Sensorless Proprioception in Multi-DoF Dielectric Elastomer Soft Robots via System-Level Self-Sensing

Johannes Prechtl, Matthias Baltes, Kathrin Flaßkamp, and Gianluca Rizzello, *Senior Member, IEEE*

***Abstract*—Proprioception in soft robots remains an ongoing challenge, owing to practical issues such as tight sensor integration while maintaining mechanical compliance. Soft robots based on dielectric elastomer (DE) technology could provide a compelling answer to these challenges, due to the inherent flexibility and compliance of such transducers as well as their ability to simultaneously work as actuators and sensors, i.e., self-sensing. In this work, we propose a novel real-time self-sensing scheme for DE soft robotic systems. By combining an actuator-level recursive least squares identification with an extended Kalman filter, our architecture provides an estimation of the mechanical state of the structure without requiring additional electromechanical sensors, solely relying on electrical measurements performed on the DEs during high voltage actuation. Experimental validation, conducted on a DE soft robot prototype, reveals that the proposed solution reconstructs the system state during actuation in a robust and accurate way, and under various external loading conditions.**

***Index Terms*—Dielectric elastomers (DE), extended Kalman filter, proprioception, self-sensing, sensorless estimation, soft robots.**

I. Introduction

THE field of soft robotics has attracted significant interest from researchers worldwide [1]. Compared to conventional rigid robots, the main defining characteristic of soft robots is their mechanical compliance, which enables potential applications in fields such as human–robot interaction, inspection tasks, or rescue operations in confined spaces [2], [3]. The low mechanical stiffness of soft robots, however, poses significant challenges for the sensing and control of their shape [4].

Manuscript received 26 July 2023; revised 23 November 2023 and 24 January 2024; accepted 5 March 2024. Recommended by Technical Editor M. Rakotondrabe and Senior Editor Y. Li. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through Priority Program SPP 2100 "Soft Material Robotic Systems" under Project SE704/6-2 and Project RI3030/1-2. *(Corresponding author: Johannes Prechtl.)*

The authors are with the Department of Systems Engineering, Saarland University, 66121 Saarbrücken, Germany (e-mail: johannes.prechtl@imsl.uni-saarland.de).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2024.3375923.

Digital Object Identifier 10.1109/TMECH.2024.3375923

Several survey papers have recently emphasized the need for soft robot proprioception, i.e., the capability of a soft robot to sense its own shape in real time, so as to enable closed-loop motion control [4], [5], [6], [7]. It is remarked, however, that achieving this goal still remains an open challenge in soft robotics [8]. The main challenge of shape sensing for soft robots arises from the fact that a large number (up to infinity) of degrees of freedom (DoF) must be estimated from a limited number of measurements [4]. To practically address this issue, a possible solution consists of using a large number of sensors distributed over the robot body. As an example, the local curvature of long and slender robots can be reconstructed using strain gauges, and the overall shape estimated by means of interpolation [9], [10]. A disadvantage of this method is the larger number of measured signals, which require electrical connections, dedicated electronics, and significant processing power. Alternatively, model-based or machine learning approaches can be used to infer missing information based on a restricted number of measurements. In [11], a state observer relying on an optical curvature sensor was used for shape estimation but, due to the adoption of a simplified modeling approach, the method only provided a steady-state result. Shu et al. [12] used measurements from flexible resistive sensors as input to a neural network for posture perception. The approach was effective in avoiding complex modeling efforts for the highly nonlinear and hysteretic sensor behavior, but had to rely on a comparatively large number of feedback variables to estimate a single bending angle. In general, using external sensors leads to an increase in system stiffness and overall size, due to the lack of tight integration between sensors and actuators. In some cases, like in cable-driven soft robots, the sensing feature can be directly integrated within the same component performing actuation [13]. For instance, in [14] the configuration and tension of a cable-driven soft robot were estimated via the actuator lengths, and reconstructed through motor-level sensors. The approach produces accurate sensing, but requires detailed nonlinear modeling and calibration of friction and hysteresis, as well as a custom and rather complex real-time numerical solving strategy.

All methods discussed so far are either sensor-based, and thus, agnostic to the actuation mechanism, or specifically tailored to pneumatically- and cable-driven soft robots. These mechanisms allow for soft and compact designs, but lack intrinsic self-sensing features and have to rely on external hardware such as

cable-driving motors or air compressors. Some of those shortcomings were addressed in [15], where the authors utilized the actuation fluid as an integrated resistive sensor, or in [16], where currently unused actuators can serve as sensors. Those approaches rely either on additional wiring or on alternating between actuation and sensing functions, thus, leaving room for technologies offering even tighter integration. To this end, dielectric elastomer (DE) transducers may represent an interesting alternative technology for soft robotics [17], [18], [19], [20]. A DE actuator (DEA) consists of a polymeric membrane (e.g., silicone) sandwiched between compliant electrodes, whose in-plane expansion can be controlled via an external voltage signal. Other than exhibiting high stretchability, inherent softness, smooth controllability, and ease of integration, DEs can also work as *artificial muscles* and *artificial nerves* at the same time, exploiting a so-called self-sensing operating mode. More specifically, changes in DEA electrical capacitance can be detected in real time during high voltage actuation, and used to reconstruct the deformation [21], [22] and/or the force [23] of the transducer, thus, allowing to implement *sensorless* position [24], [25], [26] and compliance [23] control loops. The term sensorless, widely used in the literature on electric motors, here refers to architectures in which mechanical quantities are estimated without including additional electro-mechanical sensors into the system, but instead relying on available electrical measurements (e.g., voltage, current) performed on already existing transducers (i.e., the actuators themselves). Similar techniques have been applied to other smart transducers such as piezoelectric positioning devices [27], [28], [29] to obtain real-time estimation of displacement, force, and system state. It is remarked that the existing DE literature solely focused on *actuator-level self-sensing*, i.e., techniques to reconstruct the deformation and/or force of a simple, single DoF DE component. In case of multi-DoF structures actuated by several DE transducers, a more advanced concept referred to as *system-level self-sensing* can be implemented, in which different actuator-level estimations are combined together to reconstruct some global system information. In case of DE soft robots, this approach allows reconstructing the full configuration of the system during an actuation task based on DEA voltage and current signals only, without the need to embed extra mechanical or electrical sensors in the structure. This concept will open up the possibility of implementing integrated proprioception and sensorless closed loop control for DE soft robots. To the authors' best knowledge, no previous work has investigated similar concepts for DEA-driven multi-DoF systems. Factors which contribute to making this problem challenging, compared to the single-DoF case, include underactuation as well as several system nonlinearities.

In this work, we introduce a system-level self-sensing scheme for a DE-driven soft robotic system. The case study consists of a planar module driven by a pair of pretensioned rolled DEAs, capable of bending and translation motions. The proposed self-sensing architecture relies on the combination of two functional blocks, i.e., an actuator-level estimation scheme and a system-level state observer. The former estimates in real time the electrical capacitance of the two actuators via recursive least squares, and uses this information to reconstruct their lengths, based on measurements of DEA voltage and current signals. Since the underactuated and nonlinear nature of our system makes it not possible to uniquely reconstruct the robot configuration solely via DEA lengths information, a second functional block consisting of an extended Kalman filter (EKF) is introduced. The EKF receives as inputs actuator lengths as well as the actuation voltages, and provides an estimation of the system mechanical configuration and generalized velocity while also accounting for model nonlinearities. After presenting the novel architecture, an extensive experimental validation campaign is conducted, in which the estimated configuration is compared with measurements acquired with a camera system. It is shown that the robot configuration can be estimated precisely for a variety of input trajectories with sub-mm/sub-degree accuracy. In addition, the system-level self-sensing scheme permits not only to estimate changes in the robot configuration which are caused by actuation voltage but also by unmodeled external loads. To the authors' best knowledge, this is the first work demonstrating full system-level self-sensing in a multi-DoF DEA soft robot. We also point out that this article is an extension of [30], where we proved feasibility of system-level self-sensing by successfully estimating the system configuration based on optically measured DEA lengths. Here, we extend the preliminary results of [30] by including: the first experimental demonstration of actual system-level self-sensing, which solely requires DEA voltage and current measurements; an extensive experimental campaign with various actuation signals and external mechanical loads; an investigation of the algorithm real-time feasibility; a new reduced-order estimator formulation, which lowers the real-time computational burden without compromising accuracy.

The rest of this article is organized as follows. Section II begins with an explanation of the system operating principle. Then, a dynamic model is formulated in Section III, followed by the presentation of the system-level self-sensing in Section IV. Experimental results are discussed in Section V. Finally, Section VI concludes this article.

