$ rad | 533.95 | 70.508 | 5.696 |

Figure 9 compares the robot trajectory positions and torques with the input signals during the three experiments. Figures 9a, 9b and 9c correspond to experiment 1. Figures 9d, 9e and 9f correspond to experiment 2. Figures 9g, 9h and 9i correspond to experiment 3. Note that for the three experiments, the trajectory position was smoothly tracked. Negligible positioning errors were recorded along each trajectory. Final goal positions were attained with relative errors in the order of $10^{-3}$ rad at each joint for the three experiments. Note also that the experiments were extended past the prescribed trajectory duration $T$; afterwards, torque values remain constant so that the robot remains in its goal position.

As per equation (21), our control strategy compensates friction effects online during the experiments by taking into account the robot parameters. The applied control law in the experiment took these effects into account in order to cancel them out. The reader is referred to [25] for more details on this procedure. In order to analyze the impact of this compensation on the input signal, we have recorded the robot torques both with and without friction compensation. Let us remark that torques without friction compensation do not produce the required motion, these are just the robot torques with friction compensation to which we subtracted friction terms in (21) for analysis purposes. Therefore, Figures 9b, 9e and 9h show robot torques with friction compensation for experiments 1, 2 and 3 respectively. Additionally, Figures 9c, 9f and 9i show robot torques without friction compensation for experiments 1, 2 and 3 respectively.

When comparing robot torques with friction compensation against torques without friction compensation for each trajectory, we can see that this compensation does increase the required torques in order to produce the desired motion. We note that torques with friction compensation do follow the tendency of the optimal input signal, but noticeably modify its value. Online friction compensation naturally affects the performance index values for the experiment (see Table 3), which are larger for the actual robot motion (column 3) than for the simulation case (input signal, last column of Table 3). However, maximum output torque values $u$ remain very low compared to the maximum output that our robot can generate in all three experiments. Let us now present the specifics of each experiment.

