Optimal Control for Articulated Soft Robots

Saroj Prasad Chhatoi[1,2]   Michele Pierallini[1,2]   Franco Angelini[1,2]
Carlos Mastalli[3,4]   Manolo Garabini[1,2]

Abstract—Soft robots can execute tasks with safer interactions. However, control techniques that can effectively exploit the systems' capabilities are still missing. Differential dynamic programming (DDP) has emerged as a promising tool for achieving highly dynamic tasks. But most of the literature deals with applying DDP to articulated soft robots by using numerical differentiation, in addition to using pure feed-forward control to perform explosive tasks. Further, underactuated compliant robots are known to be difficult to control and the use of DDP-based algorithms to control them is not yet addressed. We propose an efficient DDP-based algorithm for trajectory optimization of articulated soft robots that can optimize the state trajectory, input torques, and stiffness profile. We provide an efficient method to compute the forward dynamics and the analytical derivatives of series elastic actuators (SEA)/variable stiffness actuators (VSA) and underactuated compliant robots. We present a state-feedback controller that uses locally optimal feedback policies obtained from DDP. We show through simulations and experiments that the use of feedback is crucial in improving the performance and stabilization properties of various tasks. We also show that the proposed method can be used to plan and control underactuated compliant robots, with varying degrees of underactuation effectively.

Index Terms—articulated soft robots, underactuated compliant robots, optimal and state-feedback control, feasibility-driven differential dynamic programming

I. INTRODUCTION

Across many sectors such as the healthcare industry, we require robots that can actively interact with humans in unstructured environments. To enable safe interactions and increase energy efficiency, we often include soft elements in the robot structure [1] [2]. For instance, in an articulated soft robot (ASR) the rigid actuators connect to the joints through passive elements with or without variable stiffness (Fig. 1). These types of robots aim to mimic the musculoskeletal structure in vertebrated animals [3], [4], which enables them to perform highly dynamic tasks efficiently [5], [6]. A series elastic actuator (SEA) has a linear spring between the actuator and the load [7]. Instead, a variable stiffness actuator (VSA)

[1]Centro di Ricerca "Enrico Piaggio", Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy
[2]Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy
[3]Institute of Sensors, Signals and Systems, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK
[4]National Robotarium, Edinburgh, UK
Corresponding Author: Saroj Prasad Chhatoi email: s.chhatoi@studenti.unipi.it

(a) variable stiffness actuation.   (b) series elastic actuation.
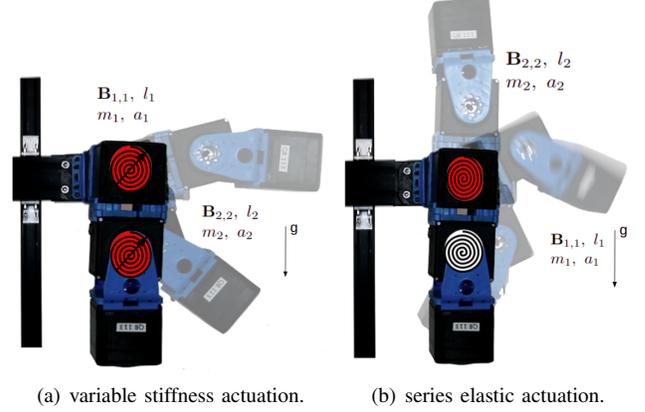
Fig. 1. Examples of articulated soft robots with joints that present torsional springs. The red joints are actuated, while the white one is unactuated. (a) Two degrees of freedom robot actuated by VSA performing a regulation task. (b) Two degrees of freedom robot with an actuated SEA joint, and an unactuated elastic joint.

integrates an elastic element that can be adjusted mechanically. These actuators provide many potential advantages but also increase the control complexity [8]. Similarly, compliant robots, a subclass of soft robots, are systems with rigid links and elastic joints (e.g., flexible joint robot SEA/VSA) in which a generic number of unactuated joints can also be present [9]. This mechanism further increases the modeling and control complexity. In addition, this class resembles other modeling formulations used in the soft robotics literature [1], [10]: Pseudo Rigid Body(PRB) model [11], [12], Cosserat Model [13], Constant Curvature models [14] and thus are an important class of models.

Applying controllers derived for rigid robots tends to provide an undesired performance in soft robots (see Sec. III-A). It may even have a detrimental effect, as it provides dynamically infeasible motions and controls. Therefore, we need to design control techniques that fully exploit the dynamic potential of soft robots. In this regard, optimal control solutions promise to be an effective tool.

Differential dynamic programming (DDP) is an optimal control method that offers fast computation and can be employed in systems with high degrees of freedom and multi-contact setups [15], [16].

In the context of soft robots, iterative LQR (iLQR) has been used to perform explosive tasks such as throwing a ball with maximum velocity by optimizing the stiffness and controls [17]. Similarly, DDP has enabled us plan time-energy optimal trajectories for systems with VSAs as well [18]. These works employ numerical differentiation to compute

the derivatives of the dynamics and cost functions. But such an approach is computationally expensive, and it is often prone to numerical errors. These works also completely rely on feed-forward control. Further, to the best of the authors' knowledge, the control of underactuated compliant systems has not been addressed using DDP algorithms. Furthermore, devising control laws for such systems is known to be difficult [19]. We propose an optimal control method for articulated soft robots in this work.

A. Contribution

In this paper, we propose an efficient optimal control method for articulated soft robots based on the feasibility-driven differential dynamic programming (FDDP)/Box-FDDP algorithm that can accomplish different tasks. It boils down to three technical contributions:

(i) an efficient approach to compute the forward dynamics and its analytical derivatives for robots with SEAs, VSAs and under-actuated compliant arms,

(ii) empirical evidence of the benefits of analytical derivatives in terms of convergence rate and computation time, and

(iii) a state-feedback controller that improves tracking performance in soft robots.

Our approach boosts computational performance and improves numerical accuracy compared to numerical differentiation. The state-feedback controller is validated in experimental trials on systems with varying degrees of freedom. We provide the code to be publicly accessible.[1]

The article is organized as follows: after discussing state of the art in optimal control for soft robots (Section II), we describe their dynamics and formulate their optimal control problem in Section III. Section IV begins by summarizing the DDP formalism and ends with the state-feedback controller. In Section V, we introduce various systems that we use for validating the proposed method. Finally, Section VI shows and discusses the efficiency of our method through a set of simulations and experimental trials.

II. RELATED WORK

Compliant elements introduce redundancies in the system that increases the complexity of the control problem. Optimal control is a promising tool to solve such kinds of problems. It can be classified into two major categories: 1) Direct, 2) Indirect methods. Indirect methods first optimize the controls using Pontryagin's Maximum Principle (PMP) and then discretize the problem. This approach has been used to compute optimal stiffness profiles while maximizing the terminal velocity as shown in [20], [21]. In [22], the authors use linear quadratic control of an Euler beam model and show its effectiveness w.r.t. PD/ state regulation method. But such methods have poor convergence under bad initialization and cannot handle systems with many degrees.

Instead, direct methods transcribe the differential equations into algebraic ones that are solved using general-purpose nonlinear optimizers. In [23], authors propose a time-optimal

control problem for soft robots, and it is solved using the direct method where the non-convexity of the problem is converted into bilinear constraints. Similarly, in [24], [25] direct methods are used to solve minimum time problems, and in [18], [26] direct methods are used to solve energy-optimal problems for soft robots. However, these methods often cannot be used in model predictive control settings as they are computationally slow.

Dynamic programming uses the Bellman principle of optimality to solve a sequence of smaller problems recursively. But this approach suffers from the curse of dimensionality and depends on input complexity. Rather than searching for global solutions, DDP finds a local solution [27]. These methods are computationally efficient but are highly sensitive to initialization, which limits their application to simple tasks. However, recent work proposes a feasibility-driven DDP (FDDP) algorithm improves the convergence under poor initialization [15] enabling us to compute motions subject to contact constraints. DDP-based approaches provide both feed-forward actions and feedback gains within the optimization horizon. Both elements enable our system to track the optimal policy, which increases performance as shown in [28]. The FDDP algorithm is efficiently implemented in the CROCODDYL library. Similarly, as described in [29], the Box-FDDP algorithm handles box constraints on the control variables and uses a feasibility-driven search. Both FDDP/Box-FDDP increases the basin of attraction to local minima and the convergence rate when compared to the DDP algorithm.

DDP and its variants have been used in the planning and control of robots with soft actuators. For instance, we can execute explosive tasks with VSAs using the iLQR algorithm [17]. Similarly, we can apply DDP to describe a hybrid formulation for robots with soft actuators [30]. Both works demonstrate the benefits of modeling their VSAs in highly dynamic tasks like jumping hopper and brachiation. But two major drawbacks of these approaches are their dependence on numerical differentiation, which increases computational time, and the lack of feedback terms, which decreases performance. To analytically compute the derivatives of rigid systems, [31] exploits the induced kinematic sparsity pattern in the kinematic tree. This method reduces the computation time obtained using other common techniques: automatic or numerical differentiation. It is possible to use the tools developed for the rigid body case and tailor them for applications related to systems with soft actuators. This will be beneficial for both the online deployment of the algorithms and the control of systems with high degrees of freedom. Secondly, in [28], the feedback policy obtained from DDP, is employed instead in place of a user- tuned tracking controller. The results show that the local feedback policy obtained from DDP could be a promising solution for state feedback.

III. PROBLEM DEFINITION

In this section, we formulate the optimal control problem to plan a desired task with an articulated soft robot with a fixed-base and without any contacts.

---

[1] github.com/spykspeigel/aslr_to
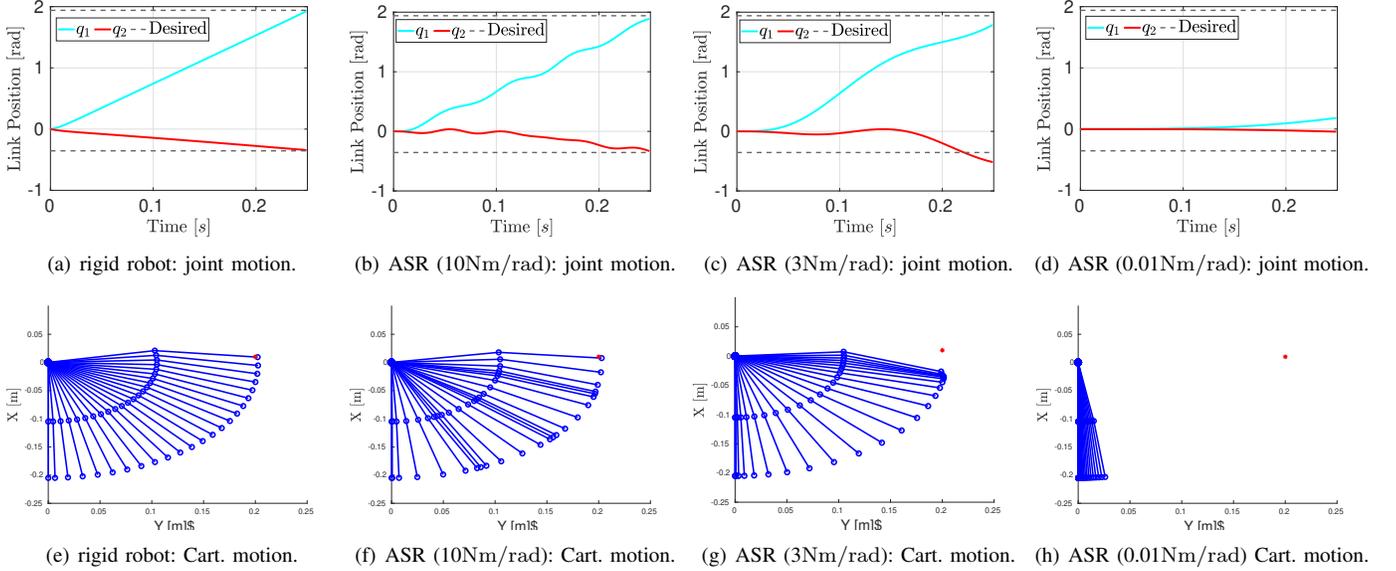
Fig. 2. Motivational example: a 2DoF robot affected by gravity performing a regulation task with the desired final end-effector position equal to [0.01, 0.2] m. The top row shows the joint evolution, the bottom row shows the Cartesian evolution. The dashed lines in the plots of top row indicate the desired final joint angles and the red dot in the plots of bottom row indicates the desired position. The control input is obtained considering the robot as rigid (a),(e); in this case, the robot reaches the final position [0.009, 0.2034] m. Then, the same control input is applied to three 2DoF articulated soft robots (ASR) with different joint stiffness values. In (b),(f) the joint stiffness is 10 Nm/rad, and the the final end-effector position is [0.012, 0.204] m. In (c),(g) the joint stiffness is 3 Nm/rad, and the the final end-effector position is [0.04, 0.21] m. In (d),(h) the joint stiffness is 0.01 Nm/rad, and the the final end-effector position is [−0.0156, 0.2023] m. These results highlight the limit of modeling soft robot links as rigid models.

