2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Model-based Engineering Framework for Soft Continuum Robots

Linus Witucki[1], Jan Eike Rösler[1] and Mike Barth

Abstract— **Soft continuum robots are gaining attention for their potential to enable inherently safe and adaptive human-robot collaboration, especially in dynamic industrial environments. However, the development of these robots varies drastically and no standardization exists. This is particularly problematic for soft continuum robots, because of the variety of different actuation methods, and control strategies. This paper addresses the challenge of engineering soft continuum robots by introducing a generalized framework that enables hardware abstraction and controller reuse. The approach combines a modular robot design by extending the Unified Robot Description Format (URDF) information model to support soft continuum robotics. Enabling the decoupling of hard- and software development. For the concept validation, a modular tendon-driven continuum robot was developed and integrated into the framework. The extended Unified (Continuum) Robot Description Format (U(C)RDF) enables visualization and controller parameterization through standardized interfaces, allowing for reusable software components across different actuation principles. This approach achieves a flexible and scalable engineering process for soft continuum robots, bridging the gap between research prototypes and industrial deployment. It lays the foundation for future developments in model-based design, automated control, and interoperability of soft continuum robotic systems.**

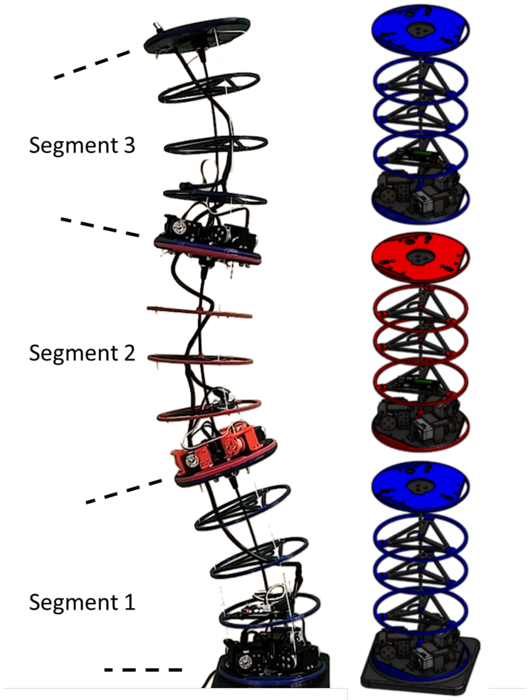


Fig. 1: Modular tendon driven soft continuum robot

I. INTRODUCTION

Soft continuum robots (SCRs) are emerging as a promising solution for inherently safe and adaptive human-robot collaboration [1], which can be applied to industrial environments and applications, where sensors and threshold monitoring are currently responsible for safe human interaction [2]. Unlike conventional rigid-body robots, SCRs are constructed from compliant materials such as silicone, hydrogels, rubber, and shape memory polymers, allowing them to adapt to complex and dynamic environments with increased safety and flexibility [3]. This intrinsic compliance allows safe robot-human interaction [4, 5], making them ideal for collaborative applications. However, because of their intensive engineering process, these technologies are not yet suitable for industrial applications and only a few applications are known.

In an effort to bridge the gap between research and industrial application, the entire SCR engineering process should be standardized [6]. In this contribution, the authors present a framework for developing SCRs. A key aspect to this end is the abstraction of the hardware. Hardware abstraction enables robotic applications to be developed independently of specific actuator technologies or mechanical configurations [7, 8]. This abstraction is particularly relevant for soft continuum robots, whose actuation principles (e.g., tendon-driven, pneumatic) vary widely and require flexible software interfaces.

When working with independent hard- and software development, parameterization and system description play a crucial role [9]. This can be done by using an information model as digital representations of the physical system. Considering the description of articulated robots, the Unified Robotics Description Format (URDF) is used mainly [10].