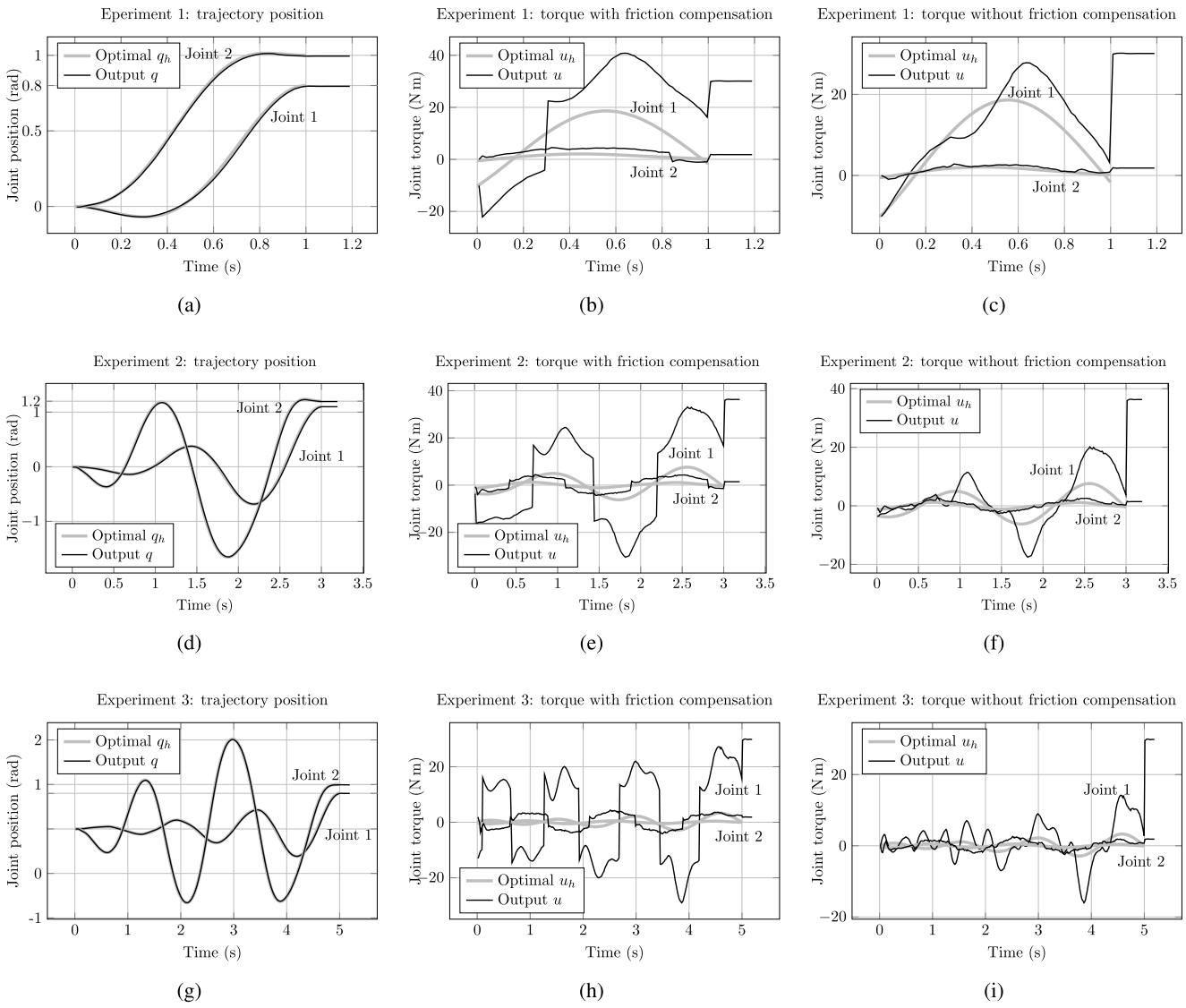



**FIGURE 9.** Controlled motion experiments on a 2-DOF manipulator platform. Experiment 1: $T = 1$ s; final positions $q(T) = (0.8, 1)^T$ rad; (a) trajectory position; (b) torque with friction compensation; (c) torque without friction compensation. Experiment 2: $T = 3$ s; final positions $q(T) = (1.1, 1.2)^T$ rad; (d) trajectory position; (e) torque with friction compensation; (f) torque without friction compensation. Experiment 3: $T = 5$ s; final positions $q(T) = (0.8, 1)^T$ rad; (g) trajectory position; (h) torque with friction compensation; (i) torque without friction compensation. Trajectory positions were smoothly tracked for each experiment.

### 1) EXPERIMENT 1

Optimal input signals $(q_h, u_h)$ were calculated with the four solvers used in section IV (NDSolve (shoot), NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation) according to the strategy described therein. For this particular example, differences between solvers were negligible and thus we retained the solution obtained with our QHFE/Perturbation method. This provided with optimal torques $u_h$, used as input signals, and optimal positions $q_h$ and velocities $\dot{q}_h$ that are compared with the feedback signals.

Figure 9a displays the optimal position $q_h$, used as reference for feedback, compared with the output system position $q$ during the experiment. As previously mentioned, the trajectory was smoothly tracked, following the reference during the experiment. Negligible positioning errors were recorded along the trajectory. Figure 9b compares the optimal torque $u_h$ with the system output $u$. Table 3 reports the performance index (PI) values obtained for this experiment in the first row. The increase in the performance index for the experiment (column 3) compared against the simulation case (column 5) is noticeable (almost five times as large) but results from the online friction compensation as previously mentioned.

### 2) EXPERIMENT 2

For this experiment, we also calculated optimal input signals $(q_h, u_h)$ with the four solvers used in IV (NDSolve (Shooting), NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation). However, as shown by Figure 4b, NDSolve

(Shooting) could not calculate the required motion because final conditions on position and velocity could not be met. The remaining three solvers (NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation) successfully calculated the required motion, satisfying all boundary conditions with little differences as shown also by Figure 4b. Here, we retained the solution obtained with our CHFE/Perturbation method. Note that we could have retained either of the other two solution methods for this experiment without affecting the outcome of the experiment because the obtained solutions are so similar.

