Optimal Control for Articulated Soft Robots

Saroj Prasad Chhatoi[1,5] 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, controllers that can 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 robots by using numerical differentiation and fully actuated robots. 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. The proposed method can plan and control underactuated compliant robots, with varying degrees of underactuation, effectively.**

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], [3].

A series elastic actuator (SEA) has a linear spring between the actuator and the load [4]. Instead, a variable stiffness actuator (VSA) integrates an elastic element that can be adjusted mechanically. These actuators provide many potential advantages but also increase the control complexity [5].

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 [6], [7].

We propose an efficient optimal control method for articulated soft robots based on the feasibility-driven DDP (FDDP)/Box-FDDP algorithm. It boils down to three technical contributions [3]: (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. More details can be found here [3] [1].

This research is partially supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101016970 (Natural Intelligence), No. 101017274 (DARKO), and No. 780684 (Memory of Motion), in part by the Ministry of University and Research (MUR) as part of the PON 2014-2021 "Research and Innovation" resources – Green Action - DM MUR 1062/2021, and in part by the Italian Ministry of Education and Research in the framework of the "FoReLab" (Future-oriented Research Lab) Project (Departments of Excellence). (*Corresponding Author: Saroj Prasad Chhatoi)*

[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]IHMC, Pensacola, US

[5]LAAS-CNRS, Toulouse, France

II. Problem Definition

We report a model for soft articulated robotic arms, i.e.,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{K}(\mathbf{q} - \mathbf{S}\boldsymbol{\theta}) = \mathbf{0}, \quad (1)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{S}^\top\mathbf{K}(\mathbf{S}\boldsymbol{\theta} - \mathbf{q}) - \boldsymbol{\tau} = \mathbf{0}, \quad (2)$$

where, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ contains the Coriolis terms, and $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity term, $\mathbf{B} \in \mathbb{R}^{m\times m}$ is the motor inertia, $\mathbf{U}(\mathbf{q},\boldsymbol{\theta})$ is the elastic potential, $\boldsymbol{\tau} \in \mathbb{R}^m$ is the torque, and $\mathbf{S} \in \mathbb{R}^{n\times m}$ is the selection matrix.

A. Optimal control formulation

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

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

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; $\mathrm{FD}(\cdot)$ represents the forward dynamics of the soft robot; $\mathscr{Q}$ represents the admissible state space; $\mathscr{V}$ describes the admissible velocity space and $\mathscr{U}$ defines the allowed control. it is instrumental for the method to define $\mathbf{x}$ 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.

[1]https://github.com/michelepierallini/aslr_to

III. Problem Solution

We solve the optimal control problem described in Section II-A using the Box-FDDP algorithm.

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 backward in time.

The Bellman relation is stated as

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k} \ell_k(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)), \tag{3}$$

where, $V(\mathbf{x}_k)$ is the value function at the node k. FDDP uses a quadratic approximation of the differential change in (3), i.e., $\mathbf{Q}$. and it is divided into two parts.

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

$$\begin{aligned} \delta\mathbf{u}_k =& \arg\min_{\delta\mathbf{u}_k} \mathbf{Q}(\delta\mathbf{x}_k, \delta\mathbf{u}_k) = \hat{\mathbf{k}} + \hat{\mathbf{K}}\delta\mathbf{x}_k, \\ & \text{s.t.} \quad \underline{\mathbf{u}} \leq \mathbf{u}_k + \delta\mathbf{u}_k \leq \overline{\mathbf{u}}, \end{aligned} \tag{4}$$

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 (4), 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

$$\hat{\mathbf{u}}_k = \mathbf{u}_k + \alpha\hat{\mathbf{k}} + \hat{\mathbf{K}}(\hat{\mathbf{x}}_k - \mathbf{x}_k), \tag{5}$$

$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) - (1-\alpha)\bar{\mathbf{f}}_{k-1}, \tag{6}$$

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 [6]. The interested reader is referred to [3], [6], [8] for more details about the algorithm.

A. Dynamics for soft robots

The forward dynamics computation in (1)-(2) can be done differentiating $\boldsymbol{\tau}_l$ and $\boldsymbol{\tau}_m$

$$\boldsymbol{\tau}_l \triangleq -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q}) - \mathbf{K}(\mathbf{q} - \mathbf{S}\boldsymbol{\theta}), \tag{7}$$

$$\boldsymbol{\tau}_m \triangleq -\mathbf{S}^\top\mathbf{K}(\mathbf{S}\boldsymbol{\theta} - \mathbf{q}) + \boldsymbol{\tau}. \tag{8}$$

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

$$\begin{bmatrix} \delta\ddot{\mathbf{q}} \\ \delta\ddot{\boldsymbol{\theta}} \end{bmatrix} = -\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}^{-1} \left(\begin{bmatrix} \frac{\partial\boldsymbol{\tau}_l}{\partial\mathbf{x}} \\ \frac{\partial\boldsymbol{\tau}_m}{\partial\mathbf{x}} \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \frac{\partial\boldsymbol{\tau}_l}{\partial\mathbf{u}} \\ \frac{\partial\boldsymbol{\tau}_m}{\partial\mathbf{u}} \end{bmatrix} \delta\mathbf{u} \right). \tag{9}$$

IV. Validation

We employ a 2DoF compliant system where the first joint is actuated by a SEA and the second elastic joint is punctuated.