However, the variety of mathematical models used to describe the bending behavior hardware abstracted soft robots cannot be captured adequately by current information models, like the URDF, using rigid-body assumptions. As a result, there is no standardized way to represent SCRs in model-based engineering tools, making their design and deployment inconsistent and intuition-driven. Extending or adapting URDF-like formats to accommodate soft continuum robotics would enable structured, machine-readable descriptions of soft continuum robot properties, such as curvature. This can serve as the backbone of an adaptable engineering process, including simulation and control. Addressing this gap is essential for standardizing engineering workflows, improving interoperability, and supporting automatic simulation and control of continuous robotic systems.

---

[1] Linus Witucki and Jan Eike Rösler are co-first authors

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

The main contribution of this paper is a generalized engineering framework for the development of SCRs, enabled by extending the existing URDF model. The resulting Unified (Continuum) Robot Description Format (U(C)RDF) enables the description of SCRs in order to parameterize reusable software components. The concept was validated by developing the SCR depicted in Figure 1. In order to showcase the adaptability of the framework and U(C)RDF, a modular SCR with 3 segments has been developed. Using a U(C)RDF parameterized feed forward controller and joint trajectory controller validates the software independence and re-usability. Both controllers can handle any modular configuration of the SCR.

II. REQUIREMENTS ANALYSIS

In the field of continuum robotics, the majority of research efforts can be classified into four main areas: design, materials, modeling, and control [11]. In terms of existing continuum robots, Lee et al. [5] provide a comprehensive overview of existing soft robotic systems — including SCRs — and categorize them according to actuation, sensing, control, material, and manufacturing aspects. Wang et al. [12] investigate the deformation modes of soft and flexible materials, such as elongation, shortening, and bending, and relate these to potential materials, material properties, and actuation methods. The actuator strategies used in soft robotic systems have been systematically reviewed by El-Atab et al. [4], who classify them as electrically responsive, magnetically responsive, thermally responsive, photo-responsive, pressure-driven and explosive mechanisms. Beyond these general approaches, several studies address actuation principles tied to specific design and manufacturing techniques, as well as variable stiffness modulation in continuum robots by Dou et al. [13]. Alongside advances in physical construction, modeling and control of continuum robots have been intensively studied. Armanini et al. [14] summarize various modeling approaches in the field of continuum mechanics. In addition, Della Santina et al. [15] review current control strategies.

As a consequence of these main research areas, the diversity of actuation principles, actuation setups, sensors, and materials continues to grow. This leads to a wide range of physical SCRs whose behavior differs significantly depending on their actuation, design, and material. Resulting in a wide range of mathematical modeling methods and dedicated controllers for each SCR.

Based on the described challenges of

1. High variance in design, actuation and physical behavior
2. High variance in mathematical modeling
3. System-specific software solutions

the development of reusable software is necessary. Therefore, three requirements can be formulated for the engineering framework. In order to decouple hardware and software development, hardware abstraction methods should be used. This would increase the re-usability of software components and decrease the development time of new SCRs. In order to match a software component to an available SCR the software must be parameterizable. This can be done using information models of the robot. Additionally, the framework should be able to accommodate for different mathematical modeling approaches. In conclusion, the following three requirements to enable reusable software can be derived:

Req 1 Hardware abstraction.
Req 2 Software parameterization
Req 3 Plugin modeling approach

The following section reviews current developments in the field of robotics engineering, with an emphasis on existing development frameworks and approaches to information modeling, highlighting their capabilities, limitations, and relevance to the engineering of SCRs.

III. RELATED WORK

A. ROS Control Framework

The ros_control framework, introduced by Sachin Chitta et al. [16], provides a generic and extensible control architecture for robotic systems within the Robot Operating System (ROS). It separates hardware from control logic, enabling modular and reusable controller implementations. The framework supports real-time control through a plugin-based architecture, allowing developers to integrate various controllers, such as position, velocity, and effort controllers. By abstracting hardware communication through standardized interfaces, ros_control facilitates the integration of diverse robotic platforms and simplifies the development of complex control strategies. Each controller and other software components within the framework, like the robot state publisher, are parameterized using the URDF information model of the robot, which therefore plays a significant role within the framework.

With regard to the requirements, formulated in section II, the ros_control framework fulfills the abstraction requirement (REQ 1), as well as the software parameterization (REQ 2). Currently, it is limited to the development of rigid body robotics and does not support the development of SCRs. However, the ros_control framework is a good basis for the aim of this contribution.