A. Motivational example

Soft robots present a model with larger state space dimension compared to rigid robots with the same number of degrees of freedom (DoF). Therefore, including the soft robot model into the optimal control problem inevitably increases the computational load, which is caused by operations like dynamics computation and other such operations part of the optimal control routine. Thus, it is natural to question if we really need to use the *soft models*. To answer this, we consider an end-effector regulation task. In this task, we command a 2DoF soft actuated system (the physical parameters of this 2DoF system are introduced in Section V-A) using an optimal control sequence that ignores the actuation dynamics (i.e., a *rigid model*). The desired final position is [0.01, 0.2] m. Fig. 2(a), 2(e) show the optimal trajectory, and robot motion, respectively. When the same control sequence is applied to an articulated soft robot with a low stiffness value, we observe an inconsistent behavior, and the end-effector position at the end of the task is far from the desired point 2(b), 2(c), 2(d). We also observe that the same control sequence shows different performance when applied to systems with varying stiffness values. Thus the use of control solutions devised for a rigid actuated model may not work well for soft actuated systems and may prove to be inconsistent.

B. Model

Consider a robot with an open kinematic chain with $n + 1$ rigid links, and n compliant joints. Let the link-side coordinates be $\mathbf{q} \in \mathbb{R}^n$, link-side velocity be $\dot{\mathbf{q}} \in \mathbb{R}^n$, motor-side coordinates $\boldsymbol{\theta} \in \mathbb{R}^m$, and motor-side velocity be $\dot{\boldsymbol{\theta}} \in \mathbb{R}^m$.

These kinds of systems usually present large reduction ratios, and the angular velocity of the rotor is due only to their own spinning. Therefore, the energy contributions due to inertial couplings between the motors and link can be neglected. Given this observation, we assume the following:

Assumption 1. *We assume that the inertial coupling between the rigid body and the motor is negligible.*

Under Assumption 1, using the Lagrangian formulation for the coupled system, one can derive the equations of motion as [32],

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t)$$
$$+ \mathbf{G}(\mathbf{q}(t)) + \frac{\partial \mathbf{U}(\mathbf{q}(t), \boldsymbol{\theta}(t))}{\partial \mathbf{q}(t)}^{\top} = \mathbf{0} \quad (1)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}}(t) + \frac{\partial \mathbf{U}(\mathbf{q}(t), \boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)}^{\top} - \boldsymbol{\tau}(t) = \mathbf{0}, \quad (2)$$

where, $\mathbf{M}(\mathbf{q}(t)) \in \mathbb{R}^{n \times n}$ is the robot inertia matrix, $\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \in \mathbb{R}^{n \times n}$ contains the centripetal and Coriolis terms, and $\mathbf{G}(\mathbf{q}(t)) \in \mathbb{R}^n$ is the gravity term, $\mathbf{B} \in \mathbb{R}^{m \times m}$ is the motor inertia, $\mathbf{U}(\mathbf{q}(t), \boldsymbol{\theta}(t))$ is the elastic potential, and $\boldsymbol{\tau}(t) \in \mathbb{R}^m$ is the torque. The general nonlinear characterization of the motor-side can be considered but we operate in the linear region of the deflection. Thus we assume that:

Assumption 2. *The elastic coupling is linear in $\mathbf{q}$ and $\boldsymbol{\theta}$.*

Using Assumption 2, the torque due to elastic potential is linear i.e. $\frac{\partial \mathbf{U}(\mathbf{q}(t), \boldsymbol{\theta}(t))}{\partial \mathbf{q}(t)}^{\top} = \mathbf{K}(t)(\mathbf{q}(t) - \mathbf{S}\boldsymbol{\theta}(t))$ and $\frac{\partial \mathbf{U}(\mathbf{q}(t), \boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)}^{\top} = \mathbf{S}^{\top}\mathbf{K}(t)(\mathbf{S}\boldsymbol{\theta}(t) - \mathbf{q}(t))$. Here $\mathbf{K}(t)$ is a

stiffness matrix and $\mathbf{S} \in \mathbb{R}^{n \times m}$ is the selection matrix. The selection matrix $\mathbf{S}$ is of rank m. The stiffness matrix $\mathbf{K}$ can be either constant or time-varying corresponding to SEA and VSA, respectively. In the case of SEA, the stiffness of each actuated joint is fixed to some σ. In the case of VSA, the stiffness of each actuated joint can vary between $\sigma_{\min}$ and $\sigma_{\max}$ and to maintain positivity of the spring stiffness we impose $\sigma_{\min} > 0$. Similarly, under-acutated compliant arm refers to the systems with the rank of selection matrix (rank($\mathbf{S}$)) being less than m and the joints can be either actuated by SEA/VSA.

Now, using the linearity of elastic coupling, (1)-(2) reduce to,

$$
\begin{aligned}
\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + \\
\mathbf{G}(\mathbf{q}(t)) + \mathbf{K}(t)(\mathbf{q}(t) - \mathbf{S}\boldsymbol{\theta}(t)) = \mathbf{0},
\end{aligned}
\tag{3}
$$

$$
\mathbf{B}\ddot{\boldsymbol{\theta}}(t) + \mathbf{S}^\top \mathbf{K}(t)(\mathbf{S}\boldsymbol{\theta}(t) - \mathbf{q}(t)) - \boldsymbol{\tau}(t) = \mathbf{0}. \tag{4}
$$

It is worth mentioning that the model class in (3)-(4) is also used to model flexible link robots in some state of the art papers [11], [12], [19], [33] and in some soft robot simulators [34].

In the case where all the joints are actuated, $\mathbf{S}$ is the identity matrix, and $n = m$, then (3)-(4) can be written as

$$
\begin{aligned}
\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + \\
\mathbf{G}(\mathbf{q}(t)) + \mathbf{K}(t)\mathbf{q}(t) - \boldsymbol{\theta}(t)) = \mathbf{0},
\end{aligned}
\tag{5}
$$

$$
\mathbf{B}\ddot{\boldsymbol{\theta}}(t) + \mathbf{K}(t)(\boldsymbol{\theta}(t) - \mathbf{q}(t)) - \boldsymbol{\tau} = \mathbf{0}. \tag{6}
$$

The rotors of the actuators are designed with their COM on the rotor axis to extend the life of the electrical drives. The motor inertia matrix is diagonal as a result of this. Further, the stiffness matrix should be invertible to ensure consistent solutions to (1)-(6).

Assumption 3. *The motor inertia matrix is diagonal.*

Using Assumption 3, $\mathbf{B}(t)$ can be written as:

$$
\mathbf{B}(t)_{i,j} = \begin{cases} B_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{7}
$$

Further to simplify computation, $\mathbf{K}(t)$ is diagonal and this resembles the case where one spring is coupled between a rotor and a link. Thus, $\mathbf{K}(t)$ can be written as,

$$
\mathbf{K}(t)_{i,j} = \begin{cases} \sigma_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{8}
$$

In the following, for the sake of simplicity, we will omit the explicit time dependence.

C. Goals

Using an optimal control approach, we aim to solve dynamic tasks for robots actuated by SEA, VSA and underactuated compliant robots. In this case, the forward dynamics will be determined by (3)-(4) or (5)-(6). Additionally, we aim to exploit feedback gains to increase performance and stabilization properties. The tasks presented in this paper are end effector regulation tasks for SEA/VSA and swing-up for the underactuated compliant systems case.

D. Optimal control formulation

We formulate a discrete-time optimal control problem for soft robots as follows:

$$
\begin{aligned}
\min_{(\mathbf{q}_s, \dot{\mathbf{q}}_s, \boldsymbol{\theta}_s, \dot{\boldsymbol{\theta}}_s), (\boldsymbol{\tau}_s)} & \ell_N(\mathbf{q}_N, \dot{\mathbf{q}}_N, \boldsymbol{\theta}_N, \dot{\boldsymbol{\theta}}_N) \\
& + \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \ell_k(\mathbf{q}_k, \dot{\mathbf{q}}_k, \boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k, \boldsymbol{\tau}_k) dt \\
\text{s.t.} \quad & [\mathbf{q}_{k+1}, \dot{\mathbf{q}}_{k+1}, \boldsymbol{\theta}_{k+1}, \dot{\boldsymbol{\theta}}_{k+1}] = \boldsymbol{\psi}(\dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k, \dot{\boldsymbol{\theta}}_k, \ddot{\boldsymbol{\theta}}_k), \\
& [\ddot{\mathbf{q}}_k, \ddot{\boldsymbol{\theta}}_k] = \text{FD}(\mathbf{q}_k, \dot{\mathbf{q}}_k, \boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k, \boldsymbol{\tau}_k), \\
& [\mathbf{q}_k, \boldsymbol{\theta}_k] \in \mathcal{Q}, [\dot{\mathbf{q}}_k, \dot{\boldsymbol{\theta}}_k] \in \mathcal{V}, \boldsymbol{\tau}_k \in \mathcal{U},
\end{aligned}
$$

where, $\mathbf{q}_k, \dot{\mathbf{q}}_k, \boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k$ and $\boldsymbol{\tau}_k$ describe the configuration point, generalized velocity, motor-side angle, motor-side velocity, joint torque commands of the system at time-step (node) k; ℓ_N is the terminal cost function; ℓ_k is the running cost function; $\boldsymbol{\psi}(\cdot)$ defines the integrator function; $\text{FD}(\cdot)$ represents the forward dynamics of the soft robot; $\mathcal{Q}$ represents the admissible state space; $\mathcal{V}$ describes the admissible velocity space and $\mathcal{U}$ defines the allowed control.

IV. SOLUTION

We solve the optimal control problem described in Section III-D using the Box-FDDP algorithm. This section first summarizes the Box-FDDP algorithm, which is a variant of the DDP algorithm, and then analyzes the dynamics and analytical derivatives of robots with SEAs, VSAs, and the underactuated compliant robots. To account for the cost incurred by the mechanism implementing the variable stiffness in VSAs, we also introduce a cost function used in systems actuated by VSA. Finally, we describe the state-feedback controller derived from Box-FDDP. We would like to emphasize that Box-FDDP is developed in [29] and is not a novel contribution of this work.

iLQR/DDP methods are known to be prone to numerical instabilities as these are single-shooting methods. Whereas, FDDP is a multiple shooting method and thus provides numerical benefits like better numerical stability. The feasibility-driven search and the nonlinear roll-out features of the algorithm ensure better convergence under poor initialization, enabling better performance for highly nonlinear problems compared to iLQR/DDP [15]. Box-FDDP is a variant of the FDDP algorithm which can handle box constraints on control variables. Box-FDDP is a more general algorithm that is based on projected Newton updates to account for the box constraints on control variables. Box-FDDP reduces to Newton updates of FDDP in the case without box constraints on the control variables. The method also provides a locally optimal feedback policy which is expected to improve performance in various tasks. This ability to handle optimal control problems for highly nonlinear systems with the option of unfeasible guess trajectory and the synthesis of feedback policies makes FDDP/Box-FDDP a suitable candidate for articulated soft robots.

