

Time Optimal Control of Variable Stiffness Actuated Systems

Altay Zhakatayev, *Member, IEEE*, Matteo Rubagotti, *Member, IEEE*, Huseyin Atakan Varol, *Senior Member, IEEE*

Abstract—Variable-stiffness actuation exhibits promising features for obtaining human-like behavior and safer human robot physical interaction. Task planning and closed-loop control of these systems pose many challenges due to their complicated structure, and the need of satisfying many constraints during task execution. This paper introduces a framework for the design and numerical solution of time-optimal control problems for VSA systems. Two different time-optimal control problems, namely “minimum time for target performance” and “minimum time for maximum performance”, are formally defined, and methods for solving them are presented based on existing numerical software tools for nonlinear optimization. Two experimental case studies, focusing on ball throwing tasks with antagonistically-actuated VSA systems, are used to test the presented methods and show their validity.

Index Terms—Variable stiffness actuation, variable impedance actuation, nonlinear optimization, time-optimal control, robotics.

I. INTRODUCTION

In the structured environment of factories, robots outperform human workers in tasks requiring fast, repetitive and precise manipulation. Their performance leaves a lot to be desired in unstructured, human-oriented manipulation environments. A vivid example of this was the recent Darpa Grand Challenge assessing the potential of humanoid robots in disaster settings [1]. Robots were assigned rather simple tasks such as opening a door, climbing a ladder and grabbing a tool, most of which are easily doable by a kindergarten-age child. The first iteration of this challenge fell short of meeting the expectations. Reaching human agility, dexterity and versatility for task execution in unstructured environments will remain as a central research problem for years to come. While one part of this enigma lies in the perception and machine intelligence, another vital component is biomimetic actuation.

In order to achieve human-level performance and safety in human-robot interaction, researchers concentrated their efforts on the development of a new actuation paradigm called *variable impedance actuation* [2]. In these systems, robot joints are decoupled from links using passive energy storing and/or dissipating elements. The robot, using multiple degrees of actuation for a link, can simultaneously change the position

This work was partially supported by the Ministry of Education and Science of the Republic of Kazakhstan under grant “Optimal Design and Control of Variable Impedance Actuated Robots”.

A. Zhakatayev and H. A. Varol are with the Dept. of Robotics and Mechatronics, Nazarbayev University, 53 Kabanbay Batyr Ave, Z05H0P9 Astana, Kazakhstan. Email: {azhakatayev, ahvarol}@nu.edu.kz. Corresponding author: H. Atakan Varol.

M. Rubagotti is with the Department of Engineering, University of Leicester, Leicester LE1 7RH, United Kingdom (e-mail: mr298@le.ac.uk).

and also the physical impedance parameters of the link. The advantages of this type of robots compared to the traditional rigid ones are discussed in [3]. While researchers continue to work on ingenious mechanical designs for variable impedance actuation [4]–[8], the control of these systems started to attract more attention [9]–[12]. Task planning and control of these robots are daunting problems due to the highly-constrained and nonlinear nature of these systems. Braun et al. [9] formulated motor trajectory generation of variable stiffness actuated (VSA) robots as an optimal control problem (OCP). Zhakatayev et al. [10] utilized nonlinear model predictive control (NMPC) for the closed-loop control of VSA robots to reliably track the references obtained from the solution of an OCP. These developments enable the treatment of a wider family of optimal control problems for VSA robots such as time and energy optimal control.

In particular, the definition of time-optimal trajectories in robotic manipulators was first considered in [13] and [14]: these two works independently proposed similar methods for time-optimal control of rigidly actuated systems with a priori specified paths, and upper bounds on control variables (actuator forces and torques). Afterwards, time-optimal control problems for robot manipulators were developed to handle more complex tasks (e.g., point-to-point motion [15] and motion in dynamic environments [16]), recently exploiting advances in optimization algorithms and related software tools [17]–[20]. Verscheure et al. [18] transformed the time-optimal tracking problem for robot systems into a convex OCP by utilizing nonlinear change of variables and convexity-preserving extensions. Stochastic planning using cubic splines was employed for minimum-time trajectory planning of a five-bar parallel robot in [20].

The time-optimal control of VSA robots was first considered for the so-called *safe brachistochrone* problem, defined for variable-stiffness and variable-impedance actuators in [21]–[23], with emphasis on analyzing the optimal stiffness/impedance modulation during the robot motion for simple configurations, rather than on defining numerical tools to be used with general robot topologies. In [24], a closed-form solution was determined for reaching maximum link speed in minimum time for robots with elastic joints, to analytically determine suitable stiffness values for executing explosive motions. In [25], the problem of braking a visco-elastic joint in minimum time was considered, and the result was then extended to a near-optimal real-time control of elastic robot manipulators with an arbitrary number of degrees of freedom.

In this work, we present a general framework for the time-optimal motion planning of VSA robots with any kind of

topology, subject to any type of set-membership constraints on inputs and states (e.g., limits on actuator torques and velocities, maximum extension or compression of the elastic elements, maximum link speeds, etc.). After recalling the general modeling framework for VSA robots (Section II), two different types of optimal control problems (OCPs) are considered. The first, namely “minimum time for target performance” (Section III), aims at finding the control (and corresponding state) signals that minimize the time for executing a task, while achieving a given performance (e.g., throwing an object at a specific distance). The proposed formulation allows one to minimize the overall execution time, including events that take place after the robot motion, but are influenced by the motion itself (e.g., one can minimize the time from when the object-throwing motion starts, until the moment when the object is supposed to touch the ground). The second, “minimum time for maximum performance” (Section IV), has the objective of minimizing the time in which the robot motion happens, at the same time maximizing the obtainable performance (e.g., minimize the time for the throwing motion, while throwing the object at the maximum achievable distance for the given robot configuration and constraints). For both kinds of problems, which in general can be formulated as nonlinear programs, the solution is determined numerically, by using the ACADO Toolkit [26]. The obtained control sequences are not provided to the robot when executing the task: instead, the corresponding state trajectories are provided as reference to an NMPC scheme, which allows closed-loop control (with consequent reduction of sensitivity to external disturbances and parameter uncertainties) while at the same time enforcing constraint satisfaction. The NMPC controller, briefly described in Section V, is designed as described in [10], and tested experimentally on a ball throwing task. More precisely, the “minimum time for target performance” problem is tested on a two-link antagonistically actuated VSA manipulator, and the “minimum time for maximum performance” problem is tested on a single link planar manipulator augmented with a reaction wheel (Section VI). To the best of the authors’ knowledge, the presented general framework for minimum-time control of VSA systems has never been reported in the literature. Our main contribution is in the definition of a practical method to allow engineers to plan minimum-time tasks by using a single software tool for a variety of different VSA systems.

II. MODELING FRAMEWORK FOR VSA SYSTEMS

For the modeling framework of VSA systems, the same approach as in [9], [10] will be followed, which divides the set of coordinates into two subsets. The *motor-side* coordinates $\theta \in \mathbb{R}^{n_\theta}$ describe the angular positions (reflected through gear reduction) of the electrical servomotors in the compliant actuators. The *link-side* coordinates $q \in \mathbb{R}^{n_q}$ describe the angles of robot joints and possibly reaction wheels added to the robot structure. The addition of reaction wheels makes it possible to exploit reactive torques, so as to improve performance for the given task [27]. The position control of servomotors is performed by an internal control loop, which receives the vector of angular position references $\theta_d \in \mathbb{R}^{n_\theta}$. Under the

standard assumption, for the servomotors, of high transmission ratio and/or of high-gain feedback position controllers, the motor-side dynamics can be considered as decoupled from the link-side (see [9, Sec. III]). The closed-loop dynamics of each servomotor, given by the action of an embedded controller, can be typically represented by a second-order critically-damped linear system

$$\ddot{\theta}_i + 2\kappa_i \dot{\theta}_i + \kappa_i^2 \theta_i = \kappa_i^2 \theta_{d,i}, \quad i = 1, \dots, n_\theta \quad (1)$$

where θ_i and $\theta_{d,i}$ are the components of θ and θ_d , respectively, while $\kappa_i \in \mathbb{R}_{>0}$ are constants that describe the dynamics of the closed-loop system. The link-side dynamics (which includes the dynamics of reaction wheels) is described by

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + G(q) = \tau_E(\tau_d, q, \theta) \quad (2)$$

where the inertia matrix $M(q) \in \mathbb{R}^{n_q \times n_q}$ is such that $M = M' > 0$ (i.e., M is symmetric and positive definite), $C(q, \dot{q}) \in \mathbb{R}^{n_q \times n_q}$ is the matrix of Coriolis and normal inertial forces, $D\dot{q} \in \mathbb{R}^{n_q}$ describes the effect of viscous friction, while $G(q) \in \mathbb{R}^{n_q}$ is the gravity force term. The value of $\tau_E(\tau_d, q, \theta) \in \mathbb{R}^{n_q}$ includes instead the joint torques generated by the elastic elements, together with the torques generated directly by current-controlled electric motors which actuate the reaction wheels. The torques actuating the reaction wheels can be compactly represented in a vector $\tau_d \in \mathbb{R}^{n_d}$, n_d being the number of reaction wheels.