B. Existing Information Models and Tools in Robotics

As stated in section I and the previous section, information models play a substantial role in robotics. Information models are used to parameterize controllers, simulations, visualizations, and inverse kinematic calculations. When dealing with conventional articulated robots, the URDF is used primarily [10]. With this XML-based information model, robots are described as a chain of rigid connections (so-called links) and point-like joints [17]. At this point, further representatives of information models for rigid body robot arms should be mentioned. These include the Simulation Description Format (SDF) and the Universal Scene Description (USD) format [17]. In addition to the articulated robot arm, these can describe other aspects, such as the direct environment of the robot.

Additionally, information models are available, considering the hardware components of an articulated robot, for example, the Hardware Robot Information Model (HRIM)

[18]. It is based on the usage of the Robot Operating System 2 (ROS2), to provide a vertical representation of all hardware components up to the communication patterns used.

All of these information models are used to parameterize different software components as required by REQ 2. The URDF is mostly used for controller parameterization, whereas the SDF and USD are commonly used for simulation parameterization. But the presented information models are not capable of representing SCRs, except for the pseudo-rigid body (PRB) approximation. The PRB simplification uses a concatenation of rigid body links and joints to approximate the continuous deformation of a soft robot [19]. This approximation has been used by Ibrahim A. Seleem et al. [20] to visualize SCRs inside the tool RViz. However, in order to capture a complete curvature, an extension to the information models is necessary. A further software tool for SCRs is the SoRoSim Toolbox[21], which provides a sophisticated offline simulation environment. However, this toolbox does not include the use of a stored information model. Instead, each parameter for the robotic system must be entered through a given user interface.

Based on the literature review, the need for a comment engineering framework for developing SCRs becomes apparent. The large variety of hardware structures and software models necessitates a modular approach. This can be supported by using information models as a means of parameterization. Allowing the separation of hardware and controller development for SCRs and enable scientists to focus on one area and reusing inventions of the other area. However, currently no existing information model is able to accurately represent a SCR, without drastic simplifications like the pseudo-rigid body method.

IV. CONCEPT

This section will provide a detailed description of the engineering framework for SCRs. As established, the ros_control framework, presented in section III, is a good reference for the development of articulated robots.

The developed framework is depicted in Figure 2. At the bottom (L1), the different actuation methods are depicted. In order to satisfy REQ 1, the hardware interface (HWI) (L2) of the ros_controll framework is adopted. The HWI coordinates the actuators, i.e. motors or valves, and provides the model-dependent interface towards the different controllers (L3). An information model (L4) describes the SCR and its northbound HWI for parameterization of generic SCR controllers, which is fulfilled by REQ 2. In contrast to articulated robots, multiple mathematical models can be used to provide the hardware abstraction of SCRs taking REQ 3 into account. Therefore, the information model must include the mathematical model used. For example, popular choices are the Piecewise Constant Curvature (PCC) model or the Piecewise Polynomial Curvature (PPC) model.

With this framework, two different SCRs, for example a tendon driven and a pneumatic bellow-actuated robot, which utilize the same modeling approach, can be controlled with the same software controller (L3).

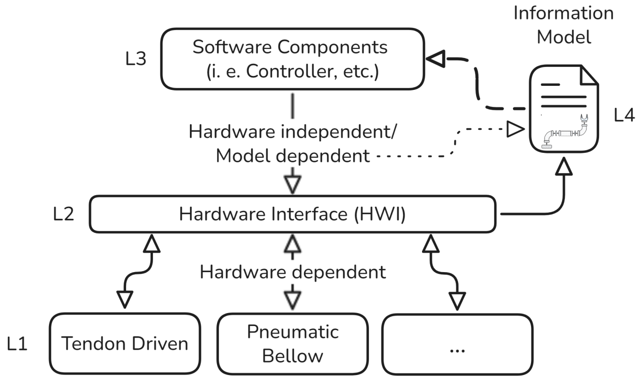


Fig. 2: General soft continuum robot engineering framework

The information model plays a key role, which brings together the different hardware and software components.