Figure 9d displays the optimal position $q_h$, used as reference for feedback, compared with the output system position $q$ during the experiment. Again, the trajectory was smoothly tracked with negligible positioning errors along the trajectory. Figure 9e compares the optimal torque $u_h$ with the system output $u$. Table 3 reports the performance index values obtained for this experiment in the second row. The increase in the performance index for the experiment (column 3) compared against the simulation case (column 5) is also noticeable (this time being twenty times as large) and also results from the online friction compensation as previously mentioned.

#### 3) EXPERIMENT 3

For this final experiment, we calculated optimal input signals $(q_h, u_h)$ with the three solvers that succeeded experiment 2: NDSolve (RK), CHFE/Perturbation and QHFE/Perturbation. However, this time, NDSolve (RK) could not calculate the required motion because boundary conditions could not be met, as shown by Figure 4b. Both our CHFE/Perturbation and QHFE/Perturbation methods successfully calculated the required motion, exactly satisfying all boundary conditions. Here, we retained the solution obtained with our QHFE/Perturbation method. Note that we could have instead retained the CHFE/Perturbation method solution for this experiment without affecting its outcome.

Figure 9g displays the optimal position $q_h$, used as reference for feedback, compared with the output system position $q$ during the experiment. Again, the trajectory was smoothly tracked with negligible positioning errors along the trajectory. Figure 9h compares the optimal torque $u_h$ with the system output $u$. Table 3 reports the performance index values obtained for this experiment in the third row. The increase in the performance index for the experiment (column 3) compared against the simulation case (column 5) is large (this time being one hundred times as large) and also results from the online friction compensation as previously mentioned.

#### 4) EXPERIMENTS DISCUSSION

All of the experiments were extended past the prescribed trajectory durations $T$; afterwards, torque values remain constant so that the robot remains in its goal position. As mentioned before, the discrepancy between the robot torques required for motion and the optimal input signal is noticeable. Table 3 shows that when suppressing friction compensation from the output signal, the torque discrepancy with respect with the input signal does diminish. It does however stay large specially for experiment 3 (third row of Table 3).

We had previously established that the performance index should decrease as $T$ increases (see Figures 4a and 4b in section IV) when the boundary conditions on position and velocity stay equal. Surprisingly, this feature was not recovered in the experimental case, as shown by Table 3. Experiments 1 and 3 had the same final positions but the performance index with friction compensation is larger for experiment 3 where $T$ is five times longer.

Therefore, we hypothesize that these problems could be solved by incorporating nonconservative forces (such as friction effects) into our model (10) as prescribed in section 2.3 of [25]. By doing this, an online friction compensation could be avoided, thus reducing the absolute value of the required robot torques to produce optimal motion. This could result in preserving optimal input torques up to a certain degree and recovering a decreasing performance index whenever $T$ increases.

Another source of uncertainty that may cause the experimental performance of the controller to be different from the theoretical one may be the discretization error induced by the experimental setup. Measurements from the sensors embedded in the manipulator are taken every 2.5 milliseconds and the control law is computed with the same frequency. Although the overall effect of this phenomenon on the performance of the control scheme should not be as important as friction, it may also have an impact on the discrepancies found between theoretical results and the experiments.

We should emphasize however that even in the experimental case, all of the trajectories were tracked with good accuracy. Also, while the trajectory for experiment 1 could be calculated with all of the solvers, experiment 2 could not be held with the NDSolve (Shooting) solver, and experiment 3 could not be held with either NDSolve (Shooting) or its RK variant. On the contrary, our proposed CHFE/Perturbation and QHFE/Perturbation solvers were able to complete all of the experiments required trajectories.

These results show that the optimal trajectory obtained with our proposed HFE/Perturbation method can be tracked with good accuracy and thus validate our approach. We now proceed to conclude our paper by recalling the main advantages of our proposed method, as well as summarizing our findings in the next section.

## VI. CONCLUDING REMARKS

Upon establishing a coupled set of second order, nonlinear, covariant ODE controlling torques and trajectories that involve the covariant derivatives of the gravitational potential and the Riemann-Christoffel curvature tensor, an algorithm has been elaborated for finding the optimal torques and trajectories. This algorithm consists in solving the system of ODE that minimizes an objective functional of the torques, regarded as a performance index.