In order to define the optimal control problem, a nonlinear state-space model of the whole system is necessary, which includes the dynamics of motor-side and link-side variables. In particular, the input vector is defined as

$$u \triangleq [\theta'_d \quad \tau'_d]' \in \mathbb{R}^{n_u}, \quad (3)$$

where $n_u \triangleq n_\theta + n_d$, while the state vector is

$$x \triangleq [q' \quad \dot{q}' \quad \theta' \quad \dot{\theta}]' \in \mathbb{R}^{n_x}, \quad (4)$$

where $n_x \triangleq 2n_q + 2n_\theta$. The state-space dynamics can be written as

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{q} \\ -M^{-1}(C(q, \dot{q})\dot{q} + D\dot{q} + G(q) - \tau_E(\tau_d, q, \theta)) \\ \dot{\theta} \\ -B\dot{\theta} - K\theta + K\theta_d \end{bmatrix} \quad (5)$$

where $B \triangleq \text{diag}\{2\kappa_i\} \in \mathbb{R}^{n_\theta \times n_\theta}$ and $K \triangleq \text{diag}\{\kappa_i^2\} \in \mathbb{R}^{n_\theta \times n_\theta}$ are matrices describing the motor-side dynamics.

III. MINIMUM TIME FOR TARGET PERFORMANCE

In the first type of problems considered in this paper, it is assumed that the robot has to satisfy a specific goal, referred to as *target performance*, in minimum time. A formulation of this class of problems is given in the following subsection.

A. Problem formulation

Let $u(t) \in \mathbb{R}^{n_u}$ be a realization of the control input, and let $u_{[0, T]}$ represent the time evolution of $u(t)$ in $[0, T]$, $T \in \mathbb{R}_{\geq 0}$. In general, the OCP determines the length $T = T^*$ of the minimum-time interval in which the target performance

can be achieved, and the control signal $u_{[0,T^*]}^*$ that leads to such an achievement. The OCP formulation only provides an optimal planning for the system dynamics, without considering problems related to model mismatches and external disturbances, therefore resulting in an open-loop control problem. A feedback scheme based on NMPC, briefly described in Section V, will be implemented to make the actual evolution of the system variables as close as possible to the planned one. A rigorous formulation of the OCP is now given, and each term in it is explained

$$[T^*, u_{[0,T^*]}^*] = \arg \min_{[T, u_{[0,T]}]} T + \mathfrak{D}(x(T), u(T)) \quad (6a)$$

$$\text{subject to } x(t) = x(0) + \int_0^t f(x(\tau), u(\tau)) d\tau \quad (6b)$$

$$[x(t) \ u(t)]' \in \mathcal{Z}, \forall t \in [0, T] \quad (6c)$$

$$\phi(x(T), u(T)) \in \Phi \quad (6d)$$

In (6a), T is the time required for the robot to execute the part of the task in which it is directly involved, while the term $\mathfrak{D}(x(T), u(T))$ accounts for a time interval during which the task is completed without direct involvement of the robot (if not needed, one can simply set $\mathfrak{D}(x(T), u(T)) = 0$). To clarify, imagine that the end effector of a robot has to hit a cart, and push it against a wall. If the task consists of minimizing the time at which the cart hits the wall, it is not possible to achieve it by simply minimizing T . In such a case, $\mathfrak{D}(x(T), u(T))$ would be the time required for the cart to hit the wall, after being hit at time T by the robot. Condition (6b), in which $x(0)$ is the fixed initial state, integrates the state equation (5), thus introducing the system dynamics in the OCP. In (6c), $\mathcal{Z} \in \mathbb{R}^{n_x+n_u}$ is a set defined in the space of state and input variables, which describes the constraints that have to be satisfied for proper operation of the robot. Typical examples are upper and lower bounds on angular positions and velocities of the actuators and of the links, and minimum and maximum displacement for the elastic elements. Condition (6d) imposes that a map $\phi(x(T), u(T)) : \mathbb{R}^{n_x+n_u} \rightarrow \mathbb{R}_\phi$ belongs to a given set $\Phi \in \mathbb{R}_\phi$ at time T : this is the constraint that imposes the target performance. In the mentioned example of the robot hitting the cart, $\phi(x(T), u(T)) \in \mathbb{R}$ can be the distance that the cart needs to travel in order to hit the wall, and is a function of robot variables at the time when the cart is hit. Imagine that the wall is situated at a distance of 1 m from the initial position of the cart. In such a case, Φ can be the set of distances greater or equal to 1 m, and (6d) would be expressed as $\phi(x(T), u(T)) \geq 1$.

The performance index $\phi(\cdot, \cdot)$ only accounts for the system variables at time T , which is suitable for explosive movement tasks. A generalization can be made by simply adding another term, which would account for the system evolution in $[0, T]$, cf. [10, Eq. 6e].

Another relevant example that can be formulated using (6) is the minimum-time execution of a point-to-point motion. For instance, a humanoid robot with a VSA arm similar to the one in [28] is given a task of hammering nails on an assembly line. The arm needs to move from an initial position and reach a specific position at a desired velocity. Minimizing the time of

this task would increase throughput, and hence productivity.

In the following, the evolution of the state variables associated with the optimal input $u^*(t)$ is denoted as $x^*(t)$. In order to find them, a numerical solution of the OCP (6) is obtained (for the shown experimental results, using the ACADO Toolkit), as described in the following subsection.

B. Numerical solution

In order to describe the input signal $u^*(t)$ in a time interval $[0, T]$, an analytical solution is needed. Since this is not possible in any practical case, OCP (6) is solved using numerical methods. In particular, the ACADO Toolkit [26] is employed in this paper. The infinite-dimensional description of the input signal is discretized, by assuming that the control signal $u(t)$ is piecewise constant with discretization step equal to $T_d \triangleq T/N_d \in \mathbb{R}_{>0}$, where N_d is the a-priori fixed number of nodes along the time interval $[0, T]$. The cost function will therefore be determined by T_d , and by the value of $\mathfrak{D}(x, u)$ evaluated at $T = N_d T_d$. For the examples shown in this paper, we choose to employ the standard settings of the ACADO Toolkit, using a multiple shooting discretization with N_d nodes, with the numerical integration of the continuous-time differential equations performed via 4th or 5th-order Runge-Kutta methods. The discretized mathematical program is solved via sequential quadratic programming (SQP).

IV. MINIMUM TIME FOR MAXIMUM PERFORMANCE

In some cases, instead of defining a target performance, the objective can consist of maximizing such performance, and finding the minimum time interval in which this can be achieved. For instance, in the previous example, if no wall is present, the task might be to maximize the distance at which the cart will eventually stop due to friction. In order to define a well-posed problem, we need to assume that the obtainable performance is a bounded function of the execution time, which is a reasonable assumption in most practical cases. In case the performance can be indefinitely increased as the available time interval increases, this kind of problem cannot be formulated. Again in the considered cart example, given the constraints on robot motion, the force with which the cart can be hit is also bounded, and cannot be indefinitely increased as the number of swings of the arm increases. Considering the increased level of complexity of this kind of problem as compared to that described in Section III, the term $\mathfrak{D}(x(T), u(T))$ is not taken into account. For the described cart example, this corresponds to minimizing the time when the robot touches the cart, having this latter stop at the maximum possible distance.

A. Problem formulation

Formally, the performance ϕ^* associated with the optimal time interval T^* and corresponding control signal $u_{[0,T^*]}^*$ will have to satisfy

$$\phi^* = \max_{T \in [0, +\infty)} \phi_T^* \quad (7)$$

where

$$\phi_T^* \triangleq \max_{u_{[0,T]}} \phi(x(T), u(T)). \quad (8)$$

In (8), ϕ_T^* is the maximum achievable performance for all possible control sequences $u_{[0,T]}$ in the *fixed* time interval $[0, T]$. The control sequence that maximizes $\phi(x(T), u(T))$ in (8) is referred to as $u_{[0,T]}^*$. Given all possible fixed time intervals $[0, T]$, ϕ^* in (7) is therefore the maximum (bounded by assumption) achievable performance in all possible time intervals. The OCP for maximum performance is formulated as

$$[T^*, u_{[0,T^*]}^*] = \arg \min_{[T, u_{[0,T]}]} T \quad (9a)$$

$$\text{subject to } \phi(x(T), u(T)) + \delta \geq \phi^* \quad (9b)$$

$$(6b), (6c) \quad (9c)$$

Condition (9b) imposes performance maximization. The fixed scalar $\delta \in \mathbb{R}_{\geq 0}$ can be set equal to zero, in case the performance has to be exactly maximized. In the case when a performance slightly lower than the maximum ϕ^* is satisfactory, δ can be defined as positive. This can significantly reduce the value of T^* , in cases when, after a given time interval T , the performance has a negligible increase, which does not constitute a real difference for the task at hand. A positive δ would also eliminate possible problems related to small numerical errors in the solutions. Notice that the constraint on target performance (6d) in OCP (6) is here substituted by (9b), which requires performance maximization. OCP (9) is more complex than (6), because the term ϕ^* in the constraint (9b) is in turn defined as the outcome of the maximization described in (7)-(8). As a consequence, one cannot numerically solve it as a single optimization problem (for instance using the ACADO Toolkit) as described in Section III-B: a reformulation of problem (9) is therefore needed.