Section III outlined common information models representing articulated robots, such as the URDF. As established, their application for SCRs is limited to the PRB approximation. Therefore, an extension to the URDF, which accommodates SCRs, will be presented. Unlike a separate development, the URDF extension allows the use of both soft continuum robotics and regular robotics at the same time.

In order to adapt the URDF to accommodate for continuum robots, a new element, depicted in Listing 1, will be introduced. This new element, called *bendable*, combines aspects of a standard URDF *joint* and *link*. As described in section III, a *joint* represents a point-like articulation without any physical aspects. A *link* is the inverse, as it represents a structural component with mass and inertia but without actuation. The *bendable* represents both, because a segment of a continuum robot has physical parameters, like mass, as well as actuation regarding the curvature bend. Therefore, the elements, *parent, child* and *inertia* can be taken from the standard URDF. In addition, an element representing the curvature of the robot is necessary. This *curvature* element contains the parameters of a mathematical model that represents the robot and dynamic parameters such as the restoring force of the robot, which will counteract any bending force keeping the robot upright. The restoring force can, for example, stem from a backbone structure or similar structural components common in continuum robot manipulators. In order to accommodate the multitude of mathematical models, the *model* element can be distinguished with a type attribute.

The actuation of the robot can be limited in each degree of freedom. Using the standard version of URDF, these limits are represented within an element of the joint. Translating this structure is not valid considering continuum robots, as the limits are model-dependent and are therefore contained within the model element.

An exemplary U(C)RDF is depicted in Listing 1, which shows the aforementioned aspects using XML syntax. This information model of the robot can now be visualized using, for example, the Robot Visualization (RViz) tool from the ROS2 framework.

Listing 1: XML element SCR segment

```
<robot xmlns:xacro=”...” name”softrobot”>
  <link name=”base”/>
  <link name=”eef”/>
  <bendable name=”segment”>
    <parent link=”base”>
    <child link=”eef”>
    <curvature>
      <model type=”pcc”>
        <limits .../>
        <backbone .../>
      </model>
    </curvature>
    <inertial>
      <mass value=”0.3”/>
    </inertial>
  </bendable>
  <ros2_control name=”softrobot”
       type=”system”>
    <hardware>
      <plugin>SoftRobotInterface</plugin>
      <param name=”model”>pcc</param>

         ⋮

    </hardware>
    <joint name=”Softrobot”>
      <command_interface name=”kappa” >
        <param name=”initial_value”>0</param>
        <param name=”max”>3</param>
      </command_interface>
      <command_interface name=”phi”>
        <param name=”initial_value”>0</param>
      </command_interface>
      <state_interface name=”kappa” />
      <state_interface name=”phi” />
    </joint>
  </ros2_control>
</robot>
```

Additionally, a controller model of the robot can be derived, which can be for parameterizing a ROS2 controller. To function properly, the robot controller needs additional information about the controllable aspects of a robot. Taking into account the robot developed, this would be κ and ϕ. In order to parameterize the controller, the information model provides an additional *ros_control* element, depicted in the lower half of Listing 1. Within the element, some parameters are provided that describe the hardware functionality, as well as the command and state interfaces. Command interfaces are used to signal the controllable HWI interfaces to the software controller. State interfaces mark sensor information generated as outputs. In the provided example, both κ and ϕ are command and state interfaces.

V. VALIDATION

The goal of the developed framework is a simplified handling of continuum robots, comparable to the straightforward implementation and parameterization of articulated robots using, for example, the ros2_control[1] framework. In order to validate the framework developed, the authors developed the SCR depicted in Figure 1 and implemented a control stack following the framework described in section IV. In the following, first, the developed robot and then the implemented control stack will be explained.

A. Development of a Modular Soft Continuum Robot

To validate the introduced framework, the modular SCR, depicted in Figure 1 has been developed. Using a modular segment-based approach, a continuum robot can be described through bending segments, similar to articulated robots described by joints and links. Hence, creating an adjustable SCR to validate the adaptability of the framework and U(C)RDF. The external functionalities of individual modular segments can be described in a standardized way, despite differences in actuation principles, sensing, and material behavior inside each segment. This enables a hardware abstraction of each segment by the U(C)RDF.

