Using Clarke Transform to Create a Framework on the Manifold: From Sampling via Trajectory Generation to Control

Reinhard M. Grassmann and Jessica Burgner-Kahrs

Abstract— We present a framework based on Clarke coordinates for spatial displacement-actuated continuum robots with an arbitrary number of joints. This framework consists of three modular components, *i.e.*, a planner, trajectory generator, and controller defined on the manifold. All components are computationally efficient, compact, and branchless, and an encoder can be used to interface existing framework components that are not based on Clarke coordinates. We derive the relationship between the kinematic constraints in the joint space and on the manifold to generate smooth trajectories on the manifold. Furthermore, we establish the connection between the displacement constraint and parallel curves. To demonstrate its effectiveness, a demonstration in simulation for a displacement-actuated continuum robot with four segments is presented.

I. Introduction

Soft and continuum robots must necessarily move from a current configuration to a desired configuration to accomplish meaningful real-world tasks in medical and industrial applications. Thus, a minimal framework should consist of three parts; planner, trajectory generator, and controller. However, developed frameworks are tailored to a specific robot morphology rendering some, if not all, components of the framework unsuitable for other robot morphologies. This limits the reusability and rapid advancement in the field. Overcoming that by providing a modular framework, where each component has the same interface, has obvious benefits, *e.g.*, (i) interpretability and safety due to a divide-and-conquer approach, (ii) compatibility due to the same interface, (iii) flexibility in changing and testing components, (vi) agility in developing, improving, and testing of separate components, and (v) empowering the field by sharing and reusing components. Overall, an increase in cost-efficiency and a decrease in the number of times the wheel needs to be reinvented can be expected. Therefore, providing a general principle to create frameworks is a necessary step to move forward together, rather than just improving one particular robot morphology.

As a candidate for an interface between the components, different improved state representations [1], [2], [3], [4], [5] have been introduced in recent years. It has been shown that using improved state representations has decisive advantages. Della Santina *et al.* [1] discuss and analyze the main issues arising from the use of commonly used parameterization, *i.e.*, curvature κ and bending plane angle θ used for constant

All authors are with Continuum Robotics Laboratory, Department of Mathematical and Computational Sciences, University of Toronto, Mississauga, ON L5L 1C6, Canada `reinhard.grassmann@utoronto.ca`

curvature models. The derived improved state representation [1] for $n = 4$ number of joints shows superb performance on a model-based control application. Allen *et al.* [2] introduces improved state representations for $n = 3$ and $n = 4$ number of joints. They propose closed-form and singularity-free forward kinematics as well as Jacobian-based inverse kinematics mitigating the disadvantages of previous approaches. Dian *et al.* [4] proposed an improved state representation that can be seen as an adaptation of improved state representation by Della Santina *et al.* [1] to $n = 3$. As mentioned by Dupont *et al.* [6], different representations exist and among them, that utilize Cartesian curvature components, *i.e.*, κ_x and κ_y, have the advantage of avoiding a parametric singularity. The connection between the Cartesian curvature components and all improved state representations is established by Grassmann *et al.* [5].

Grassmann *et al.* [5] links the displacement-actuated joints and the displacement constraint to electrical currents and Kirchhoff's current law, respectively. This and the reformulation of the joint representation lead to the use of the Clarke transformation matrix that allows the disentanglement of the displacement constraint. As a result, an improved state representation called Clarke coordinates is derived unifying previous improved state representations [1], [2], [3], [4], [5] and generalizing them to $n \geq 3$. Since constant curvature [7] is not enforced during their derivation, the Clarke coordinates are not limited to constant curvature assumption. In our work [8], we show that the analogy to the Clarke transformation matrix is not necessary to derive the so-called Clarke transform. This allows us to consider arbitrary joint location distribution on the constant cross-section of a displacement-actuated continuum robot.