II. DE Soft Robotic System

The soft robotic system considered in this work was initially presented in [31], and is shown in Fig. 1(a). The structure, denoted as T-Platform, consists of a flexible backbone put under compressive load by two pretensioned rolled DEAs. The DEs used in this work are made of silicone material (Wacker Elastosil 2030, 50 μm thickness) and flexible carbon-based electrodes (manufactured via screen printing). The rolled actuators are obtained by stacking together two single-layer DEs and winding them together into a spiral-like structure, as shown in Fig. 2(a). When one pretensioned rolled DEA is subjected to high voltages, attractive electrostatic forces between the charged electrodes induce a thickness reduction and a subsequent axial elongation, as in Fig. 2(a). This expansion, in turn, causes the T-Platform to bend toward the unactuated DEA. As demonstrated in our previous work [31], if the compressive loads exerted by the soft actuators onto the flexible backbone is sufficiently high, the DEA electrical activation triggers a buckling instability which causes the maximum bending angle to increase above 20°, and also

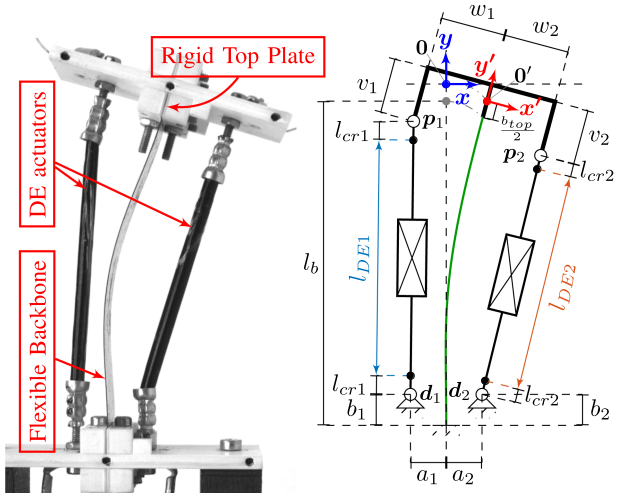

Fig. 1. (a) Physical T-Platform prototype and (b) kinematic model sketch.

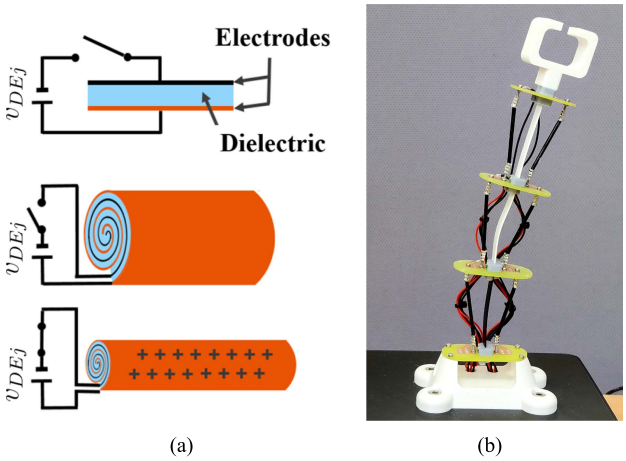

Fig. 2. (a) RDEA operating principle, and (b) soft robot prototype actuated by rolled DEAs.

makes the overall system bistable. An example of modular soft robotic arm, obtained by stacking three T-Platforms, is reported in Fig. 2(b).

III. System Modeling

In this section, kinematic and dynamic modeling of the system introduced in Section II are derived. The model describes the relationship between applied voltage and resulting T-Platform motion. Then, a suitable simplification is proposed to reduce the model computational burden.

A. Kinematic Model

The kinematic model defines the relationship between actuator lengths and the overall T-Platform pose, described by its configuration variables. From a modeling perspective, the relevant system components are the two rolled DEAs, the flexible bending element in the center, and the rigid support structure connecting those parts. The rolled DEAs are modeled as axially-expanding 1-D elements interconnected to the structure via revolute joints, to represent bending at the actuators end-point. The flexible bending element is assumed not to constrain the number of DoF of the structure, and will only play a role when considering the involved forces.

Similarly to our previous works [31], we assume that the configuration of the structure is uniquely determined once we fix the configuration of the rigid body describing the rigid top plate. Therefore, we select as generalized coordinates $\boldsymbol{q}$ the quantities describing the in-plane configuration of such body

$$\boldsymbol{q} = [q_x \quad q_y \quad \alpha]^\intercal \tag{1}$$

where q_x, q_y, and α represent the Cartesian coordinates and the angular displacement of the rigid body frame attached to the top place center of mass $\mathbf{0}' - \boldsymbol{x}'\boldsymbol{y}'$ relative to the static reference frame $\mathbf{0} - \boldsymbol{x}\boldsymbol{y}$; see Fig. 1(b). Next, based on Fig. 1(b), we express the DEAs attachment points on the lower part $\boldsymbol{d}_1$, $\boldsymbol{d}_2$ and on the upper part $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ of the structure as

$$\begin{aligned}
\boldsymbol{d}_1 &= \begin{bmatrix} -a_1 \\ b_1 - l_b - \frac{b_{\text{top}}}{2} \end{bmatrix}, \quad \boldsymbol{d}_2 = \begin{bmatrix} a_2 \\ b_2 - l_b - \frac{b_{\text{top}}}{2} \end{bmatrix}, \\
\boldsymbol{p}_1(\boldsymbol{q}) &= \begin{bmatrix} q_x \\ q_y \end{bmatrix} + \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} -w_1 \\ \frac{b_{\text{top}}}{2} - v_1 \end{bmatrix}, \\
\boldsymbol{p}_2(\boldsymbol{q}) &= \begin{bmatrix} q_x \\ q_y \end{bmatrix} + \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} w_2 \\ \frac{b_{\text{top}}}{2} - v_2 \end{bmatrix}.
\end{aligned} \tag{2}$$

In (2), parameter l_b represents the undeformed beam length, while geometric quantities b_{top} and a_j, b_j, v_j, w_j, $j = 1, 2$ are shown in Fig. 1(b). The actuator lengths l_{DE1} and l_{DE2} are given as a function of generalized coordinates $\boldsymbol{q}$ as

$$\boldsymbol{l}_{\text{DE}}(\boldsymbol{q}) = \begin{bmatrix} l_{\text{DE1}}(\boldsymbol{q}) \\ l_{\text{DE2}}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{p}_1(\boldsymbol{q}) - \boldsymbol{d}_1\| - 2l_{cr1} \\ \|\boldsymbol{p}_2(\boldsymbol{q}) - \boldsymbol{d}_2\| - 2l_{cr2} \end{bmatrix} \tag{3}$$

where $l_{crj}, j = 1, 2$, are passive regions which do not contribute to the effective actuators lengths; see Fig. 1(b). The full kinematic model is obtained by replacing (2) into (3).

B. Dynamic Model

The dynamic model is derived using a Lagrangian approach. The Lagrangian $\mathcal{L}$ of the system is given as the difference between kinetic and potential energies, as follows:

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\intercal \boldsymbol{M}\dot{\boldsymbol{q}} - \mathcal{V}(\boldsymbol{q}) \tag{4}$$

with constant inertia matrix (due to the adopted choice of $\boldsymbol{q}$)

$$\boldsymbol{M} = \begin{bmatrix} m_H & 0 & 0 \\ 0 & m_H & 0 \\ 0 & 0 & J_H \end{bmatrix}$$

where m_H and J_H are the rigid plate mass and moment of inertia, and the configuration-dependent potential energy

$$\mathcal{V}(\boldsymbol{q}) = m_H g q_y + \mathcal{V}_b(\boldsymbol{q}) + \mathcal{V}_{DE}(\boldsymbol{q}) \tag{5}$$

consisting of a gravitational contribution (with g denoting the gravitational acceleration) and elastic contributions from the flexible backbone $\mathcal{V}_b$ and the DEAs $\mathcal{V}_{\text{DE}}$. The flexible beam is solely characterized by its potential energy, expressed in a lumped fashion via the beam constrained model [32]

$$\mathcal{V}_b(\boldsymbol{q}) = \frac{EI}{2l_b}\left(\boldsymbol{\nu}^\intercal \boldsymbol{G}\boldsymbol{\nu} + \frac{\left(u_x + \frac{1}{2}\boldsymbol{\nu}^\intercal \boldsymbol{P}\boldsymbol{\nu}\right)^2}{\frac{I}{Al_b^2} - \boldsymbol{\nu}^\intercal \boldsymbol{Q}\boldsymbol{\nu}}\right) \tag{6}$$

with

$$u_x = \frac{b_{\text{top}}}{2l_b}(1-\cos\alpha) + \frac{q_y}{l_b}, \quad \boldsymbol{\nu} = \begin{bmatrix} -\frac{b_{\text{top}}}{2l_b}\sin\alpha - \frac{q_x}{l_b} \\ \alpha \end{bmatrix},$$

$$\boldsymbol{G} = \begin{bmatrix} 12 & -6 \\ -6 & 4 \end{bmatrix}, \boldsymbol{P} = \begin{bmatrix} \frac{6}{5} & -\frac{1}{10} \\ -\frac{1}{10} & \frac{2}{15} \end{bmatrix}, \boldsymbol{Q} = \begin{bmatrix} -\frac{1}{700} & \frac{1}{1400} \\ \frac{1}{1400} & -\frac{11}{6300} \end{bmatrix}$$

where E is the Young's modulus of the beam material, while A and I are the beam cross-sectional area and second moment of area, respectively. The potential energy of the rolled DEAs explicitly accounts for the material elastic nonlinearities through the following equation [31]:

$$\mathcal{V}_{\text{DE}}(\boldsymbol{q}) = \sum_{j=1}^{2}\sum_{h=1}^{3}\Omega_j C_{jh}\left(\frac{l_{\text{DE}j}(\boldsymbol{q})^2}{L_{1j}^2} + \frac{2L_{1j}}{l_{\text{DE}j}(\boldsymbol{q})} - 3\right)^h \tag{7}$$

where $\Omega_j = \pi(R_{oj}^2 - R_{ij}^2)L_{1j}$ is the volume of DEA j, with geometric parameters L_{1j}, R_{oj}, and R_{ij} representing the undeformed axial length, outer radius, and inner radius of roll j, respectively, while C_{jh} are Yeoh parameters describing the hyperelastic DE response.