The whole robot consists of three detachable segments. Each segment employs intrinsic actuation and control, enabling modular coupling of segments while allowing for independent operation. A single segment measures $316\,\mathrm{mm}$ in length and $154\,\mathrm{mm}$ in diameter. Consequently, the complete three-segment robot reaches a total length of $948\,\mathrm{mm}$. Three evenly spaced tendons are placed around a central backbone, allowing internal control of the end effector relative to the base of the segment. This configuration allows for bending along two orthogonal axes to the backbone axis of up to $60°$, although torsional actuation of the end effector is not supported. Spacer disks mounted on the backbone ensure consistent tendon positioning during segment bending. Each segment is actuated by three servo motors, which are supplied via a main power line that runs through all segments. Controllers, servo motors, nylon tendons, and carbon fiber backbone are externally sourced, while all remaining structural components are 3D printed. For fabrication, Polyethylene Terephthalate Glycol (PETG) and thermoplastic polyurethane (TPU) materials were selected to combine structural rigidity with flexibility to improve collision tolerance. Control of the servo motors is achieved using an ESP32 microcontroller, communicating via Universal Asynchronous Receiver Transmitter (UART). Each ESP32 is equipped with a WiFi module, which enables communication with an external Raspberry Pi 4B using micro ROS. An external Raspberry Pi serves as the central control unit, coordinating high-level segment actuation and overall robot behavior.

As a mathematical model, the PCC model is used. This model is suitable for multi-segment continuum robots that provide a segment-wise curvature in a 2D plane, depicted in Figure 3. The PCC model forward kinematics transformation is based on a tuple of characteristic variables $k_i = [\kappa_i \; l_i \; \phi_i]^\top$ which describes the geometric relationship between two reference coordinate systems along the robot arm, similar to the forward kinematics of an articulated robot. These variables describe the curvature κ_i, the segment length l_i and the rotation of the plane of curvature ϕ_i of a segment of the robot. Figure 3 shows the PCC configuration space of a generalized three segment continuum robot.

Based on the configuration variables k_i, of each segment of the continuum robot, a transformation matrix ${}^jH_i$ can be formed with (1). Multiple transformations can be concate-

---

[1] https://github.com/ros-controls/ros2_control

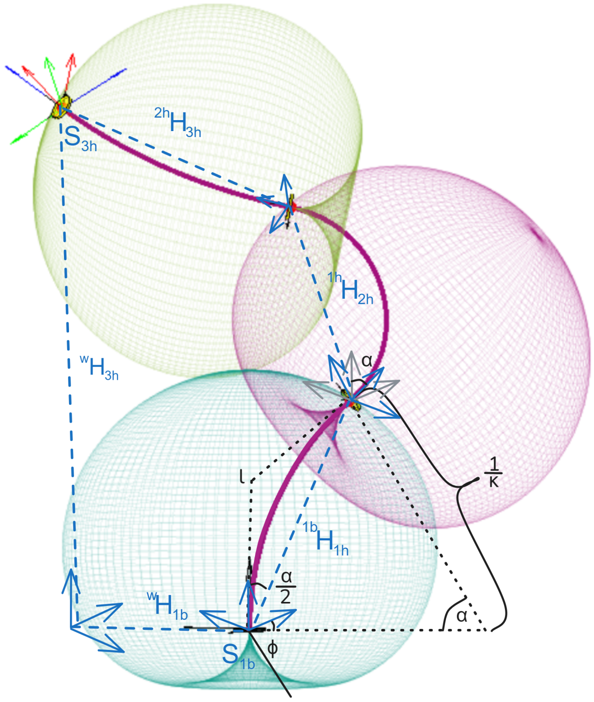


Fig. 3: Segmented continuous robot curvature

nated by multiplying the individual transformation matrices. In Figure 3 some example transformations are depicted in blue. Consequently, the transformation between the base frame S_{1b} and the top frame S_{3h} of the example robot in Figure 3 can be calculated by multiplying the tuples k_1 through k_4, i.e., the tuple of each segment.