B. Solution via time axis grid

In order to find a solution to OCP (9), some observations are necessary. For a fixed value of T , one can define an OCP for maximizing performance

$$u_{[0,T]}^* = \arg \max_{u_{[0,T]}} \phi(x(T), u(T)) \quad (10a)$$

$$\text{subject to } (6b), (6c) \quad (10b)$$

OCP (10) can be solved by setting up a nonlinear program with the ACADO Toolkit. Compared to the procedure described in Section III-B, the difference would be that the time interval T_d is fixed as $T_d = T/N_d$, instead of being determined by the numerical algorithm.

Moreover, T is a scalar, so one can define a method for solving (9) based on solving different instances of (10) for different values of T . In particular, in the general case in which ϕ_T^* has no particular properties, a practical solution consists of producing a sufficiently dense grid of values of T within a given interval $[0, T_{\max}]$, and then finding graphically the desired value of $T = T^*$, given the values of ϕ^* and δ (see (7), (9)). A graphical representation of the described method is depicted in the left side of Fig. 1: the solid red line represents the value of ϕ_T^* as a function of T , and the maximum value ϕ^* in the interval $[0, T_{\max}]$ is represented by the dashed horizontal

line. In this example, the value of T^* is determined graphically, in the simple case in which $\delta = 0$.

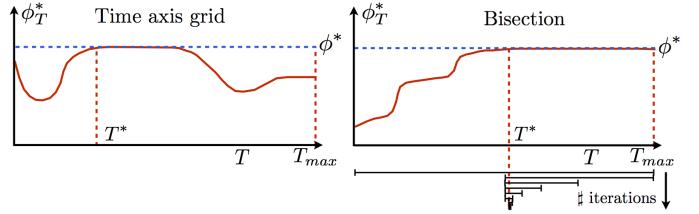


Fig. 1: Graphical representation of the time-axis-grid method (left) and of the bisection method (right).

C. Solution via bisection

Producing a grid of values of the T axis, in which each point is obtained by solving (10), is typically computationally expensive. An alternative way for solving (10) is hereafter proposed. The method is based on the result proven in the following proposition.

Proposition 1: Assume that, given $x(0)$, there exists $u_0 \in \mathbb{R}^{n_u}$ such that $f(x(0), u_0) = 0$. Then, recalling the definition of ϕ_T^* in (8) for a given $T \in \mathbb{R}_{\geq 0}$, for two fixed values T_1 and T_2 one has that

$$T_1 \leq T_2 \Rightarrow \phi_{T_1}^* \leq \phi_{T_2}^*, \quad (11)$$

i.e., ϕ_T^* is a monotonically non-decreasing function of T .

Proof: For a given $x(0)$ and time interval T_1 , let the corresponding maximizer be $u_{[0,T_1]}^* = u_1^*(t)$. We will refer to this maximization problem as “Problem 1”. By defining $\Delta T \triangleq T_2 - T_1 \geq 0$, a feasible control signal $\tilde{u}_2(t)$ in $[0, T_2]$ (i.e., for what we call “Problem 2”) can be obtained as $\tilde{u}_2(t) \equiv u_0$ for $t \in [0, \Delta T]$, and $\tilde{u}_2(t) = u_1^*(t - \Delta T)$, for $t \in [\Delta T, T_2]$.

It is immediate to verify that, since $u_1^*(t)$ corresponds to a feasible solution of the OCP (9) for Problem 1, so does $\tilde{u}_2(t)$ for Problem 2, as explained in the following. By assumption, $(x(0), u_0)$ is an equilibrium pair. Therefore, $x(\Delta T)$ obtained by applying $\tilde{u}_2(t) = u_0$ in $[0, \Delta T]$ is equal to $x(0)$. Hence, $\phi_{T_1}^* = \phi_{T_2}^*$, the latter being defined as $\phi_{T_2}^* \triangleq \phi(x(T_2), \tilde{u}_2(T_2))$, obtained by applying $\tilde{u}_2(t)$ to Problem 2. The corresponding optimal solution will be satisfying $\phi_{T_2}^* \geq \phi_{T_1}^*$ by definition, which concludes the proof. ■

This result shows that, with a larger time interval at our disposal, the achievable performance will not degrade. The assumption requiring $f(x(0), u_0) = 0$ indicates that the robot is at rest at the initial time instant. The need for this assumption can be explained by the following consideration: in the example of the robot arm hitting the cart, if the arm is already in motion at the initial time instant, it might be more convenient to hit the cart immediately, because, due to energy dissipation, the force with which it is hit might decrease if more swings of the robotic arm are performed. Therefore, the distance reached by the cart will not necessarily be a non-decreasing function of T .

The right side of Fig. 1 shows an example of ϕ_T^* (solid red line) as a monotonically non-decreasing function of T . Also in

this case, the value of ϕ^* is indicated by the dashed horizontal line. The monotonicity result in Proposition 1 allows us to use a bisection routine to solve (10). Fig. 1 also gives a graphical representation of the sequence of intervals resulting from the bisection algorithm, which is explained in detail in Algorithm 1. In this algorithm, in addition to δ , the sampling interval T_s of the closed-loop implementation is used as another tolerance value. The value of T_s is used to formulate the termination criterion, and the optimal time value T^* is finally found as $T^* = \text{ceil}(T_l, T_s)$, defined as the smallest integer multiple of T_s which is greater or equal than T_l , this latter being the lower bound of the interval determined at the end of the bisection interval.

Remark 1: The result provided in Proposition 1 would be valid for any nonlinear dynamical system, including other types of mechatronic systems other than VSA robots. However, its importance in this context is given by the fact that using OCPs for formulating tasks is a common procedure when working with VSA robots. Furthermore, in order to find ϕ_T^* in (8) for a VSA robot, a numerical solution of a medium-sized optimization problem has to be found, which requires a non-negligible computation time. Thus, the possibility of using the proposed bisection algorithm rather than proceeding as described in Section IV-B would save computation time.

Algorithm 1: Bisection Routine for OCP (9)

Input: T_{max} , OCP (10), constants T_s , $\delta \in \mathbb{R}_{\geq 0}$
Output: Optimal time T^* , optimal control signal $u_{[0,T^*]}^*$

```

 $T_l \leftarrow 0; /* initial lower limit */$ 
 $\phi_l^* \leftarrow \text{solution of OCP (10) for } T = T_l;$ 
 $T_u \leftarrow T_{max}; /* initial upper limit */$ 
 $\phi_u^* \leftarrow \phi_{max}^* \leftarrow \text{solution of OCP (10) for } T = T_u;$ 
while  $T_u - T_l > T_s$  do
     $T_m \leftarrow (T_l + T_u)/2;$ 
     $\phi_m^* \leftarrow \text{solution of OCP (10) for } T = T_m;$ 
    if  $\phi_m^* \leq \phi_{max}^* - \delta$  then
         $T_l \leftarrow T_m$ 
    else
         $T_u \leftarrow T_m$ 
     $T^* \leftarrow \text{ceil}(T_l, T_s);$ 
     $u_{[0,T^*]}^* \leftarrow \text{optimizer of OCP (10) for } T = T^*;$ 
return  $T^*, u_{[0,T^*]}^*$ .
```

V. FEEDBACK CONTROL SCHEME

The ideal scheme considered during task planning is reported in the upper part of Fig. 2. After the optimal input signal is calculated either solving OCP (6) or (9), the corresponding reference motor positions θ_d and torques τ_d are provided to the system, in order to execute the task. However, this would lead to a system evolution different from the expected one, due to the unavoidable presence of external disturbances and model mismatches. Therefore, the task is implemented by executing the scheme in the bottom part of Fig. 2. The ideal motor references, together with the corresponding evolution of the state variables, are resampled with sampling time T_s (which

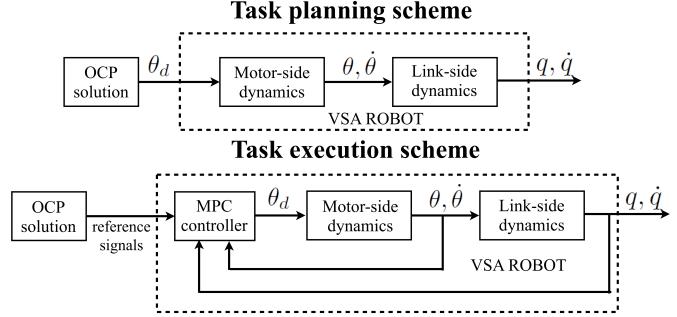


Fig. 2: The task planning scheme (upper part), and its closed-loop implementation (lower part).

can differ from the discretization step T_d), and provided as reference to a closed-loop controller. This is an NMPC controller, which reads state variables from sensors at every sampling interval T_s , and determines the inputs in real time, by solving a finite-horizon optimal control problem online. This latter enforces constraint satisfaction, at the same time tracking the ideal evolution of the system states. In this way, the actual execution of the task is closer to the ideal evolution than directly applying the computed optimal input signal in open loop. In-depth formulation and implementation of the NMPC controller is presented in [10].