Our procedure approximates the configuration parameters and the torques by piecewise HFE interpolations in the framework of the FEM. CHFE and QHFE are proposed as a more precise alternative. Physical interpretation of finite elements coefficients are given: approximations of position, velocity and acceleration are directly obtained at each node. Two examples illustrate the superconvergence of these interpolations when simulating the torques and trajectories. A perturbation method involving Hermite finite elements for time integration was proposed. Therefore, explicit expressions for perturbed motion and control equations were given.

The presented HFE/Perturbation method was specifically designed and applied to solve the OCP of a robotic manipulator. However, it can be easily adapted to solve general nonlinear second order ODE arising in multibody systems dynamics. Solutions approximations obtained with our algorithm were compared with those obtained with a commercial ODE solver, on the basis of specific performance indicators, and reveal to provide better performance overall, provided that sufficient nodes are being used. Additionally, accuracy of the obtained solutions approximations can be easily increased as required. Note also that stiffness does not affect our method for fairly large trajectory durations, thus enabling further analysis of the optimal control procedure. Therefore, when compared with the commercial ODE solver, our FE-based method has the following advantages:

- it directly provides with optimal positions and torques, along with their respective first (with CHFE) and second (with QHFE only) time derivatives at each node;
- due to the nature of the FE interpolation, there is no need to shoot the final conditions nor to rebuild the control variables;
- lower error norms can be obtained because further differentiation of positions and torques is not required;
- longer duration trajectories can be calculated.

With respect to this last point, i.e., in terms of stability with respect to trajectory duration, CHFE/Perturbation resulted to be the most stable. Our solver was able to provide solutions for durations of up to 12 s for the optimal motion of a 2-DOF robotic manipulator, where the commercial ODE solver could not go further than 4.7 s.

A convergence analysis of our proposed HFE/Perturbation algorithms revealed that the superconvergence of HFE interpolations is not inherited when applied to our optimal control methodology. However, convergence of the evaluated norms with CHFE/Perturbation resulted to be at least in the order of the cubic interpolations and even higher for errors of $q$ (position) and $u$ (torques). Interestingly, convergence of the evaluated norms with QHFE/Perturbation were lower than expected, at a rate of one degree lower than the order of the quintic interpolations.

Generally speaking, CHFE/Perturbation is more stable with respect to trajectory duration, and more cost effective (in the numerical sense) than QHFE/Perturbation. However, the latter still displays higher convergence rates and may be desirable whenever first and second time derivatives of positions and torques are required at each node.

Finally, experiments were conducted with a robotic manipulator by directly feeding optimal torques obtained with our method, as inputs. This resulted in the generation of smooth trajectories that met specified boundary conditions, thus validating our approach. Our algorithm is therefore well suited for efficiently generating optimal motion for robotic systems. However, and unlike what we had established for optimal motion simulations, the performance index did not decrease when the trajectory duration increased. On the contrary, online friction compensation and parametric uncertainties noticeably affected the performance index so that it increased with trajectory duration instead. Therefore, our optimal control method could be improved by taking nonconservative forces into account in our model. This should result in preserving optimal input torques up to a certain extent, directly leading to lower experimental torques values for motion. This will be the object of further research for which the presented HFE/Perturbation method shall be used to solve the arising nonlinear ODE.

The control scheme was shown to have robustness properties with respect to external disturbances. Namely, it was shown, using Lyapunov based analysis, that if a bounded external disturbance is present it is possible to establish that there will be a bounded deviation from the optimal trajectory. This also implies that in spite of the external disturbance, the closed-loop solutions remain bounded. In practice, this means that external disturbances would not render the system unstable.

In summary, we have presented a time FE-based method that effectively solves the optimal control problem of robotic manipulators involving covariant control equations as optimality conditions; the proposed control scheme provides the optimized control action required to take the robot from an initial to a final state in a prescribed time. This FE-based method was compared with a commercial ODE solver and revealed to be more stable for extended prescribed trajectory times. Experimental results on a robotic manipulator system validated our theoretical proposal. Motion trajectories were smoothly tracked and results exposed further methodology improvements.