Using Lagrangian mechanics, the equations of motion are

$$\begin{aligned} &\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^\intercal - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^\intercal = \boldsymbol{M}\ddot{\boldsymbol{q}} + \frac{\partial \mathcal{V}(\boldsymbol{q})}{\partial \boldsymbol{q}} \\ &= -\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}^\intercal(\boldsymbol{q})\boldsymbol{\Gamma_e}\boldsymbol{u} \end{aligned} \tag{8}$$

where the right-hand side of (8) represents nonconservative external forces. The first term on the right-hand side of (8) describes dissipation forces via positive definite matrix

$$\begin{aligned} \boldsymbol{D}(\boldsymbol{q}) = &\begin{bmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_\alpha \end{bmatrix} \\ &+ \boldsymbol{J}^\intercal(\boldsymbol{q})\begin{bmatrix} \frac{\eta_1\Omega_1}{L_{11}l_{DE1}(\boldsymbol{q})} & 0 \\ 0 & \frac{\eta_2\Omega_1}{L_{12}l_{DE2}(\boldsymbol{q})} \end{bmatrix}\boldsymbol{J}(\boldsymbol{q}) \end{aligned} \tag{9}$$

which is composed of a linear viscous damping contribution d_x, d_y, d_α in the coordinate directions and a second nonlinear contribution representing DEAs viscoelastic dissipations through parameters η_1 and η_2. The actuation Jacobian $\boldsymbol{J}$ is given by

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{l_{DE}}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{10}$$

with $\boldsymbol{l_{DE}}(\boldsymbol{q})$ given by the kinematic model (3). The second term on the right-hand side of (8) denotes the rolled DEAs actuator forces depending on $\boldsymbol{u} = [v_{DE1}^2 \quad v_{DE2}^2]^\intercal$, with $v_{\text{DE}j}$ denoting the voltage on DEA j, and

$$\boldsymbol{\Gamma_e} = \begin{bmatrix} \gamma_{e1} & 0 \\ 0 & \gamma_{e2} \end{bmatrix}, \quad \gamma_{ej} = \frac{\pi(R_{oj}^2 - R_{ij}^2)\alpha_{ej}\epsilon_0\epsilon_{rj}}{2L_{3j}^2} \tag{11}$$

where ϵ_0 is the vacuum permittivity, while L_{3j}, α_{ej}, and ϵ_{rj} represent the undeformed membrane thickness, ratio between electrode area and passive membrane surface, and the relative permittivity of DEA j, respectively.

By collecting (4)–(11), the model can be formulated in state-space form. By defining state $\boldsymbol{x} = [\boldsymbol{x_1}^\intercal \quad \boldsymbol{x_2}^\intercal]^\intercal = [\boldsymbol{q}^\intercal \quad \dot{\boldsymbol{q}}^\intercal]^\intercal$, input $\boldsymbol{u}$, and measured output $\boldsymbol{y} = \boldsymbol{l_{DE}}$, from (8), we obtain

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f_x}(\boldsymbol{x}, \boldsymbol{u}) \\ &= \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{M}^{-1}\left(-\left(\frac{\partial \mathcal{V}(\boldsymbol{x_1})}{\partial \boldsymbol{x_1}}\right)^\intercal - \boldsymbol{D}(\boldsymbol{x_1})\boldsymbol{x_2} + \boldsymbol{J}^\intercal(\boldsymbol{x_1})\boldsymbol{\Gamma_e}\boldsymbol{u}\right) \end{bmatrix}, \\ \boldsymbol{y} &= \boldsymbol{h_x}(\boldsymbol{x}) = \boldsymbol{l_{DE}}(\boldsymbol{x_1}). \end{aligned} \tag{12}$$

C. Reduced Dynamic Model

Model (12) results in a state-space description with six state variables. The full model exhibits fast vertical dynamics compared to those along the horizontal and bending directions, which is related to the fact that long and slender beams possess a significantly larger compressive stiffness compared to the bending one. This leads to high numerical stiffness of the model when operating nearby the vertical configuration. Even though this does not cause issues when using advanced ODE discretization schemes for offline simulation, problems may arise when integrating the model in real time. To minimize the estimator complexity, it might be advantageous to reduce the system order by introducing additional constraints while still approximating the original dynamics as tight as possible.

In [33], we showed that neglecting the contribution of the vertical force components leads to an equivalent kinematic constraint for the beam tip displacement. Such approach revealed to be effective when performing qualitative stability analysis on the T-Platform. Inspired by this idea, we propose here a more general approach for model order reduction. First, we define the vertical force acting on the beam

$$\frac{\partial \mathcal{V}_b(\boldsymbol{q})}{\partial q_y} =: \tau_y. \tag{13}$$

By replacing (6) in (13) and solving for q_y, we arrive at

$$q_y(q_x, \alpha) = \frac{\tau_y l_b}{EA} - \boldsymbol{\nu}^\intercal\left(\frac{l_b}{2}\boldsymbol{P} + \frac{\tau_y l_b^3}{EI}\boldsymbol{Q}\right)\boldsymbol{\nu} - \frac{b_{\text{top}}}{2}(1-\cos\alpha). \tag{14}$$

Note that q_y in (14) only depends on other kinematic variables q_x, α and on the vertical beam force τ_y. In the full order model (12), τ_y is the sum of dynamic forces such as inertia and damping as well as static forces, i.e., gravity and DEA prestrain force. By assuming that the static force contributions are almost constant and dominate the dynamic ones, we have $\tau_y \approx$ constant. In this way, (14) becomes an algebraic relationship between q_y and the variables q_x, α, and thus, it can be used to reduce the number of DoF in the model.

To formally derive the simplified model, we define the reduced configuration vector in $\boldsymbol{r} = [q_x \quad \alpha]^\intercal$, and use (14) to express the transformation between $\boldsymbol{r}$ and $\boldsymbol{q}$ through

$$\boldsymbol{q}(\boldsymbol{r}) = \begin{bmatrix} q_x \\ q_y(q_x, \alpha) \\ \alpha \end{bmatrix},$$

$$\boldsymbol{J_{qr}}(\boldsymbol{r}) = \frac{\partial \boldsymbol{q}(\boldsymbol{r})}{\partial \boldsymbol{r}} = \begin{bmatrix} 1 & 0 \\ \frac{\partial q_y(q_x,\alpha)}{\partial q_x} & \frac{\partial q_y(q_x,\alpha)}{\partial \alpha} \\ 0 & 1 \end{bmatrix}. \tag{15}$$

We can then replace (15) into the Lagrangian (4), and derive new equations of motion with respect to coordinates $\boldsymbol{r}$

$$\begin{aligned} &\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{r}}}\right)^\intercal - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{r}}\right)^\intercal \\ &\quad = \boldsymbol{M_r}(\boldsymbol{r})\ddot{\boldsymbol{r}} + \boldsymbol{C_r}(\boldsymbol{r}, \dot{\boldsymbol{r}})\dot{\boldsymbol{r}} + \left(\frac{\partial \mathcal{V}(\boldsymbol{q}(\boldsymbol{r}))}{\partial \boldsymbol{r}}\right)^\intercal \\ &\quad = -\boldsymbol{D_r}(\boldsymbol{r})\dot{\boldsymbol{r}} + \boldsymbol{J_{qr}^\intercal}(\boldsymbol{r})\boldsymbol{J}^\intercal(\boldsymbol{q}(\boldsymbol{r}))\boldsymbol{\Gamma_e}\boldsymbol{u} \end{aligned} \tag{16}$$

with inertia and damping matrices given by

$$\begin{aligned} \boldsymbol{M_r}(\boldsymbol{r}) &= \boldsymbol{J_{qr}^\intercal}(\boldsymbol{r})\boldsymbol{M}\boldsymbol{J_{qr}}(\boldsymbol{r}), \\ \boldsymbol{D_r}(\boldsymbol{r}) &= \boldsymbol{J_{qr}^\intercal}(\boldsymbol{r})\boldsymbol{D}(\boldsymbol{q}(\boldsymbol{r}))\boldsymbol{J_{qr}}(\boldsymbol{r}) \end{aligned}$$

while $\boldsymbol{C_r}$ is such that

$$\boldsymbol{C_r}(\boldsymbol{r}, \dot{\boldsymbol{r}})\dot{\boldsymbol{r}} = \dot{\boldsymbol{M}}_{\boldsymbol{r}}(\boldsymbol{r})\dot{\boldsymbol{r}} - \frac{1}{2}\frac{\partial\left(\dot{\boldsymbol{r}}^\intercal \boldsymbol{M_r}\left(\boldsymbol{r}\right)\dot{\boldsymbol{r}}\right)}{\partial \boldsymbol{r}}.$$

Finally, we introduce the reduced state $\boldsymbol{\xi} = [\boldsymbol{\xi_1}^\intercal \quad \boldsymbol{\xi_2}^\intercal]^\intercal = [\boldsymbol{r}^\intercal \quad \dot{\boldsymbol{r}}^\intercal]^\intercal$ and obtain the state-space description of the reduced model with measured output $\boldsymbol{\zeta}$ (some explicit dependencies on $\boldsymbol{\xi}$ are omitted for the ease of conciseness)

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \boldsymbol{f_\xi}(\boldsymbol{\xi}, \boldsymbol{u}) \\ &= \begin{bmatrix} \boldsymbol{\xi_2} \\ \boldsymbol{M_r}^{-1}\left(-(\boldsymbol{C_r} + \boldsymbol{D_r})\boldsymbol{\xi_2} - \left(\frac{\partial \mathcal{V}}{\partial \boldsymbol{r}}\right)^\intercal + \boldsymbol{J_{qr}^\intercal}\boldsymbol{J}^\intercal\boldsymbol{\Gamma_e}\boldsymbol{u}\right) \end{bmatrix}, \\ \boldsymbol{\zeta} &= \boldsymbol{h_\xi}(\boldsymbol{\xi}) = \boldsymbol{l}_{\mathrm{DE}}(\boldsymbol{q}(\boldsymbol{\xi_1})). \end{aligned} \tag{17}$$

Model (17) offers a means to simplify the implementation of the system-level self-sensing without sacrificing the estimation accuracy, as will be clarified in Section V.