A. Background on Box Feasibility-Driven DDP

DDP solves optimal control problems by breaking down the original problem into smaller sub-problems. So instead of finding the entire trajectory at once, it recursively solves the Bellman optimal equation backwards in time. To handle control bounds and improve globalization properties, the Box-FDDP algorithm modifies the backward and forward passes of DDP.

The Bellman relation is stated as

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k} \ell_k(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)), \qquad (9)$$

where, $V(\mathbf{x}_k)$ is the value function at the node k, $V_{k+1}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k))$ is the Value function at the node $k+1$, ℓ is the one step cost, $\mathbf{x}$ is the state vector ($\mathbf{x} \triangleq [\mathbf{q}^\top, \dot{\mathbf{q}}^\top, \boldsymbol{\theta}^\top, \dot{\boldsymbol{\theta}}^\top]^\top$), $\mathbf{u}$ is the control vector and $\mathbf{f}(\mathbf{x}, \mathbf{u})$ represents the dynamics of the system.

FDDP uses a quadratic approximation of the differential change in (9)

$$\Delta V = \min_{\delta \mathbf{u}_k} \frac{1}{2} \begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{u}_k \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_{\mathbf{xx}_k} & \mathbf{Q}_{\mathbf{xu}_k} \\ \mathbf{Q}_{\mathbf{ux}_k} & \mathbf{Q}_{\mathbf{uu}_k} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{u}_k \end{bmatrix} \qquad (10)$$
$$+ \begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{u}_k \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_{\mathbf{x}_k} \\ \mathbf{Q}_{\mathbf{u}_k} \end{bmatrix}.$$

$\mathbf{Q}$ is the local approximation of the action-value function and its derivatives are

$$\mathbf{Q}_{\mathbf{xx}_k} = \ell_{\mathbf{xx}_k} + \mathbf{f}_{\mathbf{x}_k}^\top V_{\mathbf{xx}_{k+1}} \mathbf{f}_{\mathbf{x}_k}, \qquad \mathbf{Q}_{\mathbf{x}_k} = \ell_{\mathbf{x}_k} + \mathbf{f}_{\mathbf{x}_k}^\top V_{\mathbf{x}_{k+1}}^+,$$
$$\mathbf{Q}_{\mathbf{uu}_k} = \ell_{\mathbf{uu}_k} + \mathbf{f}_{\mathbf{u}_k}^\top V_{\mathbf{xx}_{k+1}} \mathbf{f}_{\mathbf{u}_k}, \qquad \mathbf{Q}_{\mathbf{u}_k} = \ell_{\mathbf{u}_k} + \mathbf{f}_{\mathbf{u}_k}^\top V_{\mathbf{x}_{k+1}}^+,$$
$$\mathbf{Q}_{\mathbf{xu}_k} = \ell_{\mathbf{xu}_k} + \mathbf{f}_{\mathbf{x}_k}^\top V_{\mathbf{xx}_{k+1}} \mathbf{f}_{\mathbf{u}_k},$$

where, $V_{\mathbf{x}_{k+1}}^+ = V_{\mathbf{x}_{k+1}} + V_{\mathbf{xx}_{k+1}} \bar{\mathbf{f}}_{k+1}$ is the Jacobian of the value function, $\ell_{\mathbf{x}_k}$ is the Jacobian of the one step cost, $\ell_{\mathbf{xx}_k}$ is the Hessian of the one step cost and $\bar{\mathbf{f}}_{k+1}$ is the deflection in the dynamics at the node $k + 1$:

$$\bar{\mathbf{f}}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1}.$$

1) Backward Pass: In the backward pass, the search direction is computed by recursively solving

$$\delta \mathbf{u}_k = \arg\min_{\delta \mathbf{u}_k} \mathbf{Q}(\delta \mathbf{x}_k, \delta \mathbf{u}_k) = \hat{\mathbf{k}} + \hat{\mathbf{K}} \delta \mathbf{x}_k,$$
$$\text{s.t.} \quad \underline{\mathbf{u}} \le \mathbf{u}_k + \delta \mathbf{u}_k \le \bar{\mathbf{u}}, \qquad (11)$$

where, $\hat{\mathbf{k}} = -\hat{\mathbf{Q}}_{\mathbf{uu}_k}^{-1} \mathbf{Q}_{\mathbf{u}_k}$ is the feed-forward term and $\hat{\mathbf{K}} = -\hat{\mathbf{Q}}_{\mathbf{uu}_k}^{-1} \mathbf{Q}_{\mathbf{ux}_k}$ is the feedback term at the node k, and $\hat{\mathbf{Q}}_{\mathbf{uu}_k}$ is the control Hessian of the free subspace. Using the optimal $\delta \mathbf{u}_k$, the gradient and Hessian of the Value function are updated.

2) Forward Pass: Once the search direction is obtained in (11), then the step size α is chosen based on an Armijo-based line search routine. The control and state trajectory are updated using this step size

$$\hat{\mathbf{u}}_k = \mathbf{u}_k + \alpha \hat{\mathbf{k}} + \hat{\mathbf{K}}(\hat{\mathbf{x}}_k - \mathbf{x}_k), \qquad (12)$$
$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) - (1 - \alpha) \bar{\mathbf{f}}_{k-1}, \qquad (13)$$

where, $\{\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k\}$ are the state and control vectors. In problems without control bounds, the algorithm reduces to FDDP [15]. The interested reader is referred to [15], [29] for more details about the algorithm.

B. Dynamics for soft robots

The forward dynamics in (5)-(6) can be written in compact form as follows:

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\tau}_l \\ \boldsymbol{\tau}_m \end{bmatrix}, \qquad (14)$$

where,

$$\boldsymbol{\tau}_l \triangleq -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q}) - \mathbf{K}(\mathbf{q} - \boldsymbol{\theta}), \qquad (15)$$
$$\boldsymbol{\tau}_m \triangleq \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) + \boldsymbol{\tau}. \qquad (16)$$

We compute link-side dynamics efficiently via the use articulated body algorithm (ABA) for the first block of effective inertia matrix (which corresponds to the rigid body algorithm). We then use the analytical inversion of $\mathbf{B}$ to efficiently compute the motor-side dynamics in (14).

The forward dynamics computation in (3)-(4) can be done similarly to the above process with a different definition of $\boldsymbol{\tau}_l$ and $\boldsymbol{\tau}_m$

$$\boldsymbol{\tau}_l \triangleq -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q}) - \mathbf{K}(\mathbf{q} - \mathbf{S}\boldsymbol{\theta}), \qquad (17)$$
$$\boldsymbol{\tau}_m \triangleq -\mathbf{S}^\top \mathbf{K}(\mathbf{S}\boldsymbol{\theta} - \mathbf{q}) + \boldsymbol{\tau}. \qquad (18)$$

The forward dynamics computation is summarized in Algorithm 1:

Algorithm 1 Forward dynamics

1: **Input:** `robotModel`, $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$

2: **Output:** $\ddot{\mathbf{q}}, \ddot{\boldsymbol{\theta}}$

3: $\ddot{\mathbf{q}} \leftarrow$ Articulated Body Algorithm(ABA) ($\mathbf{q}, \mathbf{v}, \mathbf{0}$) $+ \mathbf{M}^{-1}(-\mathbf{K}(\boldsymbol{\theta} - \mathbf{q}))$

4: $\ddot{\boldsymbol{\theta}} \leftarrow \mathbf{B}^{-1}(\boldsymbol{\tau} + \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}))$

$\mathbf{M}^{-1}$ is computed as part of the forward dynamics algorithm. Additionally, the computation of $\ddot{\boldsymbol{\theta}}$ involves inversion of $\mathbf{B}$ which in itself is diagonal in our cases.

C. Analytical derivatives

The block diagonal structure of the inertia matrix in (14) allows us to independently evaluate the partial derivatives related to the link-side and motor-side. Among several methods to compute the partial derivatives of the dynamics, the finite difference method is popular. This is because the difference between the input dynamics is computed $n + 1$ times while perturbing the input variables. The successful implementation of numerical differentiation requires fine parallelization techniques, thus the finite difference could result in computational complexity of $\mathcal{O}(n^2)$ [35]. Another way is to derive the Lagrangian equation of motion, which requires only one function

call. We use PINOCCHIO [36], an efficient library for rigid body algorithms, which exploits sparsity induced by kinematic patterns to compute the analytical derivatives with $\mathcal{O}(n)$ cost. Now we illustrate the analytical derivatives for SEA/VSA and the underactuated compliant model.

1) Series elastic actuation: To solve the full dynamics model in the Box-FDDP/FDDP formalism we define the state vector as $\mathbf{x} \triangleq [\mathbf{q}^\top \ \dot{\mathbf{q}}^\top \ \boldsymbol{\theta}^\top \ \dot{\boldsymbol{\theta}}^\top]^\top$, and the input vector is $\mathbf{u} = [\boldsymbol{\tau}]$.

An explicit inversion of the KKT matrix is avoided in the forward pass by inverting the matrix analytically:

$$\begin{bmatrix} \delta\ddot{\mathbf{q}} \\ \delta\ddot{\boldsymbol{\theta}} \end{bmatrix} = - \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B} \end{bmatrix}^{-1} \left(\begin{bmatrix} \frac{\partial \boldsymbol{\tau}_l}{\partial \mathbf{x}} \\ \frac{\partial \boldsymbol{\tau}_m}{\partial \mathbf{x}} \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \frac{\partial \boldsymbol{\tau}_l}{\partial \mathbf{u}} \\ \frac{\partial \boldsymbol{\tau}_m}{\partial \mathbf{u}} \end{bmatrix} \delta\mathbf{u} \right). \quad (19)$$

The matrix has a diagonal block structure and can be switched separately. The motor inertia matrix is diagonal for all practical purposes and thus can be analytically inverted. Using the definition of $\boldsymbol{\tau}_l$ in (15) one can analytically compute the Jacobians. Here we only list the non-zero components, i.e.,

$$\frac{\partial \boldsymbol{\tau}_l}{\partial \mathbf{q}} = -\frac{\partial \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} - \frac{\partial \mathbf{G}(\mathbf{q})}{\partial \mathbf{q}} - \mathbf{K}, \quad (20)$$

$$\frac{\partial \boldsymbol{\tau}_l}{\partial \dot{\mathbf{q}}} = -\frac{\partial \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}, \qquad \frac{\partial \boldsymbol{\tau}_l}{\partial \boldsymbol{\theta}} = -\mathbf{K}, \quad (21)$$

$$\frac{\partial \boldsymbol{\tau}_m}{\partial \mathbf{q}} = \mathbf{K} + \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}}, \qquad \frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\theta}} = \mathbf{K}. \quad (22)$$

Similarly the Jacobian w.r.t. $\mathbf{u}$ is $\frac{\partial \boldsymbol{\tau}_m}{\partial \mathbf{u}} = \mathbf{I}$. Using the same principles, the analytical derivatives of the cost function w.r.t. state and control can be derived.