In general, the derived approaches based on improved state representations are closed-from, compact, computationally efficient, and singularity-free. This makes the Clarke coordinates are desirable representation and unified language to interface all components within a framework. Due to the recent introduction of the Clarke transform, only sampling methods [5], [8] and controllers [5], [8] have been proposed. While previous sampling methods and controllers can be used as components in a framework, a trajectory generator has yet to be formulated using Clarke coordinates.

Throughout this work, we consider a displacement-actuated continuum robot defined by kinematic design parameters consisting of segment length $l = const.$, distance d_i, angle ψ_i, and number of joints n. For n equally distributed

joints, the polar coordinate given by

$$\psi_i = \frac{2\pi}{n}(i-1) \quad \text{and} \quad d_i = d = const. \tag{1}$$

describes the location of the ith joint in the cross-section. The kinematic design parameters fully describe the displacement-actuated continuum robot in the kinematic sense encompassing a large class of soft robots and continuum robots.

In the presented work, we aim to shift the paradigm towards a unified approach by introducing a framework, where each of the components is based on the Clarke coordinates of a 2 dof manifold. In particular, the following contributions are made:

- Presentation of a framework completely based on the Clarke coordinates
- Introduction of a smooth trajectory generator that specifies and generates trajectories on the 2 dof manifold
- Derivation of the kinematics constraints for trajectories on the manifold
- Connection between displacement constraint and parallel curves also known as offset curves

II. CLARKE TRANSFORM

In this section, we provide a self-contained and brief description of the Clarke transform, the relationship to arc parameters, and virtual displacement. The description of virtual displacement is expanded to the connection between the displacement constraint and parallel curves. Furthermore, we briefly state the encoder-decoder architecture first derived in our previous work [8].

A. Clarke Transform for Displacement-Actuated Joint

Proposed by Grassmann *at al.* [5], a vector with values of each displacement-actuated joint ρ_i can be represented as

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} = \begin{bmatrix} \rho_{\text{Re}}\cos(\psi_1) + \rho_{\text{Im}}\sin(\psi_1) \\ \rho_{\text{Re}}\cos(\psi_2) + \rho_{\text{Im}}\sin(\psi_2) \\ \vdots \\ \rho_{\text{Re}}\cos(\psi_n) + \rho_{\text{Im}}\sin(\psi_n) \end{bmatrix} \in \mathbf{Q}, \tag{2}$$

where $\mathbf{Q} \subset \mathbb{R}^n$ is the joint space. All joint values in (2) are interdependent and constrained obeying the displacement constraint given by

$$\sum_{i=1}^{n} \rho_i = 0. \tag{3}$$

The Clarke coordinates [5], *i.e.*, ρ_{Re} and ρ_{Im} in (2), can be combined into a vector as well, *i.e.*,

$$\overline{\boldsymbol{\rho}} = \begin{bmatrix} \rho_{\text{Re}} \\ \rho_{\text{Im}} \end{bmatrix}. \tag{4}$$

Both representations can be transformed into each other via

$$\overline{\boldsymbol{\rho}} = \boldsymbol{M}_{\mathcal{P}}\boldsymbol{\rho} \quad \text{and} \tag{5}$$

$$\boldsymbol{\rho} = \boldsymbol{M}_{\mathcal{P}}^{-1}\overline{\boldsymbol{\rho}}, \tag{6}$$

where the generalized Clarke transformation matrix $\boldsymbol{M}_{\mathcal{P}}$ is

$$\boldsymbol{M}_{\mathcal{P}} = \frac{2}{n} \begin{bmatrix} \cos(\psi_0) & \cos(\psi_1) & \cdots & \cos(\psi_n) \\ \sin(\psi_0) & \sin(\psi_1) & \cdots & \sin(\psi_n) \end{bmatrix}, \tag{7}$$
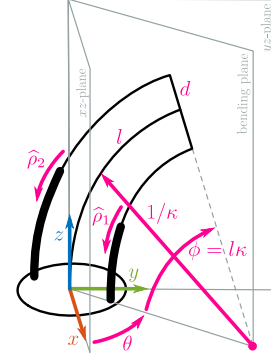