Based on the mathematical model, a hardware-independent controller can be implemented. In order for the controller to identify the attached robot, the controller parameterization can be done using information models.

B. Implementation of the Control Stack

The robot developed is controlled using the software stack, depicted in Figure 4. The stack follows the framework of section IV and utilizes the introduced U(C)RDF to represent the robot hardware.

In order to validate the adaptive nature of the framework, two controllers have been implemented, a continuum joint trajectory controller (cJTC) and a continuum Feed Forward Controller (cFFC). Using the cFFC, a handheld controller, depicted in Figure 5 is used to change the segment curvature of the robot, where the change rate is determined by the controller-stick deflection. When using the cJTC, an application (top right) sends segment-wise target PCC configuration to the cJTC. Similarly to the JTC for articulated robot, the cJTC is used to up-sample the transition from a starting configuration to a target configuration to obtain a smooth motion. The cFFC takes the user input and applies incremental curvature changes to the robot. Using the controller and resource manager from the standard ros_control stack, each key-frame of the motion is sent to the motion controller of the corresponding segments, via the HWI. The HWI abstracts the tendon-driven segments of the robot and advertises command and state interfaces. All sensor information, such as the current curvature of a segment, is communicated to the continuum Joint State Broadcaster (cJSB) via the state interfaces. The cJSB can take the information and publish the current joint states into the ROS2 network using custom message types. A joint state consists of the aforementioned curvature κ and rotation of the curvature plane ϕ_i. The continuum Robot State Publisher (cRSP) calculates the transformations ${}^{j}H_i$ based on the joint states and publishes the result. Tools, like RViz can then display the robot as depicted in Figure 5.

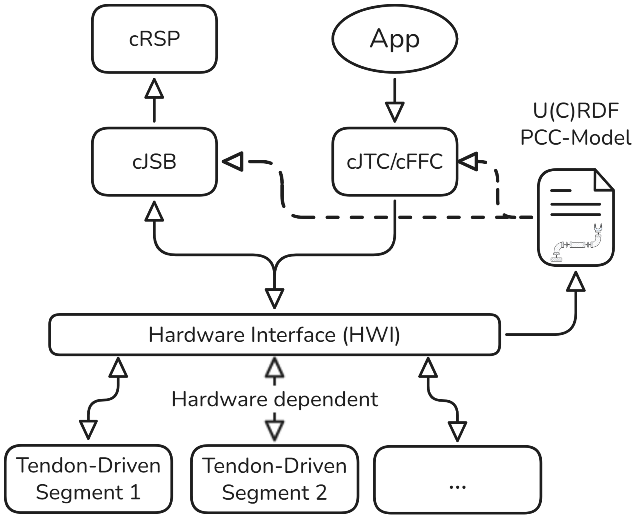


Fig. 4: Controller stack of the continuum robot

Within the software stack, the cJSB and cJTC are parameterized using the U(C)RDF, which describes the current SCR configuration.

Based on this implementation, a controller development process similar to that for articulated robots is possible.

In addition, the robot can be easily reconfigured. For example, the switch from two to three segments can be performed by providing the corresponding U(C)RDF to the controller.

All files, including robot code and step data, are available as open source under the MIT license via GitHub[2].

Current limitations

The current state of development can validate the usage of the U(C)RDF for parameterization of the controller and inverse kinematics. However, there are some limitations.

At the current time, the tool RViz is not able to depict bending curves. In order for RViz to display the robot, joints of type *floating* are used. These joint describe the transformation between the base and top link of each SCR segment. Therefore, the U(C)RDF is processed into a kind of pseudo-rigid body model of the robot using the spacer disks to visualize the curvature.