2) Variable stiffness actuation: In the case of variable stiffness actuators, we model the system using similar equations, but stiffness at each joint is treated as a decision variable. Thus the state vector is still $\mathbf{x} \triangleq [\mathbf{q}^\top \ \dot{\mathbf{q}}^\top \ \boldsymbol{\theta}^\top \ \dot{\boldsymbol{\theta}}^\top]^\top$, but the decision vector is $\mathbf{u} \triangleq [\boldsymbol{\tau}^\top \boldsymbol{\sigma}^\top]^\top$ where, $\boldsymbol{\sigma}$ is the vector of diagonal entries from $\mathbf{K}$. So the Jacobians w.r.t. the state variables remain the same as (20)-(22). Now the derivatives w.r.t. decision vector are

$$\frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\tau}} = \mathbf{I}, \qquad \frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\sigma}} = \boldsymbol{\theta} - \mathbf{q}, \qquad \frac{\partial \boldsymbol{\tau}_l}{\partial \boldsymbol{\sigma}} = \mathbf{q} - \boldsymbol{\theta}. \quad (23)$$

To effectively incorporate the constraint on $\boldsymbol{\sigma}$, we impose a box constraint on the stiffness variables $\sigma_{\min_i} < \sigma_i < \sigma_{\max_i}$, where σ_i is the i-th component of the $\boldsymbol{\sigma}$ vector. We use the Box-FDDP algorithm in [29], [37] to solve constrained optimal control problems with box constraints on control variables (Sec. III-D).

3) Under-actuated Compliant Arm: For underactuated compliant systems, $\mathbf{q} \in \mathbb{R}^n$ and $\boldsymbol{\theta} \in \mathbb{R}^m$ are of different dimensions. Moreover, their analytical derivatives are different from the fully actuated flexible joint case (i.e., (20)-(23)) as shown below:

$$\frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\theta}} = -\mathbf{S}^\top \mathbf{K} \mathbf{S}, \quad \frac{\partial \boldsymbol{\tau}_m}{\partial \mathbf{q}} = \mathbf{S}^\top \mathbf{K}, \quad \frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\sigma}} = \mathbf{S}^\top (\mathbf{S}\boldsymbol{\theta} - \mathbf{q}), \quad (24)$$

$$\frac{\partial \boldsymbol{\tau}_l}{\partial \boldsymbol{\theta}} = \mathbf{K} \mathbf{S}, \qquad \frac{\partial \boldsymbol{\tau}_l}{\partial \boldsymbol{\sigma}} = -(\mathbf{q} - \mathbf{S}\boldsymbol{\theta}). \quad (25)$$

The analytical derivatives of the dynamics are summarized in Algorithm 2:

Algorithm 2 Analytical Derivatives

  Input: `robotModel`, $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}$

2: **Output:** $\frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}}, \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}}, \frac{\partial \ddot{\mathbf{q}}}{\partial \boldsymbol{\theta}}, \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \mathbf{q}}, \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}, \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \dot{\boldsymbol{\theta}}}$

  $\frac{\partial \boldsymbol{\tau}_{rb}}{\partial \mathbf{q}}, \frac{\partial \boldsymbol{\tau}_{rb}}{\partial \dot{\mathbf{q}}}$ ← Compute Recursive Newton Euler algorithm(RNEA) derivatives$(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

4: $\frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} = \mathbf{M}^{-1}(\frac{\partial \boldsymbol{\tau}_{rb}}{\partial \mathbf{q}} - \mathbf{K}); \qquad \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} = \mathbf{M}^{-1}(\frac{\partial \boldsymbol{\tau}_{rb}}{\partial \dot{\mathbf{q}}});$
  $\frac{\partial \ddot{\mathbf{q}}}{\partial \boldsymbol{\theta}} = \mathbf{M}^{-1}(\mathbf{K});$

  $\frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \mathbf{q}} = \mathbf{B}^{-1}(\mathbf{K}); \qquad \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} = -\mathbf{B}^{-1}(\mathbf{K}); \qquad \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \dot{\boldsymbol{\theta}}} = \mathbf{0}$

D. VSA cost function

The physical mechanism that implements variable stiffness consumes energy. To include this cost of changing stiffness in the optimal control problem, we define a linear cost in stiffness [38].

$$\ell_{\text{vsa}} = \sum_{j=1}^{m} \int_0^T \lambda(\sigma_j - \sigma_r) \mathrm{d}t, \quad (26)$$

where, m is the number of actuated joints, σ_r is the stiffness value under no-load conditions, σ_j is the stiffness value of the joint j, and ℓ_{vsa} is the one step VSA mechanism cost. Using this term, we ensure that the cost is zero at σ_r and the cost is imposed only when stiffness is varied. The overall cost value is dependent on the motor mechanics and it is modulated by λ. The value of λ is related to the torque value required to maintain a particular stiffness value. To ensure that $\|\boldsymbol{\tau}\|_2^2 < \ell_{\text{vsa}}$, the λ is assumed to be a linear interpolation between the $\sigma_{\min}$ and $\sigma_{\max}$

$$\lambda = \frac{g^2(\sigma_{\max}) - g^2(\sigma_{\min})}{\sigma_{\max} - \sigma_{\min}},$$

where, $g^2(\cdot)$ is a function defined to represent the stiffness change for a specific actuator used.

For variable stiffness actuator with an antagonastic mechanism such as [39], $g^2(\cdot) = \tau_1^2 + \tau_2^2$. The actual cost incurred in change of stiffness is related to the square of the torque curve and the design of the cost function (26), ensures that it overestimates the cost incurred due to a change in stiffness [38].

E. State feedback controller

Feedback controllers based on PD gains are user tuned and sub-optimal. Instead of using sub-optimal policies to track the optimal trajectory, we propose to both plan the optimal trajectory and use the local policy obtained from the backward pass of Box-FDDP/FDDP. In Section IV-A, the feedback gain matrix is computed in the backward pass to ensure strict adherence to the optimal policy as described in [28]. Thus the controller can be written as

$$\mathbf{u} = \hat{\mathbf{k}} + \hat{\mathbf{K}}(\mathbf{x}^* - \mathbf{x}). \quad (27)$$

TABLE I
PHYSICAL PARAMETERS OF 7DoF ARM

	m_i[kg]	l_i[m]		m_i[kg]	l_i[m]
Link 1	1.42	0.1	Link 2	1.67	0.5
Link 3	1.47	0.06	Link 4	1.10	0.06
Link 5	1.77	0.06	Link 6	0.3	0.06
Link 7	0.3	0.06			

The optimal control formulation, which considers the complete dynamics and costs, is used to calculate this local and optimal control policy. The feedback matrix for a VSA-actuated system is $\hat{\mathbf{K}} \in \mathbb{R}^{2m \times (n+m)}$, which also produces state feedback to the stiffness control alongwith the input torques. Furthermore, it is computationally less expensive than using a separate feedback controller.

To calculate this local and optimal control policy, we formulate an optimal control problem that considers the complete dynamics and costs. The feedback matrix for a VSA-actuated system is $\hat{\mathbf{K}} \in \mathbb{R}^{2m \times (n+m)}$, which also produces state-feedback gains for stiffness control along with the input torques. Furthermore, it is computationally less expensive than using a separate feedback controller.

V. VALIDATION SETUP

In this section, we introduce the simulation and experimental setup. Results and discussion will be presented in Section VI.

A. System setup

We employ six different compliant systems: (a) a 2DoF robot with SEAs at each joint; (b) a 2DoF robot with VSA at each joint; (c) a 4DoF robot with SEAs at each joint; (d) a 4DoF robot with VSAs at each joint; (e) a 7DoF system with SEA at each joint; (f) a 7DoF system with VSA at each joint; (g) an underactuated compliant robot modeled as a 2DoF robot where the first joint is actuated by a SEA and the second elastic joint is unactuated; and (h) an underactuated compliant robot modeled as a 2DoF robot where the first joint is actuated by a VSA and the second elastic joint is unactuated (i) an underactuated serial manipulator with 21 elastic joints and only the first one is actuated by SEA. We perform simulations and experiments for (a), (b), (c), (d), (g), (h) and only simulations for (e), (f), (i). We now introduce the various systems

The physical parameters for 2DoF, 4DoF robots with SEA and VSA are: the mass of the links $m_i = 0.55$ kg, the inertia of the motors $\mathbf{B}_{i,i} = 10^{-3}$ kgm^2, the center of mass distance $a_i = 0.085$ m, and the link lengths $l_i = 0.089$ m. Similarly, we consider the following physical parameters for the 7DoF system actuated by SEA and VSA at each joint. The physical parameters resemble those of a Talos arm [40] (Table I), but we add a compliant actuation at each joint. The motor inertia is set to $\mathbf{B}_{i,i} = 10^{-3}$ kgm^2 for all joints.

We classify our results as follows:

1) In Section VI-A, we report the difference between the Jacobians of the dynamics computed by numerical differentiation and analytical differentiation. The simulation is conducted for the 2DoF and 7DoF cases actuated by both SEA and VSA. We show the average and standard deviation of the computed difference at 20 random configurations and with velocities and control inputs set as zero. The number of iterations for convergence and the time taken for convergence are also reported.

2) In Section VI-B, we present the simulation and experimental results of the end-effector regulation task with SEAs and VSAs for a 2DoF, 4DoF and 7DoF arms.

   - *2DoF robot*: the Cartesian target point is $[0.01, \ 0.2]$ m, the time horizon is $T = 3$ s, and the weights corresponding to control regularization is 10^{-2}, state regularization is 1, goal-tracking cost is 10^{-1}, and terminal cost contains a goal-tracking cost with 10^4 weighting factor. In the case of SEAs, we define the stiffness of each joint as $\sigma_i = 3$ Nm/rad. Instead, in the case of VSAs, the value of the stiffness lie between $\sigma_{\min} = 0.05$ Nm/rad and $\sigma_{\max} = 15$ Nm/rad. We also include an additional cost term (26) which accounts for the power consumption due to change in the stiffness and the weight is set to 1 and the λ is set to 10.

   - 4DoF robot: the Cartesian target point is $[.1, .3, .15]$ m, the time horizon is $T = 4$ s, and the weights corresponding to control regularization is 10^{-1}, state regularization is 10^{-3}, goal-tracking cost is 10^0, and terminal cost contains a goal-tracking cost with 10^4 weighting factor. In the case of SEAs, we define the stiffness of the first joint as $\sigma_i = 10$ Nm/rad each joint as $\sigma_i = 5$ Nm/rad. Instead, in the case of VSAs, the Cartesian target point is $[.15, .3, .15]$ m, the value of the stiffness lies between $\sigma_{\min} = 2$ Nm/rad and $\sigma_{\max} = 15$ Nm/rad. We also include an additional cost term (26) which accounts for the power consumption due to change in the stiffness and the weight is set to 1 and the λ is set to 10.

   - *7DoF robot*: the Cartesian target point is set as $[0, 0, 0.4]$ m, and the time horizon is $T = 1.5$ s. The weights corresponding to control regularization is 10^{-2}, state regularization is 1 and goal-tracking cost is 10^{-1}. In the case of SEAs, we define the stiffness value of each joint is set to $\sigma = 10$ Nm/rad. Instead, in the case of VSAs, the stiffness value of each joint varies between $\sigma_{\min} = 0.7$ Nm/rad and $\sigma_{\max} = 10$ Nm/rad.

3) In Section VI-C, we illustrate the results obtained for underactuated compliant robots with varying DoFs and varying degrees of underactuation.

   - *Underactuated 2DoF compliant arm:* In the case where the first joint is actuated by a SEA, the stiffness constant $\sigma = 5$ Nm/rad and the time horizon is $T = 3$ s. For the case where the first joint is actuated by VSA, the value of $\sigma_{\min} = 0.05$ Nm/rad and $\sigma_{\max} = 15$ Nm/rad and the time horizon is $T = 2$ s. The weights corresponding to control regularization is 10^{-1}, state regularization is 10^{-2}, ℓ_{vsa} is 10^{-2} and the goal-tracking cost is 10^{-1}. The stiffness of the second link is 2 Nm/rad.