The problem of finding the NMPC control variables is formulated by solving a numerical optimization problem online. Its formulation is analogous to that of the previously-described OCPs, but is tailored for real-time execution, limiting a-priori the number of SQP iterations. For the details of the mentioned numerical methods, which are also briefly recalled in [10], the reader is referred to [26] and the references therein.

VI. EXPERIMENTS

The two approaches for minimum-time trajectory planning described in Sections III and IV are tested in two case studies described in the remainder of the paper.

A. Case Study 1: Two-link planar VSA manipulator

1) *Experimental setup:* The system considered for the case study is a planar two-link manipulator, the schematic drawing of which is shown in Fig. 3. A ball is attached to the end effector through an electromagnet, and the task consists of throwing the ball at a given distance in minimum time. The same type of robotic system was also used to demonstrate the closed-loop control scheme presented in [10]. The main modeling aspects are here recalled for the sake of completeness, and a picture of the manipulator is shown in Fig. 4. For further details of the experimental setup, the reader is referred to [10].

The system is actuated by four servomotors Dynamixel MX-28T, connected to four nonlinear elastic elements (NEEs). As for the motor-side dynamics, the parameters κ_i in (1) have been found by system identification to be equal for all four motors, and precisely $\kappa_i = 40.0$. The vector of link-side angles describing the orientation of the two links (see (2)) is defined as $q = [q_1 \ q_2]' \in \mathbb{R}^2$. Again with reference to (2), the inertia

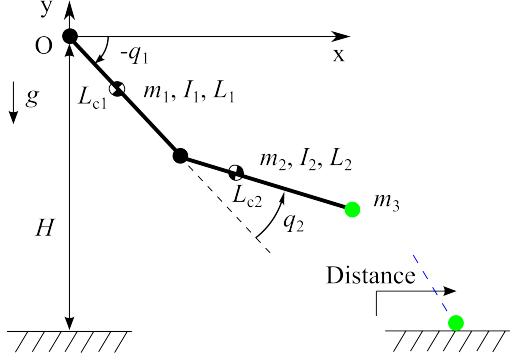


Fig. 3: Schematic drawing of the two-link planar VSA manipulator with a ball for Case Study 1.

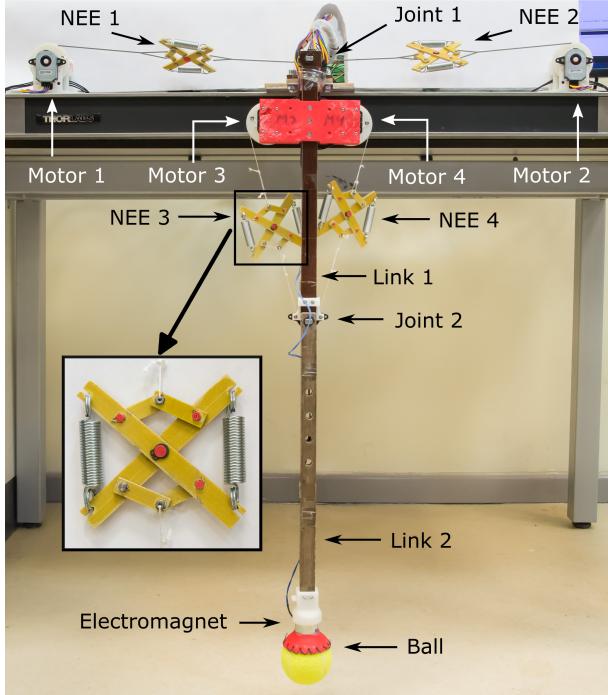


Fig. 4: Experimental setup with labeled elements for Case Study 1.

matrix $M(q)$ is defined as

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \quad (12)$$

where $m_{11} = I_1 + m_1 L_{c1}^2 + I_2 + m_2 (L_1^2 + 2L_1 L_{c2} \cos q_2 + L_{c2}^2) + m_3 (L_1^2 + 2L_1 L_2 \cos q_2 + L_2^2)$, $m_{12} = I_2 + m_2 (L_{c2}^2 + L_1 L_{c2} \cos q_2) + m_3 (L_2^2 + L_1 L_2 \cos q_2)$, $m_{22} = I_2 + m_2 L_{c2}^2 + m_3 L_2^2$, in which m_i , I_i , L_i , L_{ci} are, respectively, the mass, the moment of inertia about the center of mass, the length and the distance from the rotation point of the i -th link, up to the center of mass of the same link ($i = 1, 2$). Their values have been obtained as described in [10], and are equal to: $m_1 = 0.674$ kg, $m_2 = 0.307$ kg, $I_1 = 5.08 \cdot 10^{-3}$ kg·m², $I_2 = 6.92 \cdot 10^{-3}$ kg·m², $L_1 = 0.330$ m, $L_2 = 0.433$ m, $L_{c1} = 0.116$ m, $L_{c2} = 0.222$ m. The mass of the ball is

$m_3 = 0.074$ kg. Other terms in (2) for the case study are

$$C(q, \dot{q}) = -(m_2 L_{c2} + m_3 L_2) L_1 \sin q_2 \begin{bmatrix} 2\dot{q}_2 & \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix},$$

and $D = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$, where $b_1 = 0.010$ N·m·s and $b_2 = 0.006$ N·m·s are viscous damping coefficients. The last term in (2) to be described for this case study is $G(q) = [g_1 \ g_2]'$, in which $g_1 \triangleq g(m_1 L_{c1} + m_2 L_1 + m_3 L_1) \cos q_1 + g(m_2 L_{c2} + m_3 L_2) \cos(q_1 + q_2)$, and $g_2 \triangleq g(m_2 L_{c2} + m_3 L_2) \cos(q_1 + q_2)$.

In order to achieve variable-stiffness behavior, an antagonistic configuration of the NEEs is implemented in both joints (implying that two NEEs are present in each joint). In particular, NEE 1 and NEE 2 belong to the first joint, and NEE 3 and NEE 4 belong to the second joint. Vector τ_E , containing the elastic joint torques, is expressed as

$$\tau_E = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \rho \begin{bmatrix} (T_1 - T_2) \\ (T_4 - T_3) \end{bmatrix} \quad (13)$$

with $T_i, i = 1, \dots, 4$ being the tensions in each of the four tendons, and $\rho = 0.013$ m being the radius of both joints. The tendon tensions are quadratic polynomial functions of tendon displacements, according to the expression $T_i = \alpha_i \delta_i^2 + \beta_i \delta_i, i = 1, \dots, 4$, δ_i being the tendon displacement, and α_i, β_i being design coefficients (precisely, $\alpha_1 = 12400$ N/m², $\beta_1 = 1360$ N/m, $\alpha_2 = 13600$ N/m², $\beta_2 = 1350$ N/m, $\alpha_3 = 5320$ N/m², $\beta_3 = 1500$ N/m, $\alpha_4 = 13700$ N/m², $\beta_4 = 1410$ N/m). The value of δ_i for each tendon is computed as $\delta_1 = \delta_0 - \rho(q_1 + \frac{\pi}{2}) + \rho_p(\theta_1 - \theta_1^0)$, $\delta_2 = \delta_0 + \rho(q_1 + \frac{\pi}{2}) - \rho_p(\theta_2 - \theta_2^0)$, $\delta_3 = \delta_0 + \rho q_2 + \rho_p(\theta_3 - \theta_3^0)$, $\delta_4 = \delta_0 - \rho q_2 - \rho_p(\theta_4 - \theta_4^0)$, where $\rho_p = 0.026$ m and $\delta_0 = 0.005$ m are the radius of the pulleys on all motors and the initial displacement of the NEEs, respectively. The values of $\theta_i^0, i = 1, \dots, 4$ represent the initial servomotor positions corresponding to the initial NEE displacement.