## REFERENCES

[1] K. M. Liew, X. Q. He, T. Y. Ng, and S. Sivashanker, "Active control of FGM plates subjected to a temperature gradient: Modelling via finite element method based on FSDT," *Int. J. Numer. Methods Eng.*, vol. 52, no. 11, pp. 1253–1271, Dec. 2001, doi: 10.1002/nme.252.

[2] P. Mäkinen and J. Mattila, "Finite element-based control of a single-link flexible hydraulic manipulator," in *Proc. ASME/BATH 2017 Symp. Fluid Power Motion Control* (Fluid Power Systems Technology), Oct. 2017, pp. 1–10, doi: 10.1115/FPMC2017-4264.

[3] T. M. Bieze, F. Largilliere, A. Kruszewski, Z. Zhang, R. Merzouki, and C. Duriez, "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *Soft Robot.*, vol. 5, no. 3, pp. 348–364, 2018, doi: 10.1089/soro.2017.0079.

[4] P. Ning, Z.-Q. Feng, J. A. R. Quintero, Y.-J. Zhou, and L. Peng, "Uzawa algorithm to solve elastic and elastic–plastic fretting wear problems within the bipotential framework," *Comput. Mech.*, vol. 62, no. 6, pp. 1327–1341, Dec. 2018, doi: 10.1007/s00466-018-1567-8.

[5] Y.-J. Zhou, Z.-Q. Feng, J. A. R. Quintero, and P. Ning, "A computational strategy for the modeling of elasto-plastic materials under impact loadings," *Finite Elements Anal. Design*, vol. 142, pp. 42–50, Mar. 2018, doi: 10.1016/j.finel.2018.01.003.

[6] F. Williamson, "Richard courant and the finite element method: A further look," *Historia Math.*, vol. 7, no. 4, pp. 369–378, 1980, doi: 10.1016/0315-0860(80)90001-4.

[7] G. A. Evans, J. M. Blackledge, and P. D. Yardley, *Numerical Methods for Partial Differential Equations*. London, U.K.: Springer, 2000, doi: 10.1007/978-1-4471-0377-6.

[8] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis Fundamentals*, 7th ed. London, U.K.: Butterworth-Heinemann, 2013, doi: 10.1016/B978-1-85617-633-0.00001-0.

[9] G. Allaire and A. Craig, *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation* (Numerical Mathematics and Scientific Computation). Oxford, U.K.: Oxford University Press, 2007.

[10] G. Marchouk and V. Agochkov, *Introduction aux Méthodes des éléments Finis (French)*. Moscow, Russia: Mir, 1985.

[11] A. Eriksson, "Temporal finite elements for target control dynamics of mechanisms," *Comput. Struct.*, vol. 85, nos. 17–18, pp. 1399–1408, Sep. 2007, doi: 10.1016/j.compstruc.2006.08.080.

[12] M. Dorosti, R. H. B. Fey, M. F. Heertjes, M. M. J. van de Wal, and H. Nijmeijer, "Iterative pole-zero finite element model updating using generic parameters," *Mechatronics*, vol. 55, pp. 180–193, Nov. 2018, doi: 10.1016/j.mechatronics.2018.06.012.

[13] L. Paunonen and D. Phan, "Reduced order controller design for robust output regulation," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2480–2493, Jun. 2020, doi: 10.1109/TAC.2019.2930185.

[14] D. Wang and X. Chen, "Closed-loop simulation integrating finite element modeling with feedback controls in powder bed fusion additive manufacturing," in *Proc. Int. Symp. Flexible Autom.*, Jul. 2020, pp. 1–7, doi: 10.1115/isfa2020-9611.

[15] G. Zheng, O. Goury, M. Thieffry, A. Kruszewski, and C. Duriez, "Controllability pre-verification of silicone soft robots based on finite-element method," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7395–7400, doi: 10.1109/ICRA.2019.8794370.

[16] R. Cavin and S. Tandon, "Distributed parameter system optimum control design via finite element discretization," *Automatica*, vol. 13, no. 6, pp. 611–614, 1977, doi: 10.1016/0005-1098(77)90082-6.

[17] J. Liu and Z. Zhou, "Finite element approximation of time fractional optimal control problem with integral state constraint," *AIMS Math.*, vol. 6, no. 1, pp. 979–997, 2021, doi: 10.3934/math.2021059.