- *Underactuated serial manipulator*: The stiffness matrix of a flexible link can be written as $\mathbf{K} = \mathrm{diag}(\mathbf{K}_{11}, ..., \mathbf{K}_{mm})$. Further, the motor inertias are considered to be negligible. We model an underactuated serial manipulator as a 21 joint under-actuated compliant arm with the total length being 3.15 m and only the first joint is actuated and has 20 passive elastic joints.

4) In the final Section VI-D, we compare the energy consumption in an end-effector regulation task, for rigid, SEA, and VSA cases. We report the results with three different systems: a fully actuated 2DoF system, a fully actuated 7DoF system and, a 2DoF underactuated compliant arm. For comparison purposes, the weights of the various cost terms in the objective function for the task for rigid, SEA, and VSA are the same.

For cases involving SEAs, we use the FDDP solver, and for cases with VSAs, we impose these box constraints on the stiffness values using the Box-FDDP solver. Both solvers used in the simulations and experiments are available in the CROCODDYL library [15].

B. Experimental setup

We use the experimental setup illustrated in Fig. 1 for all experiments. At each joint, we utilize QBMOVE ADVANCED [39] as the elastic actuator. It features two motors attached to the output shaft and is based on an antagonistic mechanism. The motors and the links of the actuator both include AS5045 12 bit magnetic encoders. The actuator's elastic torque τ and nonlinear stiffness function σ satisfy the following equation

$$\tau = 2\beta \cosh \alpha\theta_{\mathrm{s}} \sinh \alpha(q - \theta_{\mathrm{e}}) , \qquad (28)$$

$$\sigma = 2\alpha\beta \cosh(\alpha\theta_{\mathrm{s}}) \cosh \alpha(q - \theta_{\mathrm{e}}) , \qquad (29)$$

where, $\alpha = 6.7328 \ \mathrm{rad}^{-2}$, $\beta = 0.0222 \ \mathrm{N\,m}$, θ_{e} is the motor equilibrium position, θ_{s} tunes the desired motor stiffness and q is the link-side position.

For the experiments related to systems with an un-actuated joint, we set the θ_{s} of the passive joint as constant and θ_{e} is set to be null. This pragmatic choice equips the passive joint with a torsional spring and position encoder. This enables us to resemble an underactuated compliant robot.

The optimal control with VSAs returns τ and σ as the output of the optimal control problem. We need to invert the equations (29) to find $\boldsymbol{\theta}_{\mathrm{e}}$ and $\boldsymbol{\theta}_{\mathrm{s}}$ as these will be input to the actual motors. The parameters like α and β can be found in the manufacturer's datasheet, and $\mathbf{q}$ is the link trajectory seen from the optimization routine. In the results obtained from the experiments, we compare the results of feedback control with pure feed-forward control. To quantify the tracking performance of these controllers, we use as metric the root mean square (RMS) error.

VI. RESULTS AND DISCUSSIONS

In this section, we present and discuss the simulation and experimental results. The code necessary to reproduce the results reported in this section are publicly available. [2]

[2]github.com/spykspeigel/aslr_to

TABLE II
RATIO BETWEEN DIFFERENCE OF NUMDIFF AND ANALYTICAL DIFF AND MAX ANALYTICAL DIFF

Problems	SEA		VSA	
	Average	Std. Dev.	Average	Std. Dev.
2DoF	$2 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$8 \cdot 10^{-6}$	$3 \cdot 10^{-7}$
7DoF	$5 \cdot 10^{-7}$	$5 \cdot 10^{-6}$	$9 \cdot 10^{-6}$	$6 \cdot 10^{-6}$

A. Analytical derivatives vs. numerical derivatives

Using analytical derivatives of the dynamics is expected to improve the numerical accuracy. We present simulation results in this subsection to support this claim. In Table II, we show the average and standard deviation of the difference between numerical differentiation and analytical differentiation. This is for at 20 random configurations and with initial velocities and controls set to zero. The values reported are the ratio with respect to the maximum value of the analytical derivative obtained among the randomized configurations for each case.

In Table III, we show the average and standard deviation of the number of iterations for convergence between numerical differentiation and analytical differentiation-based optimal control method. The 2DoF and 7DoF systems are assigned end-effector regulation tasks with 20 random desired end-effector positions which include non-zero terminal velocity. For the underactuated 2DoF system, the task is to swingup to the vertical position with 20 random initial configurations which include nonzero initial velocity. The standard deviation for numerical derivatives is significantly more than that of analytical derivative-based OC for underactuated 2DoF VSA cases, even though the average value is similar to analytical derivatives-based OC.

For the 7DoF system with each joint actuated by SEA case, the optimal control based on numerical derivatives converges more than 400 iterations in 8 out of 20 cases. Similarly, in the 7DoF system with each joint actuated by VSA, 12 out of 20 problems take more than 400 iterations to converge. One can obtain similar results for cases with higher degrees of freedom and generic under-actuation.

TABLE III
ITERATIONS TAKEN BY OPTIMAL CONTROL BASED ON NUMDIFF AND ANALYTICAL DIFF

Problems	AnalyticalDiff		NumDiff	
	Average	Std.	Average	Std.
2DoF SEA	6	2	7.1	1.6
2DoF VSA	24	2.1	24.3	2.3
under 2DoF SEA	31.1	2.1	32.6	3.3
under 2DoF VSA	39.1	6.7	40.9	38
7DoF SEA	45.6	19.68	8 cases take > 400	
7DoF VSA	47.83	15.3	12 cases take > 400	

Most importantly, the computation of analytical derivatives is computationally cheaper. In Table IV, we provide a comparison of the time taken per iteration for the optimal problem with an end-effector regulation task for 2DoF and 7Dof case and swingup-task for the under-actuated 2DoF system.

Fig. 3. Snapshots of the end-effector regulation task with the 2DoF system actuated by SEA in both joints. Please refer to the video attachment for more details.

As can be noted, the analytical derivatives provide about 100 times increased performance. A C++ implementation is expected to show increased performance and enable MPC-based implementation as can be noted in our recent works [28], [41].

TABLE IV
AVERAGE TIME TAKEN PER ITERATIONS FOR OPTIMAL CONTROL BASED ON NUMDIFF AND ANALYTICAL DIFF (IN SECONDS)

	AnalyticalDiff		NumDiff	
Problems	SEA	VSA	SEA	VSA
2DoF	0.098	0.099	3.9	4.3
under 2DoF	0.13	0.14	7.1	7.9
7DoF	0.17	0.32	23.01	23.22

B. Optimal trajectory for regulation tasks of SEA and VSA systems

Fig. 4 shows the simulation and experimental results of the 2DoF SEA case. This includes the optimal trajectory (Fig. 4(a)) and the input sequence (Fig. 4(b)). At the end of the task, the end -effector position was $[0.0098, \ 0.20]$ m. The link position in the experiment is presented in Fig. 4(c). A photo-sequence of the experiment is depicted in Fig. 3; please also refer to the video attachment. The RMS error for the first joint in the case of pure feed-forward control was 0.2503 rad and in the case of feedback control was 0.2296 rad. Similarly, for the second joint, we observe that the RMS error in the pure feed-forward case was 0.1274 rad, and with feedback control was 0.1076 rad. So, this illustrates the advantage of using feedback gain along with the feed-forward action for the task.

Similarly, Fig. 5 illustrates the results of the same end-effector regulation task for the 2DoF system actuated by VSA as described in section V-A. The end-effector position at the end of the task was $[0.011, \ 0.202]$ m. Fig. 5(d) shows plot of the link positions obtained from the experiments.

Fig. 6(a) shows a configuration of 4DoF system with SEA in each joint. It also illustrates the photo sequence of the experiment for an end-effector regulation. The desired end-effector position is [0.1, 0.3, 0.15] m, and the method was able to generate a trajectory that reaches [0.11, 0.33, 0.13] m in simulations. The link position obtained from the experiment is shown in Fig. 6(d)-6(g). In the case of pure feedback control, the RMS error for the first joint was 0.0361 rad, for the second joint was 0.0545 rad, the third joint was 0.0659 rad and the fourth joint was 0.0388 rad. Whereas using feedback control, the RMS error for the first joint was 0.0344 rad, for the second
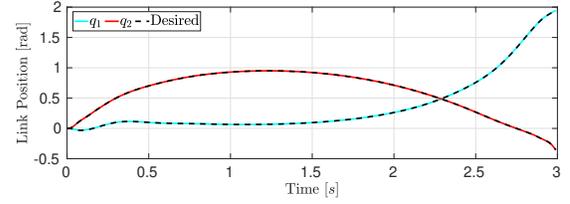
(a) Link positions
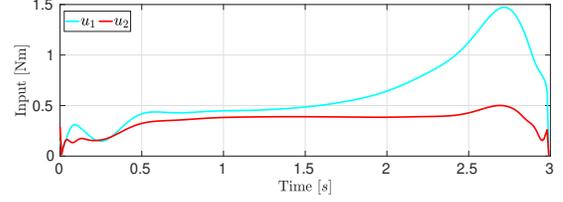
(b) Input torques
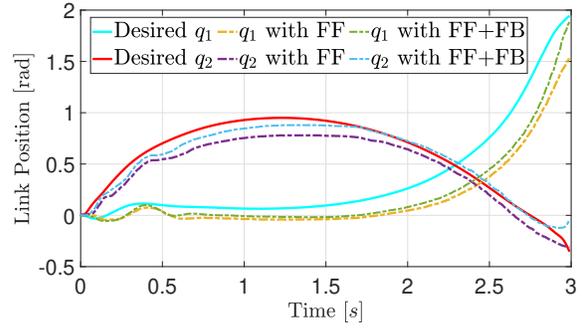
(c) Link 1 position

Fig. 4. End-effector regulation task of a 2DoF system with SEA in both joints. (a) Joint evolution in simulation. (b) Input torque evolution in simulation. (c) Evolution of joint 1 and joint 2 in experiments. We compare the desired and the link positions using pure feed-foward (FF) and the feedback and feed-forward (FF+FB) cases, which shows better performance.

joint was 0.0550 rad, for third joint was 0.0653 rad and for the fourth joint was 0.0240 rad.

Fig. 7 shows the results for the 4Dof system with VSA in each of the joints. We show the results for an end-effector regulation task with desired end-effector position as [.15, .3, .15] m and the end-effector position in the simulation was [0.134, 0.36, 0.13] m. The link position obtained from the experiment is shown in Fig. 7(d)-7(g). In the case of pure feedback control, the RMS error for the first joint was 0.0428 rad, for the second joint was 0.0230 rad, the third joint was 0.0294 rad and the fourth joint was 0.0222 rad. Whereas using feedback control, the RMS error for the first joint was 0.0429 rad, for the second joint was 0.0213 rad, for third joint was 0.0294 rad and for the fourth joint was 0.0136 rad.

Using the proposed approach, we were also able to produce optimal solutions for higher dimensional systems. In Fig. 9 we provide the simulation results, which includes the joint positions and the input sequence, for a 7DoF system with SEA at each joint.