IV. System-Level Self-Sensing Architecture

The goal of the system-level self-sensing architecture is to estimate in real time the system state, consisting of the robot configuration $\boldsymbol{q}$ and corresponding velocity $\dot{\boldsymbol{q}}$, based on available measurements of the DEA voltages $\boldsymbol{u}$ and lengths $\boldsymbol{l}_{\mathrm{DE}}$, as shown in Fig. 3. The overall architecture is based on a two-level estimation process. At actuator level, the voltage and current measurements are used to estimate the individual actuator lengths $\boldsymbol{l}_{\mathrm{DE}}$. At system level, those actuator lengths serve as measured outputs of a model-based observer for the system state. Both processes are described in this section.

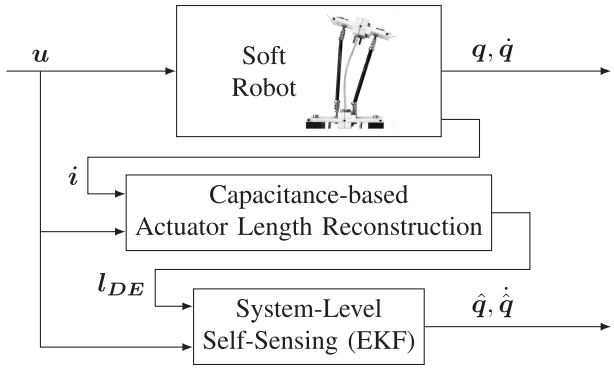

Fig. 3. Block diagram of the system-level self-sensing architecture.

A. Estimation of the DEA Lengths via Electrical Measurements

In previous works, a recursive-least-squares algorithm was proposed to estimate the DEA capacitance, and, in turn, its length based on voltage and current measurements collected during actuation [23]. This actuator-level self-sensing scheme is briefly summarized hereafter. To generate a sufficiently large current signal which allows extracting the capacitance information, a low-amplitude and high-frequency (f_e) voltage signal is superimposed to the high-amplitude and low-frequency actuation signals $v_{\mathrm{DE}j}$. If we assume that each DEA can be modeled as an RC series circuit, discretization of the circuit equation via prewarped Tustin method leads to the following equations [23]:

$$\begin{aligned} v_{\mathrm{DE}j}[k] - v_{\mathrm{DE}j}[k-1] &= \theta_{1j}[k]\left(i_{\mathrm{DE}j}[k] - i_{\mathrm{DE}j}[k-1]\right) \\ &\quad + \theta_{2j}[k]\left(i_{\mathrm{DE}j}[k] + i_{\mathrm{DE}j}[k-1]\right), \end{aligned} \tag{18}$$

$$R_{\mathrm{DE}j}[k] = \theta_{1j}[k], \quad C_{\mathrm{DE}j}[k] = \frac{\tan(\pi f_e T_s)}{2\pi f_e}\frac{1}{\theta_{2j}[k]} \tag{19}$$

with $i_{\mathrm{DE}j}$, $R_{\mathrm{DE}j}$, and $C_{\mathrm{DE}j}$ denoting electric current, instantaneous series electrical resistance, and instantaneous electrical capacitance of actuator j, respectively; T_s is the sampling time; and k denotes a discrete-time index variable. Since both $R_{\mathrm{DE}j}$ and $C_{\mathrm{DE}j}$ explicitly depend on the DEA deformation [23], [25], the $v_{\mathrm{DE}j} - i_{\mathrm{DE}j}$ relationship is directly affected by the actuator mechanical state. Equation (18) corresponds to a linear regression problem for the unknown (slowly time-varying) parameters θ_{1j} and θ_{2j} (related to R_{DEj} and C_{DEj}), which is solved in real time based on online measurements of voltage and current via recursive least squares techniques [23]. The estimated capacitance $C_{\mathrm{DE}j}$ can be used to reconstruct the corresponding DEA lengths $l_{\mathrm{DE}j}$, as described in Section V.

B. Estimation of the Robot Configuration Via Self-Sensing Data

In this section, the goal of estimating the T-Platform configuration $\boldsymbol{q}$ based on available actuator lengths $\boldsymbol{l}_{\mathrm{DE}}$ and squared voltages $\boldsymbol{u}$ is addressed. In principle, this objective, could be seen as an inverse kinematics problem, i.e., reconstruct $\boldsymbol{q}$ based on $\boldsymbol{l_{DE}}$ by inverting (3). Since the T-Platform represents a

3-DoF parallel kinematic structure, and only two independent measurements are available, equation $\boldsymbol{l}_{\mathrm{DE}}(\boldsymbol{q})$ cannot be uniquely inverted solely on the basis of (3). One possible solution to this underdetermined problem is to eliminate q_x via kinematic constraint (14), and solve (3) numerically for the two remaining DoFs. Despite seeming appealing, this approach fails in practice due to numerical issues (a more detailed discussion is postponed to Section V-C). On the other hand, while the dynamic model from Section III well captures the open loop system response, it fails to reach the fidelity needed for closed-loop control, since it cannot anticipate unavoidable external disturbances or parameter uncertainties.

To overcome the abovementioned shortcomings, we adopt a method that combines DEA length estimation with the dynamic model from Section III, in such a way to have enough information to resolve the kinematic ambiguity while improving the precision and robustness. To this end, we recall both full-order (12) and reduced-order (17) models, which provide two nonlinear state-space description of the T-Platform by considering the squared voltage $\boldsymbol{u}$ as input and $\boldsymbol{l}_{\mathrm{DE}}$ as measured output. The models expressed in this form offer a basis to use an observer-based approach, making use of the estimated actuator lengths $\boldsymbol{l}_{\mathrm{DE}}$ from Section IV-A as measured outputs. The EKF is a natural choice for this estimation task, due to its ability to perform state estimation in the presence of nonlinearities as well as its ease of implementation. For the EKF framework, we consider a scenario in which the system dynamics is expressed in continuous time while measurements are available in discrete time, given by

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t)) + \boldsymbol{w}_{\boldsymbol{f}}(t)\\ \boldsymbol{z}[k] &:= \boldsymbol{h}(\boldsymbol{x}(kT_s)) + \boldsymbol{w}_{\boldsymbol{h}}(kT_s)\end{aligned} \tag{20}$$

where time dependency of each variable has been made explicit for the ease of clarity, with round and squared brackets denoting continuous and discrete time, respectively. In (20), $\boldsymbol{f}$ and $\boldsymbol{h}$ can correspond to either the full system dynamics $\boldsymbol{f}_{\boldsymbol{x}}$ and measurement model $\boldsymbol{h}_{\boldsymbol{x}}$ [cf. (12)], or the reduced system dynamics $\boldsymbol{f}_{\boldsymbol{\xi}}$ and measurement model $\boldsymbol{h}_{\boldsymbol{\xi}}$ [cf. (17)], respectively. Normally distributed system noise $\boldsymbol{w}_{\boldsymbol{f}} \sim \mathcal{N}(0, \boldsymbol{S}_{\boldsymbol{f}})$ and measurement noise $\boldsymbol{w}_{\boldsymbol{h}} \sim \mathcal{N}(0, \boldsymbol{S}_{\boldsymbol{h}})$ are considered. Since the system dynamics are highly nonlinear and sensitive to discretization strategy, the so-called continuous-discrete EKF [34] is deemed the most suitable EKF variant for this estimation task, as it performs the prediction step by numerically integrating the continuous system dynamics until new measurements become available, and then computes a discrete update step. The a priori state $\hat{\boldsymbol{x}}^{-}[k]$ and covariance $\boldsymbol{P}^{-}[k]$ estimates are computed as the solution of

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}},\boldsymbol{u}(t)), \tag{21a}$$

$$\dot{\boldsymbol{P}} = \boldsymbol{A}(\hat{\boldsymbol{x}},t)\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^{\intercal}(\hat{\boldsymbol{x}},t) + \boldsymbol{S}_{\boldsymbol{f}} \tag{21b}$$

in the interval $t \in [(k-1)T_s \quad kT_s]$ starting from the last estimates of $\hat{\boldsymbol{x}}^{+}[k-1]$ and $\boldsymbol{P}^{+}[k-1]$, where $\boldsymbol{S}_{\boldsymbol{f}}$ is the the system noise covariance and $\boldsymbol{A}(\hat{\boldsymbol{x}},t)$ is defined as

$$\boldsymbol{A}(\hat{\boldsymbol{x}},t) = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}},\boldsymbol{u}=\boldsymbol{u}(t)}. \tag{22}$$

In this work, we will consider several possible methods to approximate the solution of Section (IV-B) (cf. Section V-E). The update step computes the Kalman gain matrix $\boldsymbol{K}[k]$ via the predicted state $\hat{\boldsymbol{x}}^{-}[k]$ and the covariance matrix $\boldsymbol{P}^{-}[k]$, and applies a correction to the a posteriori estimates $\hat{\boldsymbol{x}}^{+}[k]$ based on the predicted error of the measurement model and $\boldsymbol{P}^{+}[k]$, i.e.,

$$\boldsymbol{K}[k] = \boldsymbol{P}^{-}[k]\boldsymbol{C}^{\intercal}[k]\left(\boldsymbol{C}[k]\boldsymbol{P}^{-}[k]\boldsymbol{C}^{\intercal}[k] + \boldsymbol{S}_{\boldsymbol{h}}\right)^{-1}, \tag{23a}$$

$$\hat{\boldsymbol{x}}^{+}[k] = \hat{\boldsymbol{x}}^{-}[k] + \boldsymbol{K}[k]\left(\boldsymbol{z}[k] - \boldsymbol{h}(\hat{\boldsymbol{x}}^{-}[k])\right), \tag{23b}$$

$$\begin{aligned}\boldsymbol{P}^{+}[k] = {} & \left(\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{C}[k]\right)\boldsymbol{P}^{-}[k]\left(\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{C}[k]\right)^{\intercal}\\ & + \boldsymbol{K}[k]\boldsymbol{S}_{\boldsymbol{h}}\boldsymbol{K}^{\intercal}[k]\end{aligned} \tag{23c}$$

with $\boldsymbol{C}[k]$ given by

$$\boldsymbol{C}[k] = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}^{-}(kT_s)}. \tag{24}$$

V. Experimental Results and Discussion

This section deals with experimental validation of the proposed method. After presenting the experimental setup and the parameter calibration process, the proposed self-sensing schemes are experimentally validated and compared in terms of accuracy as well as online computation effort.