The motors and the links of theactuator both include AS5045 12 bit magnetic encoders. The actuator's elastic torque τ and nonlinear stiffness function σ satisfy the following equation $\tau = 2\beta\cosh\alpha\theta_s\sinh\alpha(q-\theta_e)$ and $\sigma = 2\alpha\beta\cosh(\alpha\theta_s)\cosh\alpha(q-\theta_e)$ where, $\alpha = 6.7328$ rad^{-2}, $\beta = 0.0222$ Nm, θ_e is the motor equilibrium position, θ_s tunes the desired motor stiffness and q is the link-side position.

a) 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. The weights corresponding to control regularization is 10^{-1}, state regularization is 10^{-2}, ℓ_k is 10^{-2} and the goal-tracking cost is 10^{-1}. The stiffness of the second link is 2 Nm/rad.

Fig. 2 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. 2(a)) and the input sequence (Fig. 2(b)). Fig. 2(c) illustrates the link positions of both the joints obtained from the experiments. Snapshots of the experiments are depicted in Fig. 1. 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, RMS for the pure feed-forward case was 0.1607 rad and for feedforward plus feedback control was 0.1571 rad.

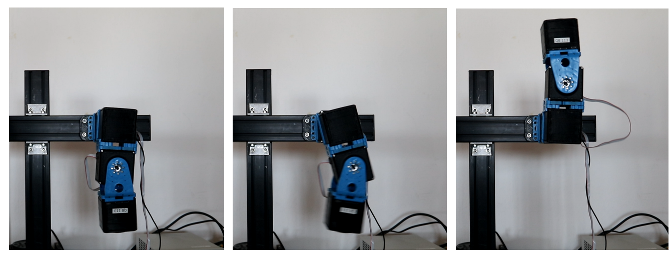
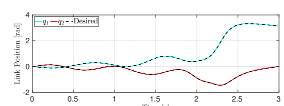
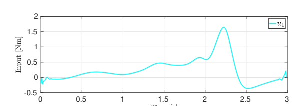


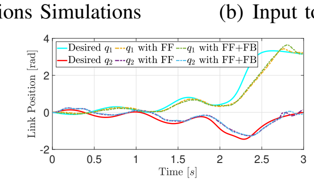
Fig. 1. Photo-sequence of the swing-up task for 2DoF underactuated compliant arm with a SEA in the first joint.



(a) Link positions Simulations



(b) Input torque of first joint



(c) Links position

Fig. 2. Swing-up task for the 2DoF underactuated compliant robot with a SEA in the first joint. We compare the desired and the link positions using both the pure feed-forward (FF) and the feedback plus feed-forward (FF+FB) cases, which shows better performance in the latter.

V. Conclusion

In this work, we proposed 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. Finally, the approach's effectiveness is tested on real hardware varying tasks. Further, an MPC solution based on the proposed framework is a natural extension of this work.

References

[1] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft robots," *Encyclopedia of Robotics*, vol. 489, 2020.

[2] B. Vanderborght, B. Verrelst, R. V. Ham, M. V. Damme, D. Lefeber, B. M. Y. Duran, and P. Beyl, "Exploiting Natural Dynamics to Reduce Energy Consumption by Controlling the Compliance of Soft Actuators," *The Int. J. of Rob. Res. (IJRR)*, vol. 25, pp. 343–358, 2006.
[3] S. P. Chhatoi, M. Pierallini, F. Angelini, C. Mastalli, and M. Garabini, "Optimal control for articulated soft robots," *IEEE Transactions on Robotics*, 2023.
[4] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 1995.
[5] B. Vanderborght, A. Albu-Schäffer, A. Bicchi, E. Burdet, D. G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh *et al.*, "Variable impedance actuators: A review," in *Rob.: Sci. Sys. (RSS)*, 2013.
[6] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2020.
[7] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics," in *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2018.
[8] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Sola, N. Mansard, and S. Vijayakumar, "A Feasibility-Driven Approach to Control-Limited DDP," *Auton. Robots.*, 2023.