In Fig. 10, we provide the simulation results, which includes the input sequence 10(b) and the stiffness profile 10(c), for 7DoF system with VSAs at each joint. A photo-sequence of the task is showed in Fig. 8. The error in the end-effector position for 7DoF SEA systems is 0.0091 m and for 7DoF

(a) Link Position

(b) Input Torques

(c) Stiffness Profile

(d) Links position
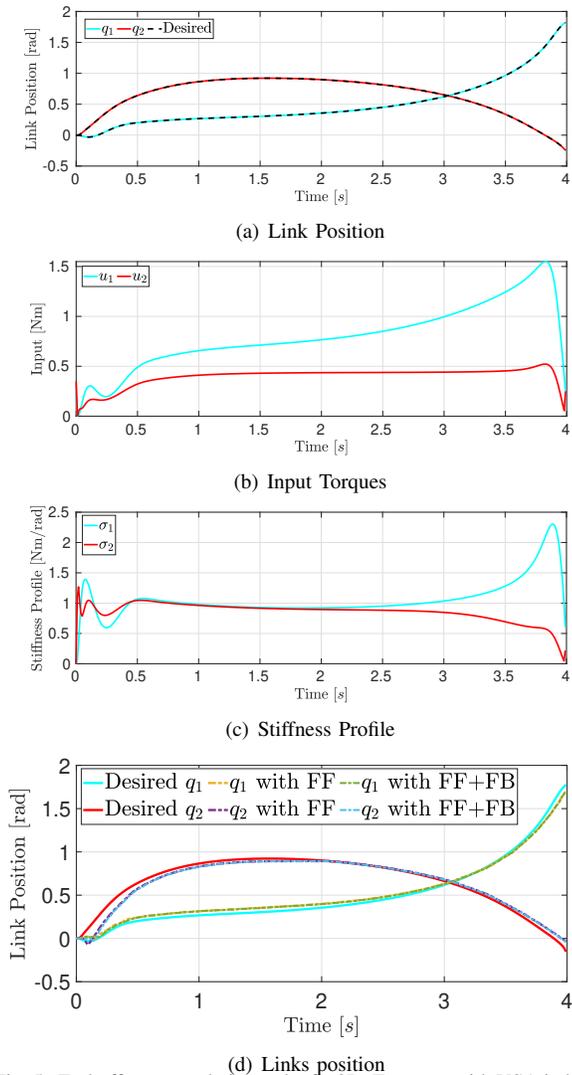
Fig. 5. End-effector regulation task of a 2DoF system with VSA in both joints. (a) Link position in simulations (b) Input torque evolution in simulation. (c) Input stiffness evolution in simulation. (d) Evolution of joint 1 and joint 2 in experiments. We compare the desired and the link positions using pure feed-foward (FF).

VSA system is 0.005.

Thus, the proposed method is capable of achieving successful results both in the case robots actuated by SEA and VSA. It can also be applied to platforms with high degree of freedom. We also show that feedback gain matrix is helpful in reducing the RMS error.

In comparison to earlier works [42], [43], the presented method was able to synthesize dynamic motion and control with a smaller time horizon (1-4 seconds). By dynamic motions, we refer to the tasks where the contribution provided by $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is comparable to the static contribution to the torque and therefore is not negligible, such that $||\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}|| \approx ||\mathbf{G}(\mathbf{q}) + \mathbf{K}\mathbf{q}||$.

Furthermore, it is worth mentioning that the employed actuator present an highly nonlinear dynamics even in the SEA case (see Section V-A). Thus, achieving good tracking performance also proves the robust aspect of the proposed method.
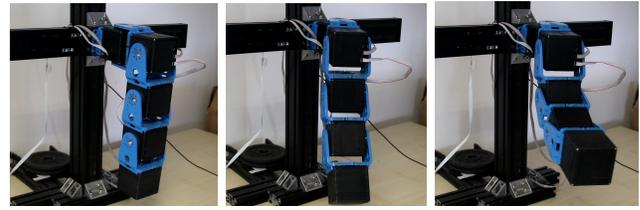
(a) Snapshots of the end-effector regulation for 4DoF with a SEA in each joint. Please refer to the video attachment for more details.

(b) Link position

(c) Input Controls

(d) Link 1 position

(e) Link 2 position

(f) Link 3 position

(g) Link 4 position
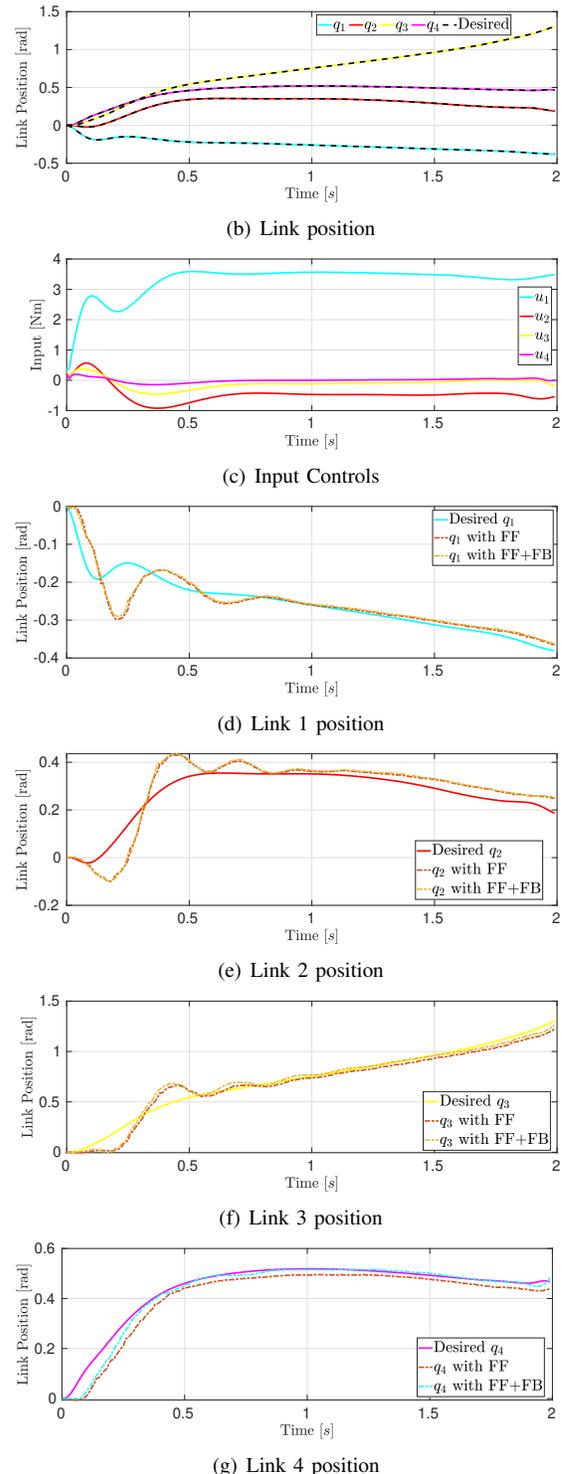
Fig. 6. End-effector regulation task of a 4DoF system with SEA in all joints. (a) Link position in simulation (b) Input Torque. The desired end-effector position was [0.1, 0.3, 0.15] m and it reached [0.11, 0.33, 0.13] m. (c), (d), (e), (f) Evolution of all joints in experiments. We compare the desired (simulation) and the link positions using pure feed-forward (FF) and feed-forward with feedback (FF+FB), which shows better performance.
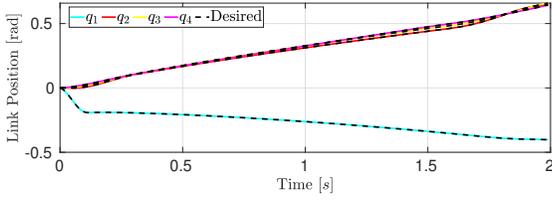
(a) Link position
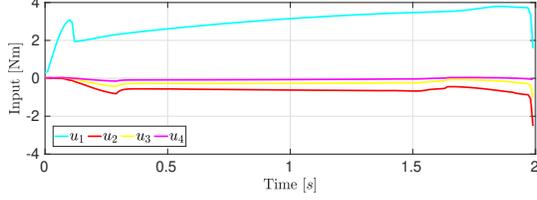
(b) Input Control
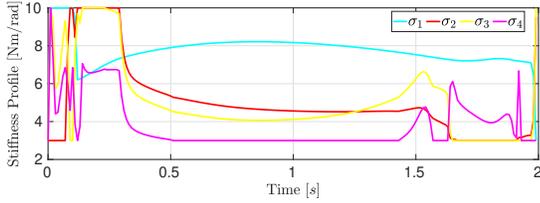
(c) Stiffness Evolution
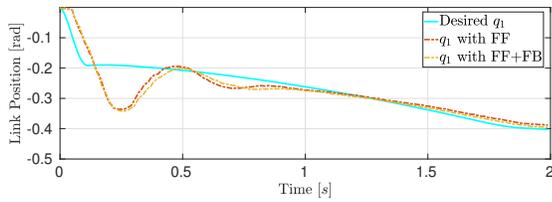
(d) Link 1 position
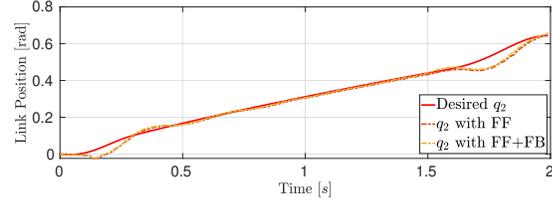
(e) Link 2 position
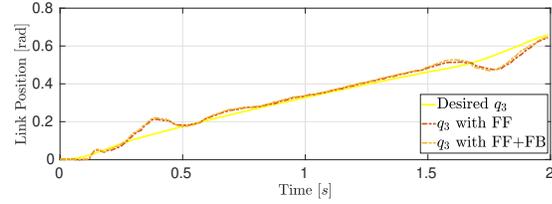
(f) Link 3 position
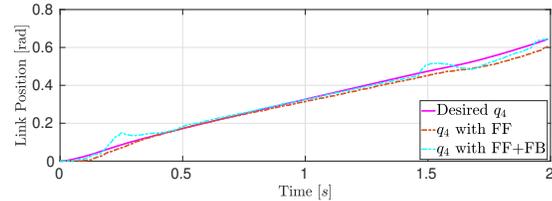
(g) Link 4 position

Fig. 7. End-effector regulation task of a 4DoF system with VSA in all joints. (a) Link position in simulation. (b) Input torque evolution in simulation. (c) Stiffness profile. The desired end-effector position was [.15, .3, .15] m and it reached [0.134, 0.36, 0.13] m. (d),(e),(f),(g) Evolution of all the joints in experiments. We compare the desired and the link positions using pure feed-forward (FF) and feed-forward with feedback (FF+FB), which shows better performance.
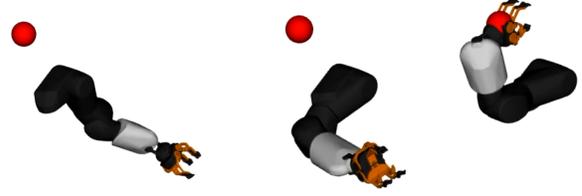
Fig. 8. Motion of the 7DoF arm with VSA in all the joints performing an end-effector regulation task. The red ball indicate the desired position.
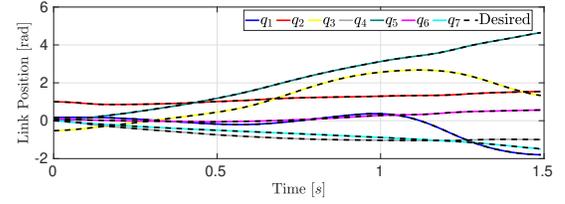
(a) Link positions
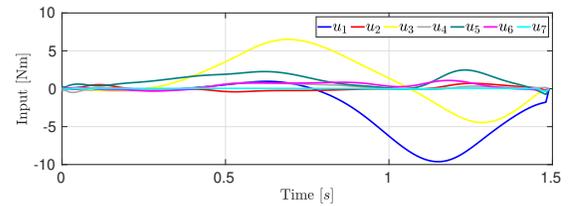
(b) Input torques

Fig. 9. End-effector regulation task of a 7DoF system with SEA in all joints. (a) Evolution of the link positions in simulation. (b) Input torque evolution in simulation.
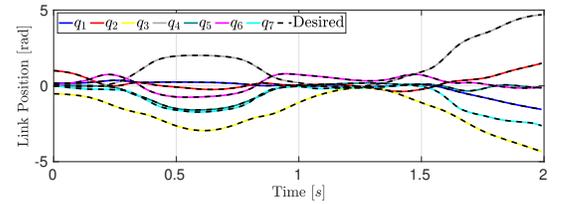
(a) Link positions
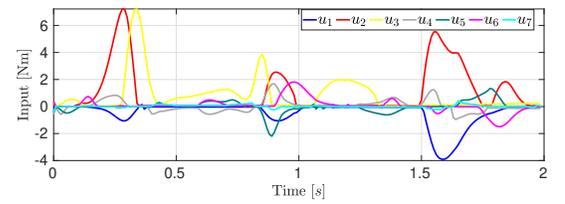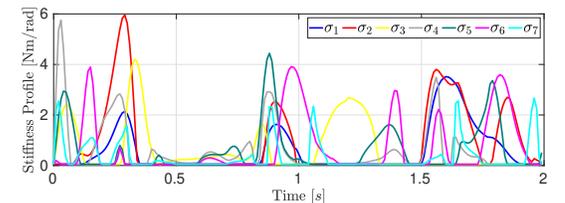
(b) Input torques

(c) Stiffness profile

Fig. 10. End-effector regulation task of a 7DoF system with VSA in all joints. (a)Evolution of the link positions in simulation. (b) Input torque evolution in simulation. (c) Input stiffness evolution in simulation.