The stiffness of NEEs and the elastic energy stored in them can be found accordingly as $\sigma_i = 2\alpha_i \delta_i + \beta_i, i = 1, \dots, 4$, and $E_i = \alpha_i \delta_i^3 / 3 + \beta_i \delta_i^2 / 2, i = 1, \dots, 4$. The stiffness of the joints can be found from (13), as

$$\sigma_E = \begin{bmatrix} \sigma_{E,1} \\ \sigma_{E,2} \end{bmatrix} = \rho^2 \begin{bmatrix} (2\alpha_1 \delta_1 + \beta_1 + 2\alpha_2 \delta_2 + \beta_2) \\ (2\alpha_4 \delta_4 + \beta_4 + 2\alpha_3 \delta_3 + \beta_3) \end{bmatrix} \quad (14)$$

For a different VSA system the equation connecting link-side and actuator-side dynamics will be different than (13), but the general procedure for the OCP solution will not change. For more information about the design of these types of NEE and their characteristics, the reader is referred to [10] and [29]. The dynamics of the whole system can be expressed as in (5), with 12 states and 4 controlled inputs. In particular, we can define $x = [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2 \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \dot{\theta}_4]'$ and $u = [\theta_{d,1} \ \theta_{d,2} \ \theta_{d,3} \ \theta_{d,4}]'$.

The objective of the OCP in form (6) is to minimize the time interval from the initial time instant until the ball touches the ground. Therefore, T in (6a) represents the time interval up to the release moment, while the flight time for the ball, only dependent on the state variables at time T , is

$$\mathfrak{D}(x(T)) = \frac{1}{g} \left(\dot{y}_r + \sqrt{\dot{y}_r^2 + 2g(y_r + H)} \right) \quad (15)$$

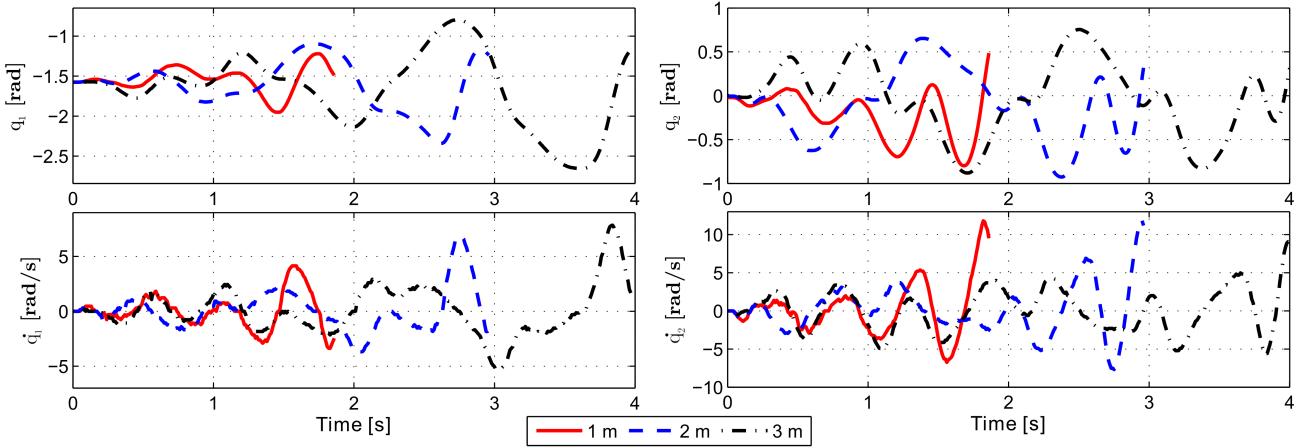


Fig. 5: Plot of experimental link angular positions and velocities for three cases: $\Phi = 1 \text{ m}$, $\Phi = 2 \text{ m}$, $\Phi = 3 \text{ m}$.

where $H = 0.810 \text{ m}$ is the height from the ground to the origin of the reference frame, $g = 9.81 \text{ m/s}^2$, $y_r = L_1 \sin(q_1) + L_2 \sin(q_1 + q_2)$ is the release vertical position, and $\dot{y}_r = L_1 \cos(q_1) \dot{q}_1 + L_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2)$ is the release vertical velocity, in which all variables are evaluated at time T . In (6b), the initial condition $x(0)$ is defined by setting vertical hanging position of the two links ($q_1 = -\frac{\pi}{2}$, $q_2 = 0$), zero link velocities ($\dot{q}_1 = \dot{q}_2 = 0$), zero motor velocities ($\dot{\theta}_i = 0$, $i = 1, \dots, 4$), and initial motor positions corresponding to initial NEE displacements. The constraint set \mathcal{Z} in (6c) is defined directly referring to physical quantities, as the set of states x and inputs u , such that the following inequalities hold:

$$-170\pi/180 \leq q_1 \leq -10\pi/180, [\text{rad}] \quad (16a)$$

$$-90\pi/180 \leq q_2 \leq 90\pi/180, [\text{rad}] \quad (16b)$$

$$-1.8 \leq \dot{\theta}_i \leq 1.8, [\text{rad/s}] \quad (16c)$$

$$0 \leq \theta_{d,i} \leq 2\pi, [\text{rad}] \quad (16d)$$

$$0 \leq \mathcal{T}_i \leq 0.9, [\text{Nm}] \quad (16e)$$

$$0.005 \leq \delta_i \leq 0.025, [\text{m}] \quad (16f)$$

where $\mathcal{T}_i = T_i \rho_p$, $i = 1, \dots, 4$, are the motor torques. The inequalities in (16) impose limits on states (link positions and motor velocities) in (16a)-(16c), on controls (desired motor positions) in (16d) and on variables (motor torques and spring compressions), that are functions of states and control inputs, in (16e)-(16f). Notice that all inequalities in (16) are functions of the components of x and u .

The performance index in (6d) is the distance reached by the ball, which is expressed as a function of the state vector x at the release time T , as

$$\phi(x(T)) = x_r + \dot{x}_r \frac{\dot{y}_r + \sqrt{\dot{y}_r^2 + 2g(y_r + H)}}{g}, \quad (17)$$

where $x_r = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2)$, $\dot{x}_r = -L_1 \sin(q_1) \dot{q}_1 - L_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2)$, evaluated at time T . The set Φ is defined in this case study as a singleton. Specifically, three different OCPs in form (6) are considered, with three different values $\Phi = 1 \text{ m}$, $\Phi = 2 \text{ m}$, $\Phi = 3 \text{ m}$, each of them enforcing a different throwing distance.

	Target Performance		
	$\Phi = 1 \text{ m}$	$\Phi = 2 \text{ m}$	$\Phi = 3 \text{ m}$
T	1.86 s	2.96 s	4.00 s
$\mathfrak{D}(x(T))$	0.34 s	0.56 s	0.68 s
$T + \mathfrak{D}(x(T))$	2.20 s	3.52 s	4.68 s

TABLE I: Minimum values of $T + \mathfrak{D}(x(T))$ and their components (robot motion time T and ball flight time $\mathfrak{D}(x(T))$) in the case of minimum time for target performance.

In order to solve the three OCPs in form (6), we used the ACADO Toolkit (Version 1.2.0 beta) set up as described in Section III-B, setting the so-called *KKT tolerance* (which defines the SQP termination criterion) to 10^{-5} , and the number of nodes $N_d = 200$. The computation time of the OCP on a desktop computer with 3.2 GHz Intel Core i5-3470 processor and 16 GB of memory was (for each of the three problems) approximately 50 minutes, which is an acceptable computation time for offline trajectory planning. The fact that the time to solve the OCP for each of the three cases was approximatively the same is due to the same value of N_d used in each of the three cases, which made their complexity comparable.

2) *Experimental results:* In order to experimentally test our method, an NMPC controller was implemented as described in Section V, and with more detail in [10, Sections V and VI.D]. The NMPC controller acted with a sampling interval $T_s = 20 \text{ ms}$, reading motor and link positions from high-resolution capacitive incremental encoders, and generated the motor reference positions θ_d in real time, in order to track the robot motion planned by the OCP (see Fig. 2). For the closed-loop experiments, the system constraints were slightly loosened to better compensate for the effect of possible external disturbances and model uncertainties, as explained in detail in [10].

As a result of the three OCPs, the minimum-time intervals $T + \mathfrak{D}(x(T))$ were obtained equal to 2.20 s for $\Phi = 1 \text{ m}$, 3.52 s for $\Phi = 2 \text{ m}$, and 4.68 s for $\Phi = 3 \text{ m}$. Table I presents rounded results of OCP solutions for three different Φ values for the case of minimization of the sum of robot motion time T and ball flight time $\mathfrak{D}(x(T))$. As a side note, we would like to remark that, when large values of Φ are

imposed, such as $\Phi = 4$ m, the OCP becomes infeasible, because it is impossible to reach such a distance without violating the imposed constraints. The experimental results, making use of the NMPC controller, led to actual distances equal to approximately 1.0 m, 1.9 m, and 2.8 m, respectively.