Fig. 1. Surrogate displacement-actuated continuum robot.

which holds for symmetrically arranged joints [5], [8]. By factor out (4) in (2), its inverse $\boldsymbol{M}_{\mathcal{P}}^{-1}$ can be found being

$$\boldsymbol{M}_{\mathcal{P}}^{-1} = \begin{bmatrix} \cos(\psi_1) & \sin(\psi_1) \\ \cos(\psi_2) & \sin(\psi_2) \\ \vdots & \vdots \\ \cos(\psi_n) & \sin(\psi_n) \end{bmatrix}, \tag{8}$$

where (8) is the right-inverse of non-square matrix (7).

B. Arc Parameters

Utilizing the constant curvature assumption [7], it is possible to identify the arc parameters, *i.e.*, curvature κ and bending-plane angle θ. Derived by Grassmann *et al.* [5], this relation between Clarke coordinates and arc parameters is

$$\overline{\boldsymbol{\rho}} = \begin{bmatrix} \rho_{\text{Re}} \\ \rho_{\text{Im}} \end{bmatrix} = \begin{bmatrix} dl\kappa\cos(\theta) \\ dl\kappa\sin(\theta) \end{bmatrix}. \tag{9}$$

Note that l is constant and not considered as arc parameters in this context. As hinted by the used analogy and naming convention by Grassmann *et al.* [5], we can tread (9) as a complex number embedded in $\mathbb{R}^2$. The modulus and argument of (9) is given by

$$|\overline{\boldsymbol{\rho}}| = dl\kappa \quad \text{and} \tag{10}$$

$$\arg\overline{\boldsymbol{\rho}} = \theta. \tag{11}$$

As can be seen, (10) is the scaled curvature κ or the scaled bending angle $\phi = l\kappa$, whereas (11) is bending-plane angle θ. Other arguments and formulæ are presented in [5], [9]. To further investigate the geometric meaning of $|\overline{\boldsymbol{\rho}}|$, we take a look at the concept of virtual displacement [10], [5] and the displacement constraint (3).

C. Virtual Displacement and Displacement Constraint

We consider a displacement-actuated continuum robot with differential actuation and an additional rotational base, see Fig. 1. Due to the rotation θ, the joint representation (2) does not apply to this toy example. Instead, we denote the value of each displacement-actuated joint as $\widehat{\rho}_1$ and $\widehat{\rho}_2$, respectively.

As mentioned by Grassmann *et al.*, the maximum and minimum values are aligned with the bending plane, assuming infinite large n, *i.e.,* infinite many joints around the center-line. In our case, $\widehat{\rho}_1$ is associated with the maximum value, whereas $\widehat{\rho}_2$ is associated with the minimum value.
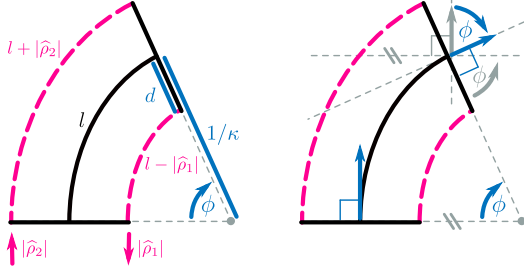
Fig. 2. Arc length of the surrogate displacement-actuated continuum robot. For convenience, the absolute values of $\widehat{\rho}_1$ and $\widehat{\rho}_2$ are used. The bending angle ϕ is identical to the tip orientation. Note that the bending plane angle θ is other tip orientation. Both angles are well-known and reflected in the rotation matrix of the kinematics for the constant curvature model.

Due to (3), $\widehat{\rho}_2 = -\widehat{\rho}_1$. Furthermore, assuming that $\widehat{\rho}_1$ and $\widehat