A. Experimental Setup

In order to calibrate the dynamic model needed for the system-level self-sensing architecture, a series of experiments were conducted on a prototype of the T-Platform depicted in Fig. 1(a). A schematic of the adopted experimental setup is shown in Fig. 2(a). It consists of an optical measurement system (Camera: Oryx ORX-10GS-32S4C-C, Lens: Computar V2528-MPY) and two high-voltage amplifiers with integrated current sensing capabilities (Hivolt HA3B3-S +-3 kV). The amplifiers are controlled by a Speedgoat Performance real-time target machine (Intel Core i7 4.2 GHz CPU, 32 GB RAM) with IO133 Analog I/O Module. The video data as well as the electrical measurements are collected on a Windows PC running MATLAB software. The setup allows us to collect video data at a rate of 120 frames per second, while the input voltages of the DEAs as well as the measured currents are sampled at 10 kHz ($T_s = 100\,\mu$s). After recording, a number of post-processing tasks are performed for camera calibration, temporal synchronization to electrical measurements, as well as extraction and tracking of points of interest over time, resulting in trajectories of optically measured T-Platform pose $\boldsymbol{q}[k]$ and DE lengths $\boldsymbol{l}_{\boldsymbol{DE}}[k]$ as well as voltage $v_{DEj}[k]$ and current $i_{DEj}[k]$ signals acquired via electrical sensors.

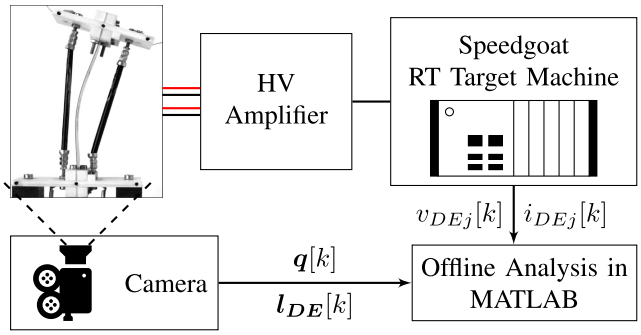

Fig. 4. Block diagram of the experimental setup.

B. Model Identification

This section presents experimental identification of the parameters required by the models presented in Section III. While some parameters can be assumed as constant, such as the ones describing system geometry and inertia, material parameters related to the DEA mechanical and electrical properties are influenced by environmental factors such as the temperature [35]. Systematic investigation of how this temperature dependency influences the self-sensing performance would certainly be of interest. However, this would require dedicated studies which are beyond the scope of our article, whose main goal is instead to demonstrate feasibility of the system-level self-sensing concept. In the sequel, we will ensure that self-sensing calibration and validation experiments are carried out under the same environmental conditions (similarly to other works dealing with DE self-sensing, e.g., [21], [22], [23], [24], [25], [26]). Fig. 5 shows the experimental data used for model calibration, based on a test in which the DEAs are alternately excited with unipolar sinusoidal signals at 0.1 Hz with a maximum amplitude of 2.9 kV. To enable capacitance sensing, a $f_e = 1\,\text{kHz}$ sinusoidal voltage with an amplitude of $\pm 100\,\text{V}$ is superimposed to the driving voltage, sampled at 10 kHz. Fig. 5(a) reports the applied input voltage signals and the optically acquired DEA lengths, while the optically measured configuration over time is shown as dashed black lines in Fig. 5(b). In the experiment, the T-Platform starts from a right-deflected position and, after exceeding a threshold voltage of the right-side actuator, performs a snap-through to the left side. The system possesses stable equilibria on both sides (bistability), an effect that has been thoroughly analyzed in our previous works [31], [33]. The total range of motion starts at a bending angle of approximately $-10°$ and ends at $+20°$, while travelling a horizontal distance of 15 mm and showing a vertical displacement of 2 mm. Asymmetries between positive and negative displacements, as well as in the actuator lengths, are reasonably due to inaccuracies during manufacturing involving DEA manual prestretching and rolling processes. Reducing those effects is in principle possible through design improvements [31], albeit the sensing performance remains unaffected from such structural optimization, and thus it is not further considered in this work. The resulting estimated capacitances, shown in the lower part of Fig. 5(a), follow the behavior of the corresponding actuator lengths. Similarly to [23], a moderate (yet repeatable) dependence of the capacitance on the applied high voltage level is observed, due to the non-ideal behavior of the rolled DEAs. Since this effect is systematic, it can be compensated using a phenomenological description of the actuator length [23]. An accurate fit is obtained via function

$$l_{DEj}(C_{DEj}, v_{DEj}) = \sum_{m=0}^{1}\sum_{n=0}^{3} p_{jmn} C_{DEj}^{m} v_{DEj}^{n} \tag{25}$$

with coefficients p_{jmn} calibrated on the experimental data based on a least squares fit. After calibration, the estimated DEA lengths well reproduce the optical measurements, as shown in Fig. 5(c). Next, we focus on the dynamic model parameters, whose identification is needed to tightly capture the asymmetric and bistable behavior of the T-Platform. Such parameters are determined in several steps. As a first step, parameters $l_b, E, I, A, m_H, J_H, \epsilon_0, \epsilon_r, g, R_{ij}, R_{oj}, \alpha_{ej}$ are fixed to their known or directly measured values, as listed in the appendix. All remaining values are obtained by formulating suitable nonlinear optimization problems (MATLAB `optimproblem/solve` and `fminsearch`) aimed at minimizing the deviation between model predictions and measurements in Fig. 5. In the second step, kinematic parameters w_j, v_j, a_j, and b_j are estimated by minimizing the integral squared error between the measured $\boldsymbol{l_{DE}}$ and the one predicted via the kinematic model (2)–(3) through the measured $\boldsymbol{q}$. During this step $l_{cr1} = l_{cr2} = 0$ is assumed, as l_{crj} cannot be determined optically. In the third step, b_{top}, l_{crj}, C_{jh}, η_j, d_x, d_y, and d_α are optimized by minimizing the integral squared error between the optically measured $\boldsymbol{q}$ and the value predicted by the full-order system model (12). The calibrated model well captures the experiments, see the blue lines in Fig. 5(b). Finally, τ_y is optimized to let (17) match the response of (12). Fine tuning of structural damping parameters d_x, d_y, and d_α is also performed at this stage, to further improve accuracy. All identified parameter values are reported in the appendix.

C. Results of Purely Kinematic System-Level Self-Sensing

As mentioned in Section IV-B, one conceivable solution of the estimation problem consists of numerically solving for $\boldsymbol{q}$ the nonlinear system of three equations given by (3) and (14), where $\boldsymbol{l}_{\text{DE}}$ is known through the capacitance while τ_y is calibrated to best fit the data. As shown in Fig. 6, this approach fails in practice. These plots are obtained by solving (3) and (14) for the calibration experiment Fig. 5 at each timestep using `fsolve()`, with initial guesses chosen based on either the measured reference trajectory at the current time [Fig. 6(a)], or based on the previous estimate [Fig. 6(b)]. Further numerical investigations revealed the reasons for this inaccuracy, i.e., the Jacobian of the transformation from $\boldsymbol{r}$ to $\boldsymbol{l}_{\text{DE}}(\boldsymbol{q}(\boldsymbol{r}))$ exhibits a nearly-singular behavior in several points of the computed trajectory. This makes the map hard to invert in a local sense, as evidenced by different trajectories in Fig. 6 obtained for different initial guesses. Since there is no mechanism to encode confidences into the model, such as tuning $\boldsymbol{S_f}$ and $\boldsymbol{S_h}$ matrices for the EKF, this approach lacks the possibility to optimize the estimation performance.