[18] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," *Fast Motions in Biomechanics and Robotics* (Lecture Notes in Control and Information Sciences), vol. 340. Berlin, Germany: Springer, 2006, doi: 10.1007/978-3-540-36119-0_4.

[19] A. Eriksson and A. Nordmark, "Temporal finite element formulation of optimal control in mechanisms," *Comput. Methods Appl. Mech. Eng.*, vol. 199, nos. 25–28, pp. 1783–1792, May 2010, doi: 10.1016/j.cma.2010.02.003.

[20] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2011.

[21] G. R. G. da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-driven LQR control design," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 180–185, Jan. 2019, doi: 10.1109/LCSYS.2018.2868183.

[22] T. Çimen, "State-dependent Riccati equation (SDRE) control: A survey," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 3761–3775, 2008, doi: 10.3182/20080706-5-KR-1001.00635.

[23] S. R. Nekoo, "Tutorial and review on the state-dependent Riccati equation," *J. Appl. Nonlinear Dyn.*, vol. 8, no. 2, pp. 109–166, Jun. 2019, doi: 10.5890/JAND.2019.06.001.

[24] A. Grancharova and T. A. Johansen, *Survey of Explicit Approaches to Constrained Optimal Control*. Berlin, Germany: Springer, 2005, pp. 47–97, doi: 10.1007/978-3-540-30560-6_3.

[25] J. A. Rojas-Quintero, J. A. Rojas-Estrada, J. Villalobos-Chin, V. Santibañez, and E. Bugarin, "Optimal controller applied to robotic systems using covariant control equations," *Int. J. Control*, pp. 1–14, Jan. 2021, doi: 10.1080/00207179.2020.1865570.

[26] F. Dubois, D. Fortuné, J. A. R. Quintero, and C. Vallée, "Pontryagin calculus in Riemannian geometry," in *Geometric Science of Information*, F. Nielsen and F. Barbaresco, Eds. Cham, Switzerland: Springer, 2015, pp. 541–549, doi: 10.1007/978-3-319-25040-3_58.

[27] Z. Li, C. Li, S. Li, and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7469–7478, Dec. 2020, doi: 10.1109/TII.2019.2957186.

[28] Z. Li, W. Zuo, and S. Li, "Zeroing dynamics method for motion control of industrial upper-limb exoskeleton system with minimal potential energy modulation," *Measurement*, vol. 163, Oct. 2020, Art. no. 107964, doi: 10.1016/j.measurement.2020.107964.

[29] P. Grinfeld, *Introduction to Tensor Analysis and the Calculus of Moving Surfaces*. New York, NY, USA: Springer, 2013, doi: 10.1007/978-1-4614-7867-6.

[30] A. Nikoobin and M. Moradi, "Optimal balancing of robot manipulators in point-to-point motion," *Robotica*, vol. 29, no. 2, pp. 233–244, Mar. 2011.

[31] C. Mirz, F. Schöler, J. P. Barreto, and B. Corves, "Optimal control based path planning for parallel kinematic manipulators utilising natural motion," in *Proc. IEEE 14th Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2018, pp. 223–228.

[32] M. R. Vezvari, A. Nikoobin, and A. Ghoddosian, "Zero-power balancing a two-link robot manipulator for a predefined point-to-point task," *J. Mech. Sci. Technol.*, vol. 34, no. 6, pp. 2585–2595, Jun. 2020.

[33] M. Asgari and A. Nikoobin, "Analysis of optimal dynamic manipulation for robotic manipulator based on Pontryagin's minimum principle," *Arabian J. Sci. Eng.*, vol. 45, no. 11, p. 9159—9169, 2020.

[34] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 362–375, Aug. 1992, doi: 10.1177/027836499201100408.

[35] M. H. Holmes, *Introduction to Perturbation Methods* (Texts in Applied Mathematics), vol. 20, 2nd ed. New York, NY, USA: Springer-Verlag, 2013, doi: 10.1007/978-1-4614-5477-9.