Fig. 5 shows the experimental time evolution of link angular positions and velocities for the three considered experiments. In addition, a video showing the three closed-loop experiments is available in the supplementary material. The difference between the real and the ideal distance, at least in the second and third case, is due to parameter uncertainties and model inaccuracies, such as hysteresis in the NEEs behavior and unaccounted frictional forces. The variations of joint stiffnesses, and of energies and stiffnesses of NEEs are shown in Fig. 6. Joint stiffnesses vary from 0.45 Nm/rad up to 0.6 Nm/rad, while NEE stiffnesses vary from 1200 N/m up to 2000 N/m. It can be observed that at the beginning of link motions, joint stiffnesses are set to low (loose) values, which is necessary to store energy in the NEEs. Close to the ball release moment, joint stiffnesses are set to high (stiff) values, which is beneficial for increasing the ball flight distance. The same behavior of stiffness variation of joints can be observed when humans throw objects with their arms. At the beginning of the swinging motion, elbow and shoulder muscles are loose, while close to the release moment these muscles are tightened.

The imprecision in task execution is strongly reduced, if compared with the application of the reference motor position obtained from the OCP in open loop, directly employing the task planning scheme for execution (cf. [9]). The corresponding experimental distances by directly applying the task planning scheme are indeed equal to 0.8 m for $\Phi = 1$ m, 1.4 m for $\Phi = 2$ m, and 1.8 m for $\Phi = 3$ m, respectively.

The minimum time for maximum performance problem was also solved for this experimental setup, minimizing the release time T . OCP (10) has been solved for T ranging from 0 to 7 s, with a grid resolution of 0.2 s in order to obtain the dependence of ϕ_T^* as a function of T . Fig. 9a shows the results of applying the bisection algorithm to the two-link manipulator, with parameters set to $T_{max} = 7$ s and $\delta = 0.05$ m. The green dot represents the final optimal point, while the red dots represent the iteration steps of the bisection algorithm. For the sake of clarity, bisection iteration points numbered 8 and 9 are not shown. The total computation time to obtain the results in the grid is 7720 minutes, while the time spent on bisection algorithm is 2273 minutes, which is 3.4 times faster. Table II summarizes the results of the bisection algorithm. The maximum obtainable throwing distance is $\phi_{max}^* = 3.72$ m (obtained for $T = 7$ s), and the optimal time value $T^* = 5.12$ s was obtained, corresponding to a throwing distance $\phi = 3.67$ m.

We also conducted simulation experiments to show the advantages of VSA systems compared to rigid ones. For this, the minimum time for maximum performance problem was solved, via time-grid method, for a modified version of the two-link manipulator. The only difference was the assumption that the actuators are rigidly connected to the links at the joints (one actuator per joint), with torque limits set (in the OCP constraints) to twice the Dynamixel motor torque limit

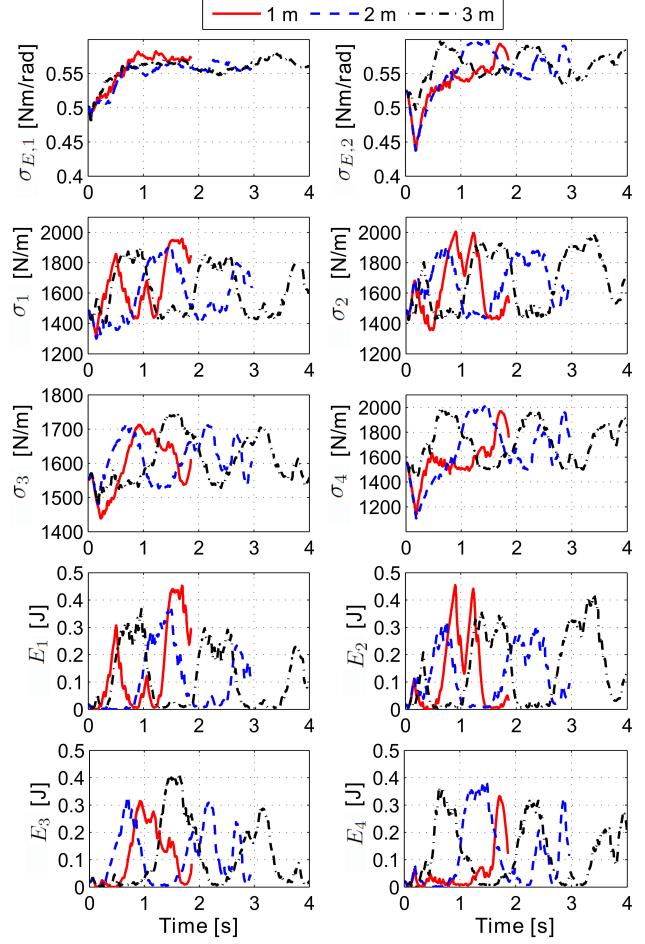


Fig. 6: Stiffness variations of joints and of stiffness and energy variations of NEEs for $\Phi = 1$ m, $\Phi = 2$ m, $\Phi = 3$ m.

Iteration	T_l [s]	T_u [s]	ϕ_l^* [m]	ϕ_u^* [m]
1	0.00	7.00	0.00	3.72
2	3.50	7.00	2.63	3.72
3	3.50	5.26	2.63	3.70
4	4.38	5.26	3.41	3.70
5	4.82	5.26	3.59	3.70
6	5.04	5.26	3.65	3.70
7	5.04	5.16	3.65	3.69
8	5.10	5.16	3.66	3.69
9	5.10	5.14	3.66	3.69
10	5.12	5.14	3.67	3.69

TABLE II: Bisection results for Case Study 1. Bold numbers denote the outputs of the current iteration.

in (16e). In the time grid, T was ranging from 0 s to 7 s with 0.2 s intervals. The maximum distance was obtained at $T^* = 0.60$ s as $\phi = 1.2$ m, while the two-link VSA system had achieved, at $T^* = 5.12$ s, the maximum distance of $\phi = 3.67$ m. We speculate that the VSA system is advantaged because the maximum joint velocity is not limited by the maximum actuator velocity thanks to the decoupling of actuators and links via NEEs.

B. Case Study 2: Reaction wheel augmented VSA system

1) *Experimental setup:* We employed another experimental setup for solving the minimum time for maximum perfor-

mance problem, in order to demonstrate the applicability of our methodologies for different types of VSA robots. The experimental setup is also a planar manipulator, but only one link is present, together with a reaction wheel. The latter generates reactive torques and thus serves as an additional actuation degree-of-freedom, enriching the dynamics of the system. The schematic drawing of this robot is shown in Fig. 7, while the corresponding experimental setup is shown in Fig. 8. The system consists of one link, one joint, two servomotors, a reaction wheel, an electromagnet and a ball. The task is to throw the ball to the maximum possible distance in minimum time. Two Dynamixel MX-28T servomotors drive the link motion as in the previous case study, and the reaction wheel is driven by a Maxon EC-45 brushless motor. For a detailed description of the experimental setup, see [27].

Similarly to Section VI-A, a vector $q = [q_1 \ q_2]' \in \mathbb{R}^2$ is defined: q_1 describes the angular position of the only link, and q_2 accounts for the angular position of the reaction wheel. With reference to (2), the inertia matrix is defined with the same form of equation (12), but now with different terms $m_{11} = I_1 + m_1 L_{c1}^2 + I_2 + m_2 L_{c2}^2 + m_3 L_1^2$, $m_{12} = m_{22} = I_2$, in which m_i , I_i , L_{ci} are, respectively, the mass, the moment of inertia about the center of mass and the distance from the rotation point of the link, up to the center of mass of link ($i = 1$) and wheel ($i = 2$), respectively, while L_1 is the length of the link. The system parameters for this case study are the following: $m_1 = 0.559$ kg, $m_2 = 0.277$ kg, $I_1 = 9.64 \cdot 10^{-3}$ kg·m², $I_2 = 0.35 \cdot 10^{-3}$ kg·m², $L_1 = 0.433$ m, $L_{c1} = 0.166$ m, $L_{c2} = 0.100$ m. Due to the assumption that the mass of the reaction wheel is located on its axis of rotation, there are no Coriolis forces present, i.e., $C(q, \dot{q}) = [0 \ 0]'$.

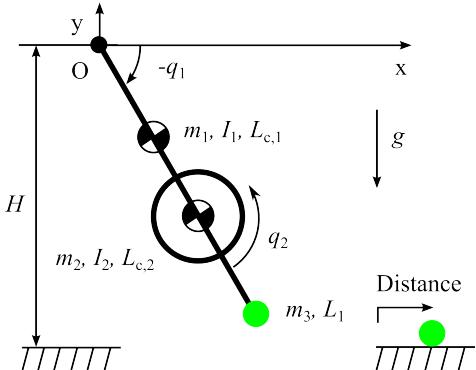


Fig. 7: Schematic drawing of the single link planar robot manipulator with reaction wheel for Case Study 2.

The other terms in (2) are defined as $D = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$, where $b_1 = 0.006$ N·m·s and $b_2 = 1.55 \cdot 10^{-4}$ N·m·s are the damping coefficients of the link and the reaction wheel, respectively, and $G(q) = [g_1 \ g_2]'$, in which $g_1 \triangleq g(m_1 L_{c1} + m_2 L_{c2} + m_3 L_1) \cos q_1$ and $g_2 \triangleq$.