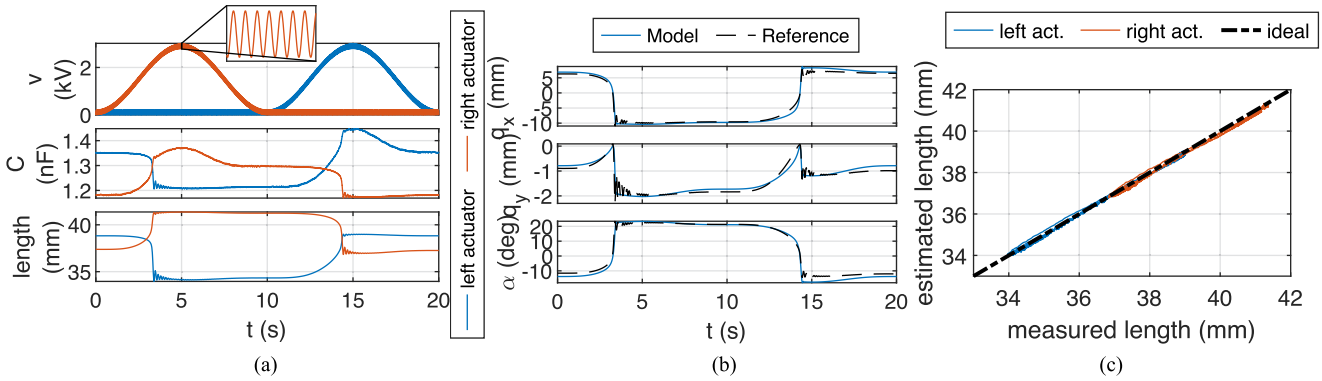

Fig. 5. Identification results for an experiment with unipolar sinusoidal excitation. Input voltages, resulting capacitances and actuator lengths shown in (a), optically measured configuration and predicted model response are shown in (b), and actuator-level self-sensing calibration is shown in (c).

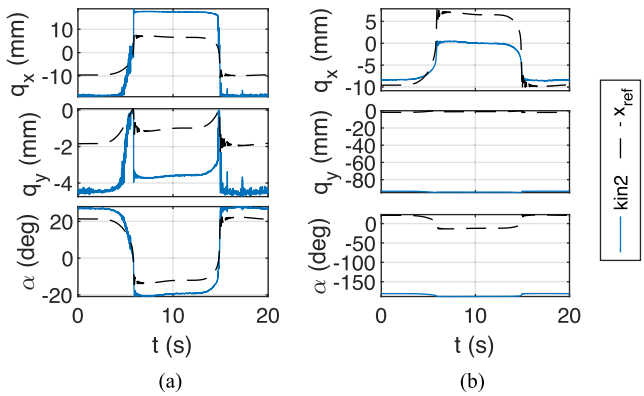

Fig. 6. Purely kinematic estimation scheme on the calibration dataset fails to provide useful estimates. Solutions shown for initial guesses at each timestep based on (a) measured reference trajectory and (b) on previous estimate.

TABLE I
PREDICTION ERRORS (RMS)

	DoF	Model	EKF3	EKF2
Sine	q_x (mm)	0.74	0.50	0.48
	q_y (mm)	0.15	0.22	0.20
	α (°)	1.82	0.44	0.63
Modulated Sine	q_x (mm)	4.49	0.60	0.55
	q_y (mm)	0.44	0.25	0.23
	α (°)	9.12	0.47	0.59
Steps	q_x (mm)	1.16	0.65	0.61
	q_y (mm)	0.20	0.28	0.26
	α (°)	2.31	0.50	0.60
Stairs	q_x (mm)	1.19	0.65	0.63
	q_y (mm)	0.23	0.27	0.25
	α (°)	2.24	0.47	0.58
Manual Pushing	q_x (mm)	8.13	0.98	0.97
	q_y (mm)	0.69	0.24	0.22
	α (°)	16.39	0.79	0.91

D. Results of EKF-Based System-Level Self-Sensing

For a qualitative and quantitative evaluation of the system-level self-sensing scheme, we performed experiments with different voltage trajectories, and evaluated the response of the underlying open-loop model as well as the proposed observer-based method. Two different versions of the methods are considered, in which the estimation is performed with full model (12) or with its reduced version (17), denoted as EKF3 and EKF2, respectively. In both cases, the update frequency is fixed at 1 kHz, while matrices $\boldsymbol{S_f}$ and $\boldsymbol{S_h}$ are tuned via numerical optimization with the aim of minimizing the integral squared difference between measured and estimated $\boldsymbol{q}$.

Fig. 7 evaluates the proposed self-sensing schemes for different experiments, and compares them to the open-loop model response. The input voltages in the first column represent the low-frequency component of the driving voltage, without considering the high-frequency injection needed for capacitance sensing for better visualization. In the second and third columns, the optically measured configuration over time (black dashed lines) is compared to the estimated trajectories (solid colored lines) according to the open-loop model (12) (center column) and according to the proposed self-sensing scheme (right column). Experiments (a)–(c) explore the system behavior for different voltage waveforms without any external disturbance, while in (d) the T-Platform was pushed by hand without applying any actuation voltage. The trend is generally well captured by the open-loop model in the undisturbed experiments (a)–(c), even though the precise postbuckling behavior is sometimes mispredicted. In addition, the open-loop model cannot account for external forces in (d), and therefore estimate no displacement over time, as expected. The proposed self-sensing schemes, on the other hand, follow the system motion at all times, while also providing more precise numerical estimations than the open-loop model. The corresponding videos are shown in the media attachment. A high level of robustness is achieved, and it is difficult to visually distinguish the differences between EKF3 and EKF2 schemes. This aspect becomes more clear in Table I, which lists the root mean squared (rms) errors for all three degrees of freedom corresponding to the experiments in Fig. 7, as well as the calibration experiment labeled *Sine*. The table confirms the visual observations, even though the EKF3 scheme generally performs slightly better than EKF2,

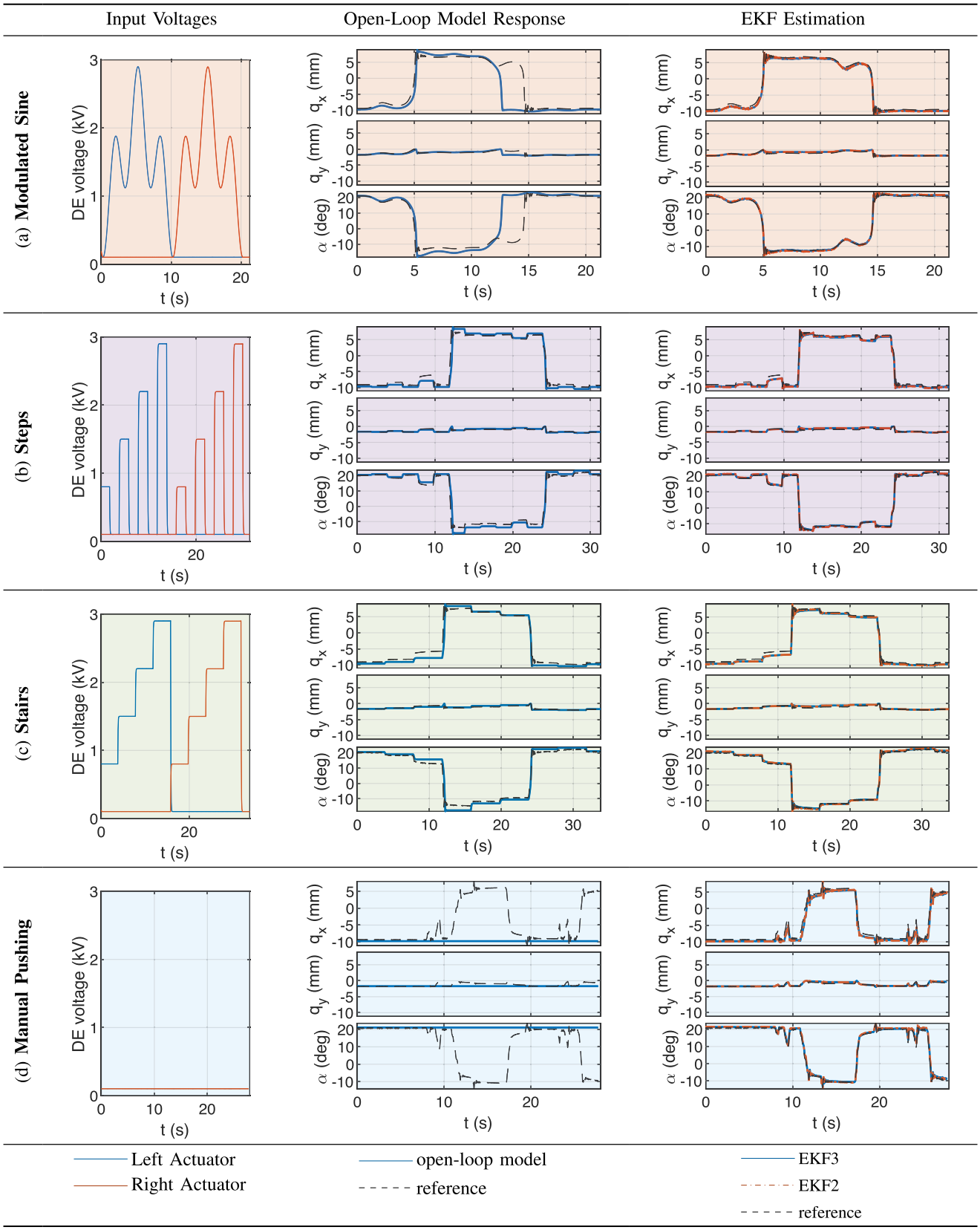

Fig. 7. Evaluation of the proposed self-sensing schemes (EKF3/EKF2) for different experiments (a)–(d) and comparison to the open-loop model predictions.