[36] A. Beléndez, E. Arribas, M. Ortuño, S. Gallego, A. Márquez, and I. Pascual, "Approximate solutions for the nonlinear pendulum equation using a rational harmonic representation," *Comput. Math. with Appl.*, vol. 64, no. 6, pp. 1602–1611, Sep. 2012, doi: 10.1016/j.camwa.2012.01.007.

[37] E. Salinas-Hernández, G. A. de Parga, S. Domínguez-Hernández, and R. Muñoz-Vega, "An analytical approximation of a pendulum trajectory," *Eur. J. Phys.*, vol. 35, no. 4, Jul. 2014, Art. no. 045027, doi: 10.1088/0143-0807/35/4/045027.

[38] A. Beléndez, C. Pascual, D. I. Méndez, T. Beléndez, and C. Neipp, "Exact solution for the nonlinear pendulum," *Revista Brasileira de Ensino de Física*, vol. 29, no. 4, pp. 645–648, 2007, doi: 10.1590/S1806-11172007000400024.

[39] F. Reyes and R. Kelly, "Experimental evaluation of identification schemes on a direct drive robot," *Robotica*, vol. 15, no. 5, pp. 563–571, Sep. 1997, doi: 10.1017/S0263574797000659.

[40] Y. Zhang, S. Li, and X. Zhou, *A Survey of Near-Optimal Control of Nonlinear Systems*. Cham, Switzerland: Springer, 2020, pp. 1–20, doi: 10.1007/978-3-030-33384-3_1.

[41] F. R. Cortés and R. Kelly, "Experimental evaluation of model-based controllers on a direct-drive robot arm," *Mechatronics*, vol. 11, pp. 267–282, Apr. 2001, doi: 10.1016/S0957-4158(00)00008-8.

[42] R. Campa, R. Kelly, and V. Santibáñez, "Windows-based real-time control of direct-drive mechanisms: Platform description and experiments," *Mechatronics*, vol. 14, no. 9, pp. 1021–1036, Nov. 2004, doi: 10.1016/j.mechatronics.2004.04.004.

[43] S. Sánchez-Mazuca and R. Campa, "An improvement proposal to the static friction model," *Math. Problems Eng.*, vol. 2013, Jun. 2013, Art. no. 946526, doi: 10.1155/2013/946526.

[44] R. Kelly, V. Santibá nez, and J. A. Loría, *Control of Robot Manipulators in Joint Space* (Advanced Textbooks in Control and Signal Processing). London, U.K.: Springer-Verlag, 2005.

[45] H. Khalil, *Nonlinear Systems* (Pearson Education). Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

**JUAN ANTONIO ROJAS-QUINTERO** received the B.E. degree in science and technology of mechanics and engineering, the M.Sc. degree in mechanics and engineering sciences, and the Ph.D. degree in mechanics and engineering sciences from the University of Poitiers, Poitiers, France, in 2007, 2009, and 2013, respectively.

He was a Postdoctoral Fellow with the School of Mechanics and Engineering, Southwest Jiaotong University, Chengdu, China, from 2014 to 2016. He is currently a CONACYT Research Fellow with the National Technology Institute (Tecnológico Nacional de México), Ensenada Campus, Mexico. His research interests include nonlinear dynamical systems, optimal control, and robotics.

**JORGE VILLALOBOS-CHIN** received the B.E. degree in mechatronic engineering and the M.Sc. degree in electric engineering from the Laguna Institute of Technology, Torreon, Coahuila, Mexico, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His research interests include stability analysis, nonlinear control design, and control of mechanical systems.

**VICTOR SANTIBANEZ** (Member, IEEE) received the B.E. and M.Sc. degrees in electronic engineering from the Instituto Tecnológico de La Laguna, Torreon, Coahuila, Mexico, in 1977 and 1984, respectively, and the Ph.D. degree from CICESE, Mexico, in 1997. He has coauthored two books on robot control: *Control of Robot Manipulators* (Prentice Hall, 2003) and *Control of Robot Manipulators in Joint Space* (Springer, 2005). He is the author or coauthor of peer-reviewed journal articles and international conference papers. He is currently a Professor and a Researcher with the Instituto Tecnológico de la Laguna. His research interests include robot control, nonlinear systems control, fuzzy control, and adaptive control.