C. Optimal control of underactuated compliant robots

Fig. 12 shows the simulation and experimental results of the swing-up task performed by the 2DoF underactuated compliant arm with a SEA in the first joint. This includes the optimal trajectory (Fig. 12(a)) and the input sequence (Fig. 12(b)). Fig. 12(c) illustrates the link positions of both the joints obtained from the experiments. Snapshots of the experiments are depicted in Fig. 11; please also refer to the video attachment. The RMS error for joint 1 in the case of pure feed-forward control was 0.3908 rad and in the case of feedforward plus feedback control was 0.3734 rad. Similarly for joint 2 we observe that RMS for pure feed-forward case was 0.1607 rad and for feedforward plus feedback control was 0.1571 rad.

The simulation and experimental results for the swing-up task of the 2DoF underactuated compliant arm with VSA in the first joint are shown in Fig. 13. The RMS error for joint 1 in the case of pure feed-forward control was 0.3857 rad and in the case of feedforward plus feedback control is 0.2068 rad. Similarly, for joint 2 we observe that RMS for the pure feed-forward case was 0.1701 rad and for feedback control was 0.1341 rad. So, the use of feedback gains reduces the error and helps to stabilize the upward-pointing position in both cases. Similarly, an underactuated 4DoF system can also be stabilized in the vertical position since the proposed controller uses state-feedback controller (and given the system reachable in the vertical equilibrium).

Fig 14 illustrates the motion synthesized by the proposed algorithm for an underactuated serial manipulator with 21 joints and only one actuated joint. The task presented is an end-effector regulation task with the desired end-effector position being: $[2.12, 2.12]$ m and the terminal velocity being: $[7.07, 0]$ m/s with the error in simulation being 0.006 m. In Fig. 15, we present the end-effector motion and the input torques.

Thus, the proposed method is successful in planning optimal trajectories for under-actuated compliant systems as well. Especially the simulation results with an under-actuated serial manipulator with 20 passive joints are promising as the error in the end-effector position is only 0.006 m.

The results presented here neglect the exact nonlinear actuator dynamics, and thus the tracking performance on the experimental setup illustrates the robustness of the method.

D. Energy consumption

The sum of torque squared over the whole trajectory is assumed to be the most suitable candidate to compare different controllers for power consumption [44] i.e., $T = \sum_{k=0}^{N} \tau_k^2$. Using this metric, we conduct simulations and show that elastic actuation reduces energy consumption. To illustrate this we use end effector regulation task for a fully actuated 2DoF system, a fully actuated 7DoF and a 2DoF underactuated compliant arm. For each of the systems, the cost weights are kept the same across rigid, SEA and VSA actuation for a fair comparison.

Table V shows that use of SEA and VSA lowers the energy consumption for all the three systems.

TABLE V
POWER CONSUMPTION COMPARISON BETWEEN RIGID AND SOFT ACTUATORS

Problems	rigid	SEA	VSA
2DoF	142.07	138.46	84.17
2DoF Flexible	101.019	87.53	50.84
7DoF	> 10000	6112.77	4545.26

VII. CONCLUSION AND FUTURE WORK

In this work, we presented an efficient optimal control formulation for soft robots based on the Box-FDDP/FDDP algorithms. We proposed an efficient way to compute the dynamics and analytical derivatives of soft articulated and underactuated compliant robots. The state-feedback controller presented in this paper based on local and optimal policies from Box-FDDP/FDDP helped to improve the performance in swing-up and end-effector regulation tasks. Overall, the application of (high authority) feedback to soft robots may be an advantage or a disadvantage [1]. For instance, in [45] it is shown how feedback can stabilize unstable equilibrium points of the system. Furthermore, feedback increases the robustness of model uncertainties and disturbances. However, the negative effect of feedback is the alteration of the mechanical stiffness of the system [43], [46] which defeats the purpose of building soft robots. In fact, compliance is purposefully inserted into robots to confer them the so-called *embodied intelligence* [47]. This derives from the interaction between the system body, its sensory-motor control, and the environment, and it has the goal to simplify the robot control thoughtfully inserting complexity, and intelligence, into the robot body [48]–[50]. For this reason, depending on the specific task, the application of feedback to soft robots should be carefully approached. Future work will focus on preserving the natural behavior of the controlled soft robot including limitations to the feedback authority in the control problem similar to [51].

Another future research direction would consist of extending the formalism to soft articulated legged robots. Accounting for compliance in legged robots is expected to improve the performance compared to existing approaches. The use of feedback alters compliance in the system. This behavior is undesirable as it defies the sole purpose of adding the soft elements in the first place. One possible research direction would be to explore a DDP-based algorithm, maybe in a bi-level setting, to obtain feedback gains that respects system compliance and still get high performance. Further, an MPC solution based on the proposed framework is seen as a natural extension of this work.

REFERENCES

[1] Cosimo Della Santina, Manuel G Catalano, and Antonio Bicchi. Soft robots. *Encyclopedia of Robotics*, 489, 2020.

[2] Bram Vanderborght, Björn Verrelst, Ronald Van Ham, Michaël Van Damme, Dirk Lefeber, Bruno Meira Y Duran, and Pieter Beyl. Exploiting Natural Dynamics to Reduce Energy Consumption by Controlling the Compliance of Soft Actuators. *The Int. J. of Rob. Res. (IJRR)*, 25:343–358, 2006.
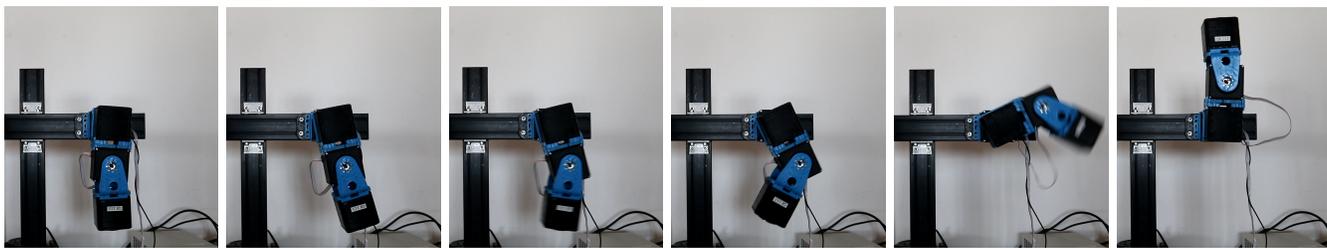
Fig. 11. Photo-sequence of the swing-up task for 2DoF underactuated compliant arm with a SEA in the first joint. Please refer to the video attachment for more details.

(a) Link positions.

(b) Input torque of first joint.

(c) Links position.
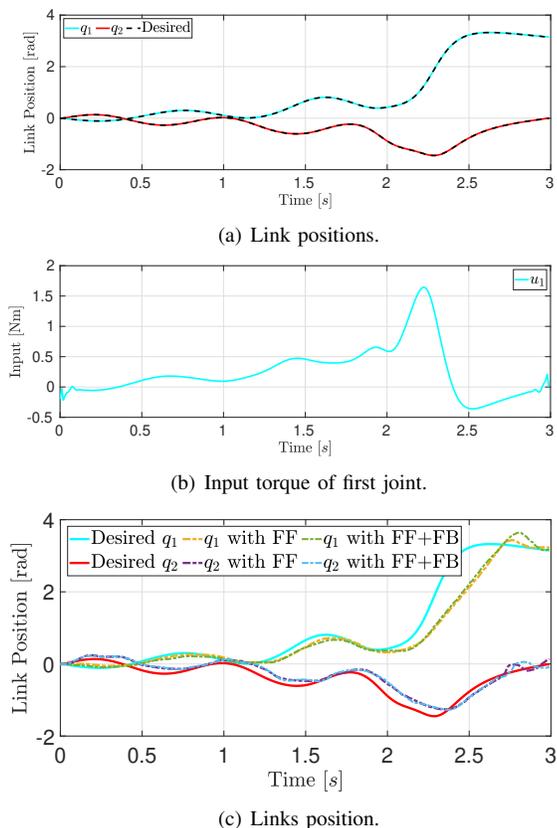
Fig. 12. Swing-up task for the 2DoF underactuated compliant robot with a SEA in the first joint. (a) Joint evolution in simulation. (b) Input torque of the first joint in simulation. (c) Evolution of joint 1 and joint 2 in experiments. We compare the desired and the link positions using both the pure feed-foward (FF) and the feedback plus feed-forward (FF+FB) cases, which shows better performance in the latter.

(a) Input torques

(b) Input torques

(c) Stiffness profile

(d) Links positions

Fig. 13. Swingup task for the 2DoF underactuated compliant arm with a VSA in the first joint. (a) Input torque evolution in simulation. (b) Input stiffness evolution in simulation. (c) Evolution of joint 1 and joint 2 in experiments. With pure feed-foward (FF) and the feedback + feed-forward(FF+FB) cases, we compare the desired and the link positions, which shows better performance in the later.

[3] Thomas J Roberts and Emanuel Azizi. Flexible mechanisms: the diverse roles of biological springs in vertebrate movement. *J. of Exp. Bio.*, 214:353–361, 2011.

[4] Handdeut Chang, Sangjoon J Kim, and Jung Kim. Feedforward motion control with a variable stiffness actuator inspired by muscle cross-bridge kinematics. *IEEE Trans. Robot. (T-RO)*, 35:747–760, 2019.

[5] Gian Maria Gasparri, Silvia Manara, Danilo Caporale, Giuseppe Averta, Manuel Bonilla, Hamal Marino, Manuel Catalano, Giorgio Grioli, Matteo Bianchi, Antonio Bicchi, et al. Efficient walking gait generation via principal component representation of optimal trajectories: application to a planar biped robot with elastic joints. *IEEE Robot. Automat. Lett. (RA-L)*, 3:2299–2306, 2018.

[6] Alexander Werner, Bernd Henze, Florian Loeffl, Sigrid Leyendecker, and Christian Ott. Optimal and robust walking using intrinsic properties of a series-elastic robot. In *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2017.

[7] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 1995.

Fig. 14. Photo sequence of motion of end-effector regulation task on underactuated serial manipulator. Red ball is the desired end-effector position with coordinates as [2.12, 2.12] m.