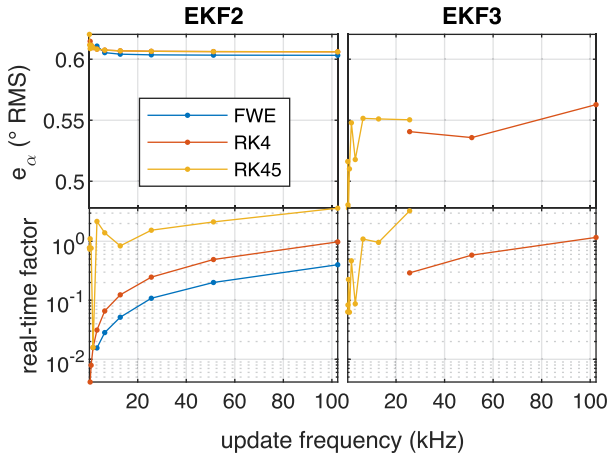

Fig. 8. Average angular error and real-time factor (ratio of computation time versus duration of the experiment) of the respective best-performing EKF2 and EKF3 operating at different sampling frequencies.

as expected. The superior robustness of self-sensing compared to the open-loop model can be explained as follows. Thanks to the EKF, the configuration $\boldsymbol{q}$ estimated by the open-loop model is corrected in real time via output $\boldsymbol{l_{DE}}$, which in our case is accurately reconstructed through voltage and current measurements as shown in Fig. 5(c). The $\boldsymbol{q}$ estimated by the EKF is thus heavily based on the kinematic model $\boldsymbol{l_{DE}}(\boldsymbol{q})$, while the dynamic model is mostly used to solve the kinematic ambiguity due to the underactuation. Therefore, we can expect the proposed self-sensing approach not to be highly sensitive to changes in the dynamic model parameters or to external disturbances, as long as $\boldsymbol{l}_{\mathrm{DE}}$ is accurately estimated and provided that kinematic model $\boldsymbol{l_{DE}}(\boldsymbol{q})$ is calibrated with good precision.

E. Real-Time Considerations

Almost all operations required by the system-level self-sensing scheme in Section IV are naturally formulated in discrete time, except for Section (IV-B). To enable effective implementation of the estimation method on real-time digital architectures, efficient ways to numerically approximate the solution of Section (IV-B) are required. In Fig. 8, a study is conducted to compare different numerical integration methods at various update frequencies in terms of estimation quality (measured in terms of the average error in the estimation of α, for simplicity) and computation speed (expressed as the ratio between computation time and duration of the experiment). We considered three numerical approximation methods, i.e., a forward Euler discretization (FWE), a fourth order Runge–Kutta method (RK4), and a fourth order Runge–Kutta method with adaptive step size regulation (RK45, allowed to take substeps based on estimated local error) [36]. Linear interpolation was used to generate the values of $\boldsymbol{l}_{\mathrm{DE}}$ at intermediate points in time in which direct estimations were not available. For each fixed update frequency, the entries of diagonal matrices $\boldsymbol{S_f}$ and $\boldsymbol{S_h}$ were determined using numerical optimization to find the filter configuration that produces a minimal estimation error, as defined in Section V-D on the calibration dataset (*Sine*). Each optimized filter configuration was then evaluated on the full dataset (calibration data as well as the evaluation experiments, c.f. Fig. 7). Diverging configurations and those with large average errors are omitted for visual clarity. In Fig. 8, the upper plots show the average angular error and the bottom plots indicates the computation speed relative to the duration of the experiment, with values below one indicating performance suitable for real-time implementation.

For the EKF2 method (left-hand side of Fig. 8), the RK4 scheme produced good estimates even at an update frequency of 400 Hz and slightly improves at larger frequencies. As expected, the FWE method starts to produce reasonable estimates only at 3.2 kHz and slightly improves with increasing frequency as well. Counterintuitively, it even outperforms the RK4 scheme. Possible explanations include compounding roundoff errors of the RK4 scheme manifesting at larger frequencies, or the appearance of non-optimal EKF configurations resulting from the nonlinear optimization procedure. Note that, for EKF2, both integration schemes are capable of producing fast (240× and 62× real-time speed for RK4 and FWE, respectively) and precise (average error of approximately 0.61°) estimates. The RK45 generally exhibits very similar quality of estimated states, but requires longer computation times as the RK4 scheme, due to the possibility of additional substeps. While this enables estimation at very low update frequencies, in a real-time setting one typically prefers a fixed step size scheme, owing to the fact that only a predetermined amount of time is available for state estimation.

In case of EKF3, the numerical optimization procedure did not produce a converging and reasonably precise FWE configuration. The main reason for this fact is the higher numerical stiffness due to the beam compression dynamics, which requires sufficiently small step sizes. In contrast, this issue is not present in EKF2 since (14) completely eliminates the stiff dynamics. The RK4 scheme is usable starting from 25.6 kHz with a real-time factor of ≈ 0.29 (i.e. 3.4× faster than real time). At frequencies up to 25.6 kHz, the adaptive RK45 method can achieve stable and precise estimates with mixed results in terms of computation time and estimation quality, but fails at larger frequencies. Similarly to before, these results seem to reflect a trade-off between truncation error, favoring higher-order methods at low frequencies, and round-off errors, favoring methods with less arithmetic operations (i.e., lower-order methods) at higher frequencies. Generally, the performance of tuned configurations at different frequencies does not exhibit a discernible trend for the adaptive step size scheme. However, from an implementation perspective, we can simply pick the best performing configuration that still meets our constraints on computation time (real-time factor smaller than one), such as the configuration tuned at an update frequency of 400 Hz with a real-time factor of 0.23.

Even though the presented evaluation was conducted on a powerful workstation (AMD Ryzen Threadripper PRO 5965WX 24 C @3.8 GHz, 256 GB RAM), we expect actual performance requirements of a suitably chosen configuration to be lower in terms of working memory and computation speed (e.g., when the real-time factor is much smaller than one). Despite both EKF2 and EKF3 schemes appear similar in terms of accuracy (cf.

Table I), we conclude that the former is preferable for real-time implementation, since it works well with simpler, faster, and fixed-step discretization methods.

VI. Conclusion

In this work, we demonstrated for the first time the feasibility of system-level self-sensing in a DEA-driven soft robot. The resulting scheme is able to accurately estimate the configuration and generalized velocity of our underactuated system solely on the basis of voltage and current signals, measured during various actuation tasks. By comparing the self-sensing estimations with camera-based measurements, an average estimation error of 0.47° (EKF3), or 0.60° (EKF2), namely 1.6%–2% of the actuation range, can be achieved. Remarkably, our algorithm is able to also reconstruct deformations caused by external loads, and not only high voltage actuation, with an accuracy of 0.79° (EKF3) and 0.91° (EKF2), respectively, thus, proving its robustness. An in-depth investigation of discretization schemes and sampling times was conducted to optimize estimation performance and online computation times, proving that EKF2 is the most suitable scheme for real-time applications. To the authors' knowledge, this is the first practical demonstration of system-level self-sensing in multi-DoF DEA soft robots. In future works, we plan to implement sensorless closed-loop control algorithms, which exploit the feedback from the proposed system-level self-sensing scheme to achieve closed-loop trajectory tracking. More extensive evaluation of the self-sensing method robustness, in particular its sensitivity to parameter variations and environmental effects, will also be systematically investigated.

Appendix
List of Parameters

$a_1 = a_2 = 5.25\,\mathrm{mm}$, $b_1 = b_2 = 7.02\,\mathrm{mm}$, $v_1 = v_2 = 8.16\,\mathrm{mm}$, $w_1 = w_2 = 9.65\,\mathrm{mm}$, $b_{top} = 5\,\mathrm{mm}$, $l_b = 53.32\,\mathrm{mm}$, $l_{cr1} = 2.74\,\mathrm{mm}$, $l_{cr2} = 2.05\,\mathrm{mm}$, $A = 1.88\,\mathrm{mm}^2$, $I = 3.52 \times 10^{-15}\,\mathrm{m}^4$, $E = 210\,\mathrm{GPa}$, $\epsilon_0 = 8.85\,\mathrm{pA\,s\,V^{-1}\,m^{-1}}$, $m_H = 12\,\mathrm{g}$, $J_H = 2.37\,\mathrm{mg\,m^2}$, $g = 9.81\,\mathrm{m/s^2}$, $d_x = 100\,\mu\mathrm{N\,s\,m^{-1}}$, $d_y = 100\,\mathrm{N\,s\,m^{-1}}$ (EKF3), $d_y = 0\,\mathrm{N\,s\,m^{-1}}$ (EKF2), $d_\alpha = 100\,\mu\mathrm{N\,s\,rad^{-1}}1$, $\tau_y = -13.37\,\mathrm{N}$, $C_{11} = 380.43\,\mathrm{kPa}$, $C_{21} = 84.16\,\mathrm{kPa}$, $C_{31} = -51.70\,\mathrm{kPa}$, $C_{12} = 248.08\,\mathrm{kPa}$, $C_{22} = 111.03\,\mathrm{kPa}$, $C_{32} = -24.89\,\mathrm{kPa}$, $L_{11} = L_{12} = 25.61\,\mathrm{mm}$, $L_{31} = L_{32} = 50\,\mu\mathrm{m}$, $R_{i1} = R_{i2} = 0\,\mathrm{mm}$, $R_{o1} = R_{o2} = 1.58\,\mathrm{mm}$, $\alpha_{e1} = \alpha_{e2} = 0.58$, $\epsilon_{r1} = \epsilon_{r2} = 2.80$, $\eta_1 = \eta_2 = 19.23\,\mathrm{kPa\,s}$.

References

[1] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft robots," in *Encyclopedia of Robotics*, M. H. Ang, O. Khatib, and B. Siciliano, Eds. Berlin, Heidelberg: Springer, 2020, pp. 1–15.

[2] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, "Biomedical applications of soft robotics," *Nature Rev. Mater.*, vol. 3, no. 6, pp. 143–153, Jun. 2018.