The variable stiffness behavior is achieved by the two NEEs (NEE1 and NEE2), connected to the first link and described in Section VI-A1. The relations between tendon tension, spring displacement, link positions and motor positions, and initial conditions, are given by the same equations reported in Section

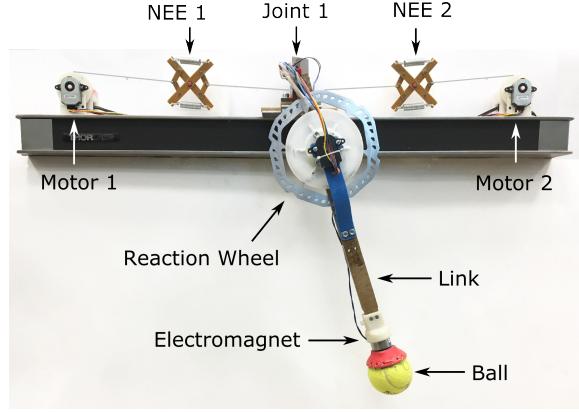


Fig. 8: Experimental setup with labeled elements for Case Study 2.

VI-A1 for the first link of the robot in Case Study 1. Also stiffness variation of the NEEs and of the joint are given by the same equations as for the first joint in VI-A1. The vector of elastic torques for Case Study 2 is written as

$$\tau_E = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \rho(T_1 - T_2) \\ \tau_d \end{bmatrix} \quad (18)$$

where $\tau_d = k^w I^w$ is the (scalar) input torque of the brushless motor driving the reaction wheel, while $k^w = 0.131$ N·m/A and I^w are the torque constant and current of the motor driving the reaction wheel. Overall, the system dynamics can be written as in (5), with 7 states and 3 control inputs as $x = [q_1 \dot{q}_1 \dot{q}_2 \theta_1 \theta_2 \dot{\theta}_1 \dot{\theta}_2]'$ and $u = [\theta_{d,1} \theta_{d,2} \tau_d]'$. Notice that the description of the state vector x defined above has a form slightly different from the general one reported in (4), since the angular position q_2 should also be present in it. However, q_2 is not necessary for describing the system dynamics in the considered task, and has therefore been omitted.

OCP (9) is defined for Case Study 2 as the problem of achieving the maximum throwing distance $\phi(x(T))$ while minimizing the time interval until the release time T . The throwing distance has the same form reported in (17), but the terms therein are defined for this case study as $x_r = L_1 \cos(q_1)$, $y_r = L_1 \sin(q_1)$, $\dot{x}_r = -L_1 \sin(q_1)\dot{q}_1$, $\dot{y}_r = L_1 \cos(q_1)\dot{q}_1$, again evaluated at time T . In (9c), the initial condition $x(0)$ is defined by setting vertical hanging position of the link ($q_1 = -\frac{\pi}{2}$), zero link and reaction wheel velocities ($\dot{q}_1 = \dot{q}_2 = 0$), zero servomotor velocities ($\dot{\theta}_i = 0$, $i = 1, 2$), and initial motor positions corresponding to initial displacements of two NEEs. The constraint set \mathcal{Z} in (9c) is defined based on the physical limitations of the system depending on states x and inputs u :

$$-225\pi/180 \leq q_1 \leq 45\pi/180, [\text{rad}] \quad (19a)$$

$$-200 \leq q_2 \leq 200, [\text{rad/s}] \quad (19b)$$

$$-2.0 \leq \dot{\theta}_i \leq 2.0, [\text{rad/s}] \quad (19c)$$

$$-2\pi \leq \theta_{d,i} \leq 3\pi, [\text{rad}] \quad (19d)$$

$$-2.29 \leq I^w \leq 2.29, [\text{A}] \quad (19e)$$

$$0 \leq \mathcal{T}_i \leq 0.9, [\text{Nm}] \quad (19f)$$

$$0.003 \leq \delta_i \leq 0.025, [\text{m}] \quad (19g)$$

where $i = 1, 2$. Similarly to the previous case study, the inequalities in (19) impose limits on states (link position, reaction wheel and motor velocities) in (19a)-(19c), on controls (desired servo motor positions and input current for the brushless motor) in (19d)-(19e) and, in (19f)-(19g), on variables (motor torques and spring compressions) that can be expressed as functions of states and control inputs.

The OCP was formulated using the ACADO Toolkit in the same general way as described in subsection VI-A1. One can verify that, for the considered case study, the assumptions required for the satisfaction of Proposition 1 are met. More precisely, the initial condition corresponds to an equilibrium point (the robot is in rest position). As a consequence, the bisection method described in Algorithm 1 was used, by experimentally choosing $T_{max} = 3$ s and $\delta = 0.05$ m.

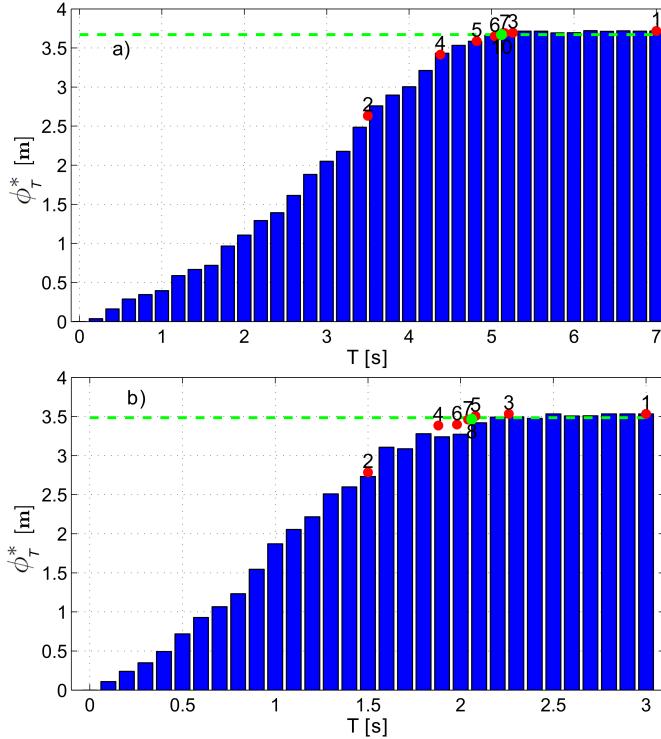


Fig. 9: OCP solution for the ball throwing distance d , for each corresponding execution time T . The red dots represent the steps of the bisection algorithm, with their corresponding iteration number, while the optimal point is shown as a green dot. The green line denotes the $\phi_{max}^* - \delta$ boundary. Results for: a) double-link manipulator (bisection algorithm results for points 8 and 9 in this case are not shown for clarity), and b) single-link manipulator with the reaction wheel.

2) *Experimental results:* In order to provide an overview of ϕ_T^* as a function of T , the OCP was solved for T ranging from 0 to 3 s, with a grid resolution of 0.1 s, (see Fig. 9b). The bisection algorithm, whose steps are also shown as red dots in Fig. 9b, was also used to find the optimal point. It took 8 iterations for the bisection algorithm to converge to the optimal point (shown as a green dot). Table III summarizes the execution of the bisection algorithm. As a result, with the maximum obtainable throwing distance equal

Iteration	T_l [s]	T_u [s]	ϕ_l^* [m]	ϕ_u^* [m]
1	0.00	3.00	0.00	3.52
2	1.50	3.00	2.79	3.52
3	1.50	2.26	2.79	3.52
4	1.88	2.26	3.38	3.52
5	1.88	2.08	3.38	3.51
6	1.98	2.08	3.40	3.51
7	2.04	2.08	3.46	3.51
8	2.06	2.08	3.47	3.51

TABLE III: Bisection results for Case Study 2. Bold numbers denote the outputs of the current iteration.

to $\phi_{max}^* = 3.52$ m (obtained for $T = 3$ s), an optimal time value $T^* = 2.06$ s was obtained, corresponding to a throwing distance $\phi = 3.47$ m. The total time spent to solve all the OCP problems in the grid is 161 minutes, while the total time spent to solve the eight OCP problems in the bisection algorithm is 52 minutes, which is about 3.1 times faster. Comparing these results with the two-link manipulator results shows that the more complicated is the system, the more time is saved by applying the bisection algorithm. Notice that, even if ϕ_T^* is in theory a monotonically increasing function of T , for values of $T \in [2, 3]$, ϕ_T^* presents small positive and negative variations, due to numerical inaccuracies in solving the OCP: this proves the usefulness of introducing the use of δ .

An NMPC controller was implemented with sampling time $T_s = 20$ ms. The reaction wheel has faster dynamics than the links, therefore some adjustments to the implementation of the NMPC controller were made, as described in [27]. The experimental value of the throwing distance was 3.42 m, which is very close to the expected value of 3.47 m. Angular position and velocity of the link, angular velocity of the reaction wheel and reaction wheel input torque for this experiment are shown in Fig. 10. Finally, a video of this experiment is provided as a supplemental material.