(a) Link positions
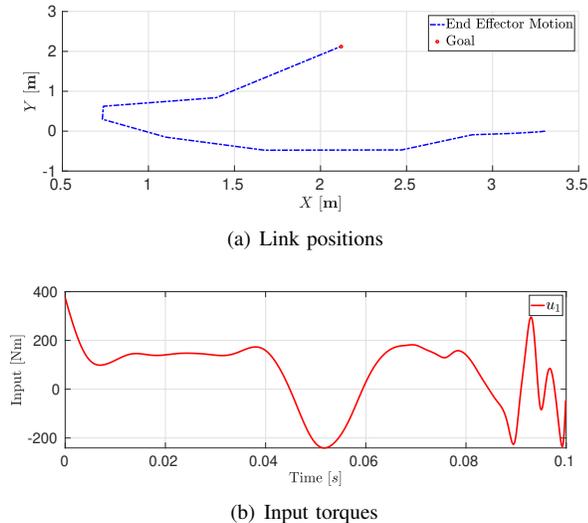
(b) Input torques

Fig. 15. End-effector regulation task of a underactuated serial manipulator. (a) Evolution of the link positions in simulation. (b) Input torque evolution in simulation.

[8] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, Manuel Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. In *Rob.: Sci. Sys. (RSS)*, 2013.

[9] Michele Pierallini, Franco Angelini, Riccardo Mengacci, Alessandro Palleschi, Antonio Bicchi, and Manolo Garabini. Iterative learning control for compliant underactuated arms. 2023.

[10] Mostafa Sayahkarajy, Z Mohamed, and Ahmad Athif Mohd Faudzi. Review of modelling and control of flexible-link manipulators. *J. of Sys. and Cnt. Eng.*, 230:861–873, 2016.

[11] Venkatasubramanian Kalpathy Venkiteswaran, Jakub Sikorski, and Sarthak Misra. Shape and contact force estimation of continuum manipulators using pseudo rigid body models. *Mechanism and Machine Theory*, 139:34–45, 2019.

[12] Mahta Khoshnam and Rajni V. Patel. A pseudo-rigid-body 3r model for a steerable ablation catheter. In *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2013.

[13] Deepak Trivedi, Amir Lotfi, and Christopher D. Rahn. Geometrically Exact Models for Soft Robotic Manipulators. *IEEE Trans. Robot. (T-RO)*, 24:773–780, 2008.

[14] III Robert J. Webster and Bryan A. Jones. Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review. *The Int. J. of Rob. Res. (IJRR)*, 29:1661–1683, 2010.

[15] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control. In *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2020.

[16] Rohan Budhiraja, Justin Carpentier, Carlos Mastalli, and Nicolas Mansard. Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics. In *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2018.

[17] David J. Braun, Matthew J. Howard, and Sethu Vijayakumar. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Auton. Robots.*, 33:237–253, 2012.

[18] Chen Ji, Minxiu Kong, and Ruifeng Li. Time-Energy Optimal Trajectory Planning for Variable Stiffness Actuated Robot. *IEEE Access*, 7:14366–14377, 2019.

[19] Michele Pierallini, Franco Angelini, Riccardo Mengacci, Alessandro Palleschi, Antonio Bicchi, and Manolo Garabini. Trajectory Tracking of a One-Link Flexible Arm via Iterative Learning Control. In *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2020.

[20] Manolo Garabini, Andrea Passaglia, Felipe Belo, Paolo Salaris, and Antonio Bicchi. Optimality principles in stiffness control: The VSA kick. In *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2012.

[21] Sami Haddadin, Felix Huber, and Alin Albu-Schäffer. Optimal control for exploiting the natural dynamics of Variable Stiffness robots. In *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2012.

[22] Andrea Cristofaro, Alessandro De Luca, and Leonardo Lanari. Linear-quadratic optimal boundary control of a one-link flexible arm. *IEEE Cntrl. Sys. Lett.*, 5:833–838, 2021.

[23] Alessandro Palleschi, Riccardo Mengacci, Franco Angelini, Danilo Caporale, Lucia Pallottino, Alessandro De Luca, and Manolo Garabini. Time-Optimal Trajectory Planning for Flexible Joint Robots. *IEEE Robot. Automat. Lett. (RA-L)*, 5:938–945, 2020.

[24] Luca Consolini, Oscar Gerelli, Corrado Guarino Lo Bianco, and Aurelio Piazzi. Minimum-time control of flexible joints with input and output constraints. In *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2007.

[25] Altay Zhakatayev, Matteo Rubagotti, and Huseyin Atakan Varol. Time-Optimal Control of Variable-Stiffness-Actuated Systems. *IEEE/ASME Trans. Mechtr.*, 22:1247–1258, 2017.

[26] Altay Zhakatayev, Matteo Rubagotti, and Huseyin Atakan Varol. Energy-Aware Optimal Control of Variable Stiffness Actuated Robots. *IEEE Robot. Automat. Lett. (RA-L)*, 4:330–337, 2019.

[27] D. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 1966.

[28] Carlos Mastalli, Wolfgang Merkt, Guiyang Xin, Jaehyun Shim, Michael Mistry, Ioannis Havoutis, and Sethu Vijayakumar. Agile Maneuvers in Legged Robots: a Predictive Control Approach, 2022.

[29] Carlos Mastalli, Wolfgang Merkt, J. Marti-Saumell, Henrique Ferrolho, Joan Sola, Nicolas Mansard, and Sethu Vijayakumar. A Feasibility-Driven Approach to Control-Limited DDP. *Auton. Robots.*, 2023.

[30] Jun Nakanishi, Andreea Radulescu, David J. Braun, and Sethu Vijayakumar. Spatio-temporal stiffness optimization with switching dynamics. *Auton. Robots.*, 41:273–291, 2017.

[31] Justin Carpentier and Nicolas Mansard. Analytical derivatives of rigid body dynamics algorithms. In *Rob.: Sci. Sys. (RSS)*, 2018.

[32] Alin Albu-Schäffer and Antonio Bicchi. Actuators for soft robotics. In *Springer Handbook of Robotics*, pages 499–530. Springer, 2016.

[33] M Poloni and G Ulivi. Iterative learning control of a one-link flexible manipulator. In *Robot Control 1991*, pages 393–398. Elsevier, 1992.

[34] Moritz A. Graule, Clark B. Teeple, Thomas P. McCarthy, Grace R. Kim, Randall C. St. Louis, and Robert J. Wood. SoMo: Fast and Accurate Simulations of Continuum Robots in Complex Environments. In *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2021.

[35] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[36] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE Int. Sym. Sys. Integ. (SII)*, 2019.

[37] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2014.

[38] Alexandra Velasco, Manolo Garabini, Manuel G. Catalano, and Antonio Bicchi. Soft actuation in cyclic motions: Stiffness profile optimization for energy efficiency. In *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2015.

[39] Riccardo Mengacci, Manolo Garabini, Giorgio Grioli, Manuel G Catalano, and Antonio Bicchi. Overcoming the torque/stiffness range tradeoff in antagonistic variable stiffness actuators. *IEEE/ASME Trans. Mechtr.*, 26:3186–3197, 2021.

[40] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux, J.-P. Laumond, L. Marchionni, H. Tome, and F. Ferro. Talos: A new humanoid research platform targeted for industrial applications. In *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2017.

[41] Carlos Mastalli, Saroj Prasad Chhatoi, Thomas Corbères, Steve Tonneau, and Sethu Vijayakumar. Inverse-dynamics mpc via nullspace resolution. *IEEE Trans. Robot. (T-RO)*, 2023.

[42] Riccardo Mengacci, Franco Angelini, Manuel G Catalano, Giorgio Grioli, Antonio Bicchi, and Manolo Garabini. On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control. *The Int. J. of Rob. Res. (IJRR)*, 40:348–374, 2021.

[43] Franco Angelini, Cosimo Della Santina, Manolo Garabini, Matteo Bianchi, Gian Maria Gasparri, Giorgio Grioli, Manuel Giuseppe Catalano, and Antonio Bicchi. Decentralized trajectory tracking control for soft robots interacting with the environment. *IEEE Trans. Robot. (T-RO)*, 34:924–935, 2018.

[44] Mitsunori Uemura and Sadao Kawamura. An energy saving control method of robot motions based on adaptive stiffness optimization - cases of multi-frequency components -. In *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2008.

[45] P. Tomei. A simple pd controller for robots with elastic joints. *IEEE Trans. Robot. (T-RO)*, 36:1208–1213, 1991.

[46] Cosimo Della Santina, Matteo Bianchi, Giorgio Grioli, Franco Angelini, Manuel Catalano, Manolo Garabini, and Antonio Bicchi. Controlling soft robots: balancing feedback and feedforward elements. *IEEE Rob. and Automat. Magaz.*, 24:75–83, 2017.

[47] Gianmarco Mengaldo, Federico Renda, Steven L Brunton, Moritz Bächer, Marcello Calisti, Christian Duriez, Gregory S Chirikjian, and Cecilia Laschi. A concise guide to modelling the physics of embodied intelligence in soft robotics. *Nature Reviews Physics*, 4:595–610, 2022.

[48] Rolf Pfeifer and Gabriel Gómez. Morphological computation–connecting brain, body, and environment. In *Creating brain-like intelligence*. 2009.

[49] Fumio Hara and Rolf Pfeifer. *Morpho-functional machines: The new species: Designing embodied intelligence*. Springer Science & Business Media, 2003.

[50] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[51] Michele Pierallini, Franco Angelini, Antonio Bicchi, and Manolo Garabini. Swing-Up of Underactuated Compliant Arms Via Iterative Learning Control. *IEEE Robot. Automat. Lett. (RA-L)*, 7:3186–3193, 2022.

Carlos Mastalli received an M.Sc. degree in mechatronics engineering from the Simón Bolívar University, Caracas, Venezuela, in 2013 and a Ph.D. degree in bio-engineering and robotics from the Istituto Italiano di Tecnologia, Genoa, Italy, in 2017.

He is currently an Assistant Professor at Heriot-Watt University, Edinburgh, U.K. He is the Head of the Robot Motor Intelligence (RoMI) Lab affiliated with the National Robotarium and Edinburgh Centre for Robotics. He is also appointed as Research Scientist at IHMC, USA. Previously, he conducted cutting-edge research in several world-leading labs: Istituto Italiano di Tecnologia (Italy), LAAS-CNRS (France), ETH Zürich (Switzerland), and the University of Edinburgh (UK). His research focuses on building athletic intelligence for robots with legs and arms. Carlos' research work is at the intersection of model predictive control, numerical optimization, machine learning, and robot co-design.

Saroj Prasad Chhatoi received his M.Sc. degree in physics from the Institute of Mathematical Sciences, Chennai, India in 2019.

He is currently pursuing a Ph.D. degree in Automatic Control at LAAS-CNRS, Toulouse, France. His research interests include Automatic Control, legged locomotion, control of soft robotic systems, model predictive control.

Michele Pierallini received the B.S. degree in biomedical engineering in 2017 and M.S. degree (cum laude) in automation and robotics engineering in 2020 from the University of Pisa, Pisa, Italy, where he is currently working toward the Ph.D. degree in robotics at the Research Center "Enrico Piaggio." His current research focuses on the control of soft robotic systems and iterative learning control.

Manolo Garabini graduated in Mechanical Engineering and received the Ph.D. in Robotics from the University of Pisa where he is currently employed as Assistant Professor. His main research interests include the design, planning, and control of soft adaptive robots. He contributed to the realization of modular Variable Stiffness Actuators, to the design of the humanoid robot WALK-MAN and, more recently, to the development of efficient and effective compliance planning algorithms for interaction under uncertainties. Currently, he is the Principal Investigator in the THING H2020 EU Research Project for the University of Pisa, and the coordinator of the Dysturbance H2020 Eurobench sub-project. Finally, he is the coordinator of the H2020 EU Research Project Natural Intelligence.

Franco Angelini received the B.S. degree in computer engineering in 2013 and M.S. degree (cum laude) in automation and robotics engineering in 2016 from the University of Pisa, Pisa, Italy. University of Pisa granted him also a Ph.D. degree (cum laude) in robotics in 2020. Franco currently has a research fellowship at the Research Center "Enrico Piaggio", Pisa. His main research interests are control of soft robotic systems, grasping and impedance planning.