---

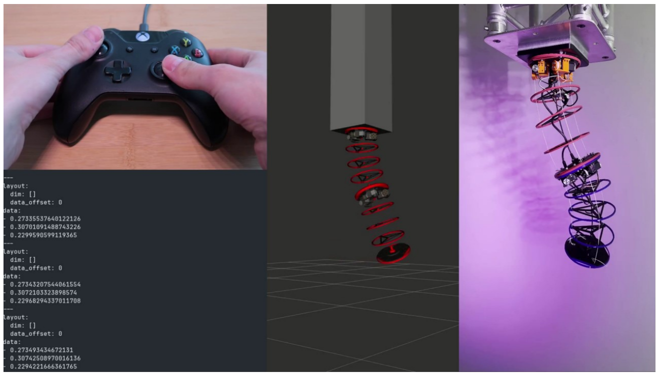
[2] https://github.com/KIT-IRS/Roberto

$$
{}^{j}H_i = \begin{bmatrix} jR_i & j\sigma_i \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{1}
$$

with $q_i = \frac{1}{\kappa_i}$:

$$
{}^{j}\sigma_i = \frac{l_i}{q_i} \begin{bmatrix} \cos\phi_i(1-\cos q_i) \\ \sin\phi_i(1-\cos q_i) \\ \sin q_i \end{bmatrix} \tag{2}
$$

$$
{}^{j}R_i = \begin{bmatrix} \cos^2\phi_i(\cos q_i - 1) + 1 & \sin\phi_i\cos\phi_i(\cos q_i - 1) & \cos\phi_i \sin q_i \\ -\sin\phi_i\cos\phi_i(\cos q_i - 1) & \sin^2\phi_i(\cos q_i - 1) + 1 & \sin\phi_i \sin q_i \\ -\cos\phi_i \sin q_i & -\sin\phi_i \sin q_i & \cos q_i \end{bmatrix} \tag{3}
$$



```
---
layout:
  dim: []
  data_offset: 0
data:
- 0.27335537640122126
- 0.30701091488743226
- 0.2299590599119365
---
layout:
  dim: []
  data_offset: 0
data:
- 0.27343207544061554
- 0.3072103323898574
- 0.22968294337011708
---
layout:
  dim: []
  data_offset: 0
data:
- 0.273493434672131
- 0.30742508970016136
- 0.2294221666361765
```

Fig. 5: Controller usage of the continuum robot

VI. SUMMARY AND OUTLOOK

In this contribution, an engineering framework for the development of SCR was introduced. The framework adapts the ros_control stack with its integrated HWI abstraction satisfying REQ 1. For parameterization of hardware-independent software components, the URDF information model was extended to accommodate for SCRs fulfilling REQ 2. The extended U(C)RDF can be used to parameterize controllers and other software that utilizes system information. In order to maintain an open approach, the *model* element of the U(C)RDF can be defined for varying mathematical approaches, fulfilling REQ 3. The novel information model provides basically the same benefits as its rigid-body counterpart.

The next steps are to implement other mathematical models and extend the optional elements of the U(C)RDF. Furthermore, some complex controller, such as an impedance controller, would greatly increase the usability of the robot.

REFERENCES

[1] Apurba Das and M. Nabi. “A Review on Soft Robotics: Modeling, Control and Applications in Human-Robot Interaction”. In: *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. Oct. 2019, pp. 306–311. DOI: `10.1109/ICCCIS48478.2019.8974461`.

[2] International Organization for Standardization. *ISO/TS 15066:2016*. https://www.iso.org/standard/62996.html.

[3] Yanmei Wang et al. “Advancements in Soft Robotics: A Comprehensive Review on Actuation Methods, Materials, and Applications”. In: *Polymers* 16.8 (Apr. 2024), p. 1087. ISSN: 2073-4360. DOI: `10.3390/polym16081087`. pmid: `38675005`.

[4] Nazek El-Atab et al. “Soft Actuators for Soft Robotic Applications: A Review”. In: *Advanced Intelligent Systems* 2.10 (2020), p. 2000128. ISSN: 2640-4567. DOI: `10.1002/aisy.202000128`.

[5] Chiwon Lee et al. “Soft Robot Review”. In: *International Journal of Control, Automation and Systems* 15.1 (Feb. 2017), pp. 3–15. ISSN: 2005-4092. DOI: `10.1007/s12555-016-0462-3`.