[3] E. Milana, "Soft robotics for infrastructure protection," *Front. Robot. AI*, vol. 9, Nov. 2022, Art. no. 1026891.

[4] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.

[5] Z. Lin et al., "Recent advances in perceptive intelligence for soft robotics," *Adv. Intell. Syst.*, vol. 5, no. 5, May 2023, Art. no. 2200329.

[6] W. Dou, G. Zhong, J. Cao, Z. Shi, B. Peng, and L. Jiang, "Soft robotic manipulators: Designs, actuation, stiffness tuning, and sensing," *Adv. Mater. Technol.*, vol. 6, no. 9, Sep. 2021, Art. no. 2100018.

[7] H. Wang, M. Totaro, and L. Beccai, "Toward perceptive soft robots: Progress and challenges," *Adv. Sci.*, vol. 5, no. 9, Sep. 2018, Art. no. 1800541.

[8] S. Wang and Z. Sun, "Hydrogel and machine learning for soft robots' sensing and signal processing: A review," *J. Bionic Eng.*, vol. 20, no. 3, pp. 845–857, May 2023.

[9] M. Cianchetti, F. Renda, A. Licofonte, and C. Laschi, "Sensorization of continuum soft robots for reconstructing their spatial configuration," in *Proc. 4th IEEE RAS Int. Conf. Biomed. Robot. Biomechatron.*, 2012, pp. 634–639.

[10] Q. Zhao, J. Lai, K. Huang, X. Hu, and H. K. Chu, "Shape estimation and control of a soft continuum robot under external payloads," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2511–2522, Oct. 2022.

[11] D. Lunni, G. Giordano, E. Sinibaldi, M. Cianchetti, and B. Mazzolai, "Shape estimation based on Kalman filtering: Towards fully soft proprioception," in *Proc. IEEE Int. Conf. Soft Robot.*, 2018, pp. 541–546.

[12] J. Shu et al., "Soft robots' Dynamic posture perception using kirigami-inspired flexible sensors with porous structures and long short-term memory (LSTM) neural networks," *Sensors*, vol. 22, no. 20, Oct. 2022, Art. no. 7705.

[13] P. Pustina, C. D. Santina, F. Boyer, A. De Luca, and F. Renda, "On the collocated form with input decoupling of lagrangian systems," 2023, doi: 10.1109/TRO.2024.3370089.

[14] Z. Liu, Z. Cai, H. Peng, X. Zhang, and Z. Wu, "Morphology and tension perception of cable-driven continuum robots," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 1, pp. 314–325, Feb. 2023.

[15] P. Preechayasomboon and E. Rombokas, "Sensuator: A hybrid sensor to soft robotic proprioception using recurrent neural networks," *Actuators*, vol. 10, no. 2, Feb. 2021, Art. no. 30.

[16] Y. Yang, H. Zhu, J. Liu, H. Lu, Y. Ren, and M. Y. Wang, "A proprioceptive soft robot module based on supercoiled polymer artificial muscle strings," *Polymers*, vol. 14, no. 11, Jan. 2022, Art. no. 2265.

[17] J. Shintake, V. Cacucciolo, H. Shea, and D. Floreano, "Soft biomimetic fish robot made of dielectric elastomer actuators," *Soft Robot.*, vol. 5, no. 4, pp. 466–474, Jun. 2018.

[18] U. Gupta, L. Qin, Y. Wang, H. Godaba, and J. Zhu, "Soft robots based on dielectric elastomer actuators: A review," *Smart Mater. Struct.*, vol. 28, no. 10, Oct. 2019, Art. no. 103002.

[19] Y. Guo, L. Liu, Y. Liu, and J. Leng, "Review of dielectric elastomer actuators and their applications in soft robots," *Adv. Intell. Syst.*, vol. 3, no. 10, 2021, Art. no. 2000282.

[20] O. Yasa et al., "An overview of soft robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 6, no. 1, pp. 1–29, May 2023.

[21] T. A. Gisby, B. M. O'Brien, and I. A. Anderson, "Self sensing feedback for dielectric elastomer actuators," *Appl. Phys. Lett.*, vol. 102, no. 19, May 2013, Art. no. 193703.

[22] T. Hoffstadt, M. Griese, and J. Maas, "Online identification algorithms for integrated dielectric electroactive polymer sensors and self-sensing concepts," *Smart Mater. Struct.*, vol. 23, no. 10, Sep. 2014, Art. no. 104007.

[23] G. Rizzello, P. Serafino, D. Naso, and S. Seelecke, "Towards sensorless soft robotics: Self-sensing stiffness control of dielectric elastomer actuators," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 174–188, Feb. 2020.

[24] S. Rosset, B. M. O'Brien, T. Gisby, D. Xu, H. R. Shea, and I. A. Anderson, "Self-sensing dielectric elastomer actuators in closed-loop operation," *Smart Mater. Struct.*, vol. 22, no. 10, Sep. 2013, Art. no. 104018.

[25] G. Rizzello, D. Naso, A. York, and S. Seelecke, "Closed loop control of dielectric elastomer actuators based on self-sensing displacement feedback," *Smart Mater. Struct.*, vol. 25, no. 3, p. 0 35034, Feb. 2016.

[26] J. Cao, W. Liang, J. Zhu, and Q. Ren, "Control of a muscle-like soft actuator via a bioinspired approach," *Bioinspiration Biomimetics*, vol. 13, no. 6, Oct. 2018, Art. no. 066005.

[27] I. A. Ivan, M. Rakotondrabe, P. Lutz, and N. Chaillet, "Quasistatic displacement self-sensing method for cantilevered piezoelectric actuators," *Rev. Sci. Instrum.*, vol. 80, no. 6, Jun. 2009, Art. no. 065102.

[28] I. A. Ivan, M. Rakotondrabe, P. Lutz, and N. Chaillet, "Current integration force and displacement self-sensing method for cantilevered piezoelectric actuators," *Rev. Sci. Instruments*, vol. 80, no. 12, Dec. 2009, Art. no. 126103.

[29] M. Rakotondrabe, "Combining self-sensing with an unkown-input-observer to estimate the displacement, the force and the state in piezoelectric cantilevered actuators," in *Proc. IEEE Amer. Control Conf.*, 2013, pp. 4516–4523.
[30] J. Prechtl, M. Baltes, J. Kunze, S. Seelecke, and G. Rizzello, "Towards sensorless configuration estimation in Multi-DoF soft robotic structures driven by rolled dielectric elastomer actuators," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2022, pp. 1152–1158.
[31] M. Baltes, J. Kunze, J. Prechtl, S. Seelecke, and G. Rizzello, "A bi-stable soft robotic bendable module driven by silicone dielectric elastomer actuators: Design, characterization, and parameter study," *Smart Mater. Struct.*, vol. 31, no. 11, Oct. 2022, Art. no. 114002.
[32] S. Awtar and S. Sen, "A generalized constraint model for two-dimensional beam flexures: Nonlinear strain energy formulation," *J. Mech. Des.*, vol. 132, no. 8, Aug. 2010, Art. no. 081009.
[33] J. Prechtl, J. Kunze, D. Bruch, S. Seelecke, and G. Rizzello, "Bistable actuation in multi-DoF soft robotic modules driven by rolled dielectric elastomer actuators," in *Proc. IEEE 4th Int. Conf. Soft Robot.*, 2021, pp. 82–89.
[34] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2008.
[35] T. Lu, C. Ma, and T. Wang, "Mechanics of dielectric elastomer structures: A review," *Extreme Mech. Lett.*, vol. 38, Jul. 2020, Art. no. 100752.
[36] J. J. Leader, *Numerical Analysis and Scientific Computation*, 2nd ed. New York, NY, USA: Chapman and Hall/CRC, Apr. 2022.

Johannes Prechtl received the M.S. degree, in 2019, in systems engineering from Saarland University, Saarbrücken, Germany, where he is currently working toward the Ph.D. degree in systems engineering.

He is currently a Research Assistant with Saarland University. His research interests include modeling and self-sensing of soft robotic structures based on dielectric elastomers.

Matthias Baltes received the master's degree in systems engineering from Saarland University, Saarbrücken, Germany.

He is currently a Research Assistant in the field of dielectric elastomer technology with Saarland University. His research interest includes static and dynamic approaches to design systems based on dielectric elastomer actuators.

Kathrin Flaßkamp received the Diploma degree in Technomathematik (applied mathematics with engineering) and the Ph.D. (Dr.rer.nat.) degree in mathematics from Paderborn University, Paderborn, Germany, in 2008 and 2013, respectively.

After two postdoc positions with Northwestern University, Evanston, IL, USA, and Bremen University, Bremen, Germany, respectively, she is currently a Full Professor of Systems Modeling and Simulation with Saarland University, Saarbrücken, Germany. Her research interests include modeling, simulation, optimization, and control, focusing on the development of numerical methods and on applications to technological systems.

Gianluca Rizzello (Senior Member, IEEE) was born in Taranto, Italy, in 1987. He received the master's (Hons.) degree in control engineering from the Polytechnic University of Bari, Bari, Italy, in 2012, and the Ph.D. degree in information and communication technologies from Scuola Interpolitecnica di Dottorato, a joint program between Polytechnic Universities of Torino, Bari, and Milano, Italy, in 2016.

He joined the Saarland University, Saarbrücken, Germany, first as a Postdoc Researcher and Group Leader Smart Material Modeling and Control, from 2016 to 2019, and subsequently as Assistant Professor in Adaptive Polymer Systems, since 2020. His research interests include modeling, control, and self-sensing of innovative mechatronic and robotic systems based on unconventional drive technologies, such as smart materials.