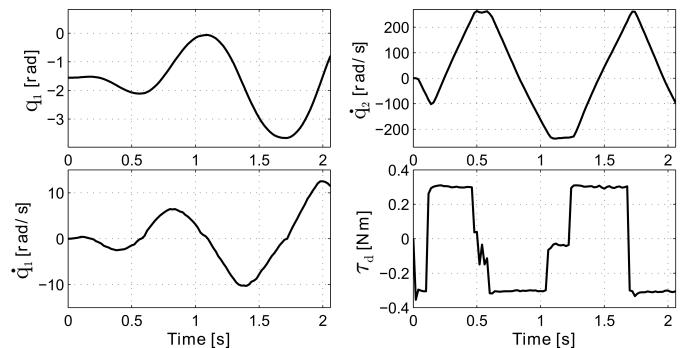


Fig. 10: Plot of experimental link angular position (q_1) and velocity (\dot{q}_1), of reaction wheel angular velocity (q_2) and input torque (τ_d) for Case Study 2.

VII. CONCLUSIONS

We presented a general framework for defining reference trajectories leading to time-optimal task execution, for VSA robots of any topology. These methods were tested experimentally on a ball throwing task on two VSA system prototypes, by following the generated trajectories using an NMPC control

scheme. The results, both in terms of computation time and performance, are encouraging, and can be considered as a further step to use VSA robots reliably in a wider set of scenarios. Even if an antagonistic configuration of NEEs is used in the two cases considered above, our method can be applied to any configuration of the joint-actuator connection.

REFERENCES

- [1] E. Guizzo and E. Ackerman, "The hard lessons of DARPA's robotics challenge," *IEEE Spectrum*, vol. 52, no. 8, pp. 11–13, 2015.
- [2] A. Verl, A. Albu-Schaffer, O. Brock *et al.*, *Soft robotics: Transferring theory to application*. Springer-Verlag, 2015.
- [3] W. Sebastian, B. Thomas, C. Maxime *et al.*, *Soft Robotics with Variable Stiffness Actuators: Tough Robots for Soft Human Robot Interaction*. Berlin: Springer, 2015, pp. 231–254.
- [4] S. Pfeifer, A. Pagel, R. Riener *et al.*, "Actuator with angle-dependent elasticity for biomimetic transfemoral prostheses," *IEEE/ASME T. Mech.*, vol. 20, no. 3, pp. 1384–1394, 2015.
- [5] F. Petit, W. Friedl, H. Hppner *et al.*, "Analysis and synthesis of the bidirectional antagonistic variable stiffness mechanism," *IEEE/ASME T. Mech.*, vol. 20, no. 2, pp. 684–695, 2015.
- [6] G. Mathijssen, D. Lefebvre, and B. Vanderborght, "Variable recruitment of parallel elastic elements: Series-parallel elastic actuators (SPEA) with dephased mutilated gears," *IEEE/ASME T. Mech.*, vol. 20, no. 2, pp. 594–602, 2015.
- [7] A. Jafari, N. G. Tsagarakis, I. Sardellitti *et al.*, "A new actuator with adjustable stiffness based on a variable ratio lever mechanism," *IEEE/ASME T. Mech.*, vol. 19, no. 1, pp. 55–63, 2014.
- [8] S. S. Groothuis, G. Rusticelli, A. Zucchelli *et al.*, "The variable stiffness actuator vsaUT-II: Mechanical design, modeling, and identification," *IEEE/ASME T. Mech.*, vol. 19, no. 2, pp. 589–597, 2014.
- [9] D. J. Braun, F. Petit, F. Huber *et al.*, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE T. Robotics*, vol. 29, no. 5, pp. 1085–1101, 2013.
- [10] A. Zhakatayev, M. Rubagotti, and H. A. Varol, "Closed-loop control of variable stiffness actuated robots via nonlinear model predictive control," *IEEE Access*, vol. 3, pp. 235–248, 2015.
- [11] ———, "Integrated optimal design and control of variable stiffness actuated robots," in *Proc. European Contr. Conf.*, 2015, pp. 1100–1105.
- [12] F. Petit, A. Daasch, and A. Albu-Schaffer, "Backstepping control of variable stiffness robots," *IEEE T. Contr. Syst. Tech.*, vol. 23, no. 6, pp. 2195–2202, 2015.
- [13] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, 1985.
- [14] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE T. Autom. Contr.*, vol. 30, no. 6, pp. 531–541, 1985.
- [15] S. Dubowsky and T. D. Blubaugh, "Planning time-optimal robotic manipulator motions and work places for point-to-point tasks," *IEEE T. Robotics Autom.*, vol. 5, no. 3, pp. 377–381, 1989.
- [16] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.
- [17] M. Diehl, H. G. Bock, H. Diedam *et al.*, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics*. Springer, 2006, pp. 65–93.
- [18] D. Verscheure, B. Demeulemaere, J. Swevers *et al.*, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE T. Autom. Contr.*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [19] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *Int. J. Control.*, vol. 87, no. 6, pp. 1297–1311, 2014.
- [20] F. Bourbonnais, P. Bigras, and I. A. Bonev, "Minimum-time trajectory planning and control of a pick-and-place five-bar parallel robot," *IEEE/ASME T. Mech.*, vol. 20, no. 2, pp. 740–749, 2015.
- [21] A. Bicchi, G. Tonietti, M. Bavaro *et al.*, "Variable stiffness actuators for fast and safe motion control," in *Proc. Int. Symp. Robot. Res.*, 2005, pp. 527–536.
- [22] A. Bicchi and G. Tonietti, "Fast and "soft-arm" tactics," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 22–33, 2004.
- [23] G. Tonietti, R. Schiavi, and A. Bicchi, "Optimal mechanical/control design for safe and fast robotics," in *Experimental Robotics IX*. Springer, 2006, pp. 311–320.
- [24] S. Haddadin, K. Krieger, N. Mansfeld *et al.*, "On impact decoupling properties of elastic robots and time optimal velocity maximization on joint level," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Sys.*, 2012, pp. 5089–5096.
- [25] N. Mansfeld and S. Haddadin, "Reaching desired states time-optimally from equilibrium and vice versa for visco-elastic joint robots with limited elastic deflection," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Sys.*, 2014, pp. 3904–3911.
- [26] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit - An open-source framework for automatic control and dynamic optimization," *Opt. Contr. Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [27] A. Baimyshev, A. Zhakatayev, and H. A. Varol, "Augmenting variable stiffness actuation using reaction wheels," *IEEE Access*, vol. 4, pp. 4618–4628, 2016.
- [28] M. Grebenstein, A. Albu-Schaffer, T. Bahls *et al.*, "The DLR hand arm system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3175–3182.
- [29] S. A. Migliore, E. A. Brown, and S. P. DeWeerth, "Biologically inspired joint stiffness control," in *Proc. IEEE Int. Conf. Robot. Autom.*, April 2005, pp. 4508–4513.



Altay Zhakatayev (M'15) received the B.S. degree in aerospace engineering from Texas A&M University, College Station, TX, USA, in 2010, and the M.S. degree in mechanical engineering from University College London, London, UK, in 2012. Since 2012, he has worked at the Advanced Robotics and Mechatronics Systems (ARMS) Laboratory of Nazarbayev University, Astana, Kazakhstan as a research assistant. His research interests include optimal control and design of variable impedance actuated robots using numerical optimization.



Matteo Rubagotti (S'07-M'11) received the Ph.D. degree in electronics, computer science, and electrical engineering from the University of Pavia, Italy, in 2010. Since 2015 he has been a lecturer of control engineering at the University of Leicester, UK. Before that, he was a postdoc at the University of Trento, Italy, and at the IMT Institute for Advanced Studies, Lucca, Italy, and then an assistant professor of robotics and mechatronics at Nazarbayev University, Astana, Kazakhstan. His research interests include the theory of model predictive control and sliding mode control, and their applications to robotics and energy systems. He has published over 40 technical papers in international journals and conferences.



Huseyin Atakan Varol (M'09-SM'16) received the B.S. degree in mechatronics engineering from Sabanci University, Istanbul, Turkey, in 2005, and the M.S. and Ph.D. degrees both in electrical engineering from Vanderbilt University, Nashville, TN, USA, in 2007 and 2009, respectively. From August 2009 to August 2011, he was first a postdoctoral research associate and then a research assistant professor with the Center for Intelligent Mechatronics, Department of Mechanical Engineering, Vanderbilt University, Nashville, TN, USA. In 2011, he joined the faculty of Nazarbayev University, Astana, Kazakhstan, as an associate professor of robotics, where he currently chairs the department and directs the Advanced Robotics and Mechatronics Systems (ARMS) Laboratory. His research interests include biomechatronics, variable impedance actuation, machine learning, and embedded systems. He has published over 50 technical papers on related topics in international journals and conferences.