[6] Nancy Velásquez Villagrán et al. “Standardization: A Key Factor of Industry 4.0”. In: *2019 Sixth International Conference on eDemocracy & eGovernment (ICEDEG)*. Apr. 2019, pp. 350–354. DOI: `10.1109/ICEDEG.2019.8734339`.

[7] Lars Kunze, Tobias Roehm, and Michael Beetz. “Towards Semantic Robot Description Languages”. In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 5589–5595. DOI: `10.1109/ICRA.2011.5980170`.

[8] Scott Graham, Girish Baliga, and P. R. Kumar. “Abstractions, Architecture, Mechanisms, and a Middleware for Networked Control”. In: *IEEE Transactions on Automatic Control* 54.7 (July 2009), pp. 1490–1503. ISSN: 1558-2523. DOI: `10.1109/TAC.2009.2022094`.

[9] Linus Witucki et al. “Introduction of a Unified Robot Integration Package”. In: *at - Automatisierungstechnik* 73.5 (May 2025), pp. 288–300. ISSN: 2196-677X. DOI: `10.1515/auto-2025-0007`.

[10] Daniella Tola and Peter Corke. “Understanding URDF: A Dataset and Analysis”. In: *IEEE Robotics and Automation Letters* 9.5 (May 2024), pp. 4479–4486. ISSN: 2377-3766, 2377-3774. DOI: `10.1109/LRA.2024.3381482`.

[11] Matteo Russo et al. “Continuum Robots: An Overview”. In: *Advanced Intelligent Systems* 5.5 (2023), p. 2200367. ISSN: 2640-4567. DOI: `10.1002/aisy.202200367`.

[12] Liyu Wang and Fumiya Iida. “Deformation in Soft-Matter Robotics: A Categorization and Quantitative

Characterization". In: *IEEE Robotics & Automation Magazine* 22.3 (Sept. 2015), pp. 125–139. ISSN: 1558-223X. DOI: `10.1109/MRA.2015.2448277`.

[13] Weiqiang Dou et al. "Soft Robotic Manipulators: Designs, Actuation, Stiffness Tuning, and Sensing". In: *Advanced Materials Technologies* 6.9 (2021), p. 2100018. ISSN: 2365-709X. DOI: `10.1002/admt.202100018`.

[14] Costanza Armanini et al. "Soft Robots Modeling: A Structured Overview". In: *IEEE Transactions on Robotics* 39.3 (June 2023), pp. 1728–1748. ISSN: 1941-0468. DOI: `10.1109/TRO.2022.3231360`.

[15] Cosimo Della Santina, Christian Duriez, and Daniela Rus. *Model Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges*. Oct. 2021.

[16] Sachin Chitta, Eitan Marder-Eppstein, and Wim Meeussen. "Ros_control: A Generic and Simple Control Framework for ROS". In: *The Journal of Open Source Software* (2017). DOI: `10.21105/joss.00456`.

[17] Daniella Tola and Peter Corke. *Understanding URDF: A Survey Based on User Experience*. Mar. 2023. arXiv: `2302.13442 [cs]`.

[18] Irati Zamalloa et al. "HRIM: The Hardware Robot Information Model". In: *ISR 2018; 50th International Symposium on Robotics*. June 2018, pp. 1–4.

[19] Hai-Jun Su. "A Pseudorigid-Body 3R Model for Determining Large Deflection of Cantilever Beams Subject to Tip Loads". In: *Journal of Mechanisms and Robotics* 1.021008 (Jan. 7, 2009). ISSN: 1942-4302. DOI: `10.1115/1.3046148`. URL: `https://doi.org/10.1115/1.3046148` (visited on 02/16/2024).

[20] Ibrahim A. Seleem et al. "Demonstration-Guided Pose Planning and Tracking for Multi-Section Continuum Robots Considering Robot Dynamics". In: *IEEE Access* 7 (2019), pp. 166690–166703. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2019.2953122`.

[21] Anup Teejo Mathew et al. "Reduced Order Modeling of Hybrid Soft-Rigid Robots Using Global, Local, and State-Dependent Strain Parameterization". In: *International Journal of Robotics Research* (2024). ISSN: 0278-3649. DOI: `10.1177/02783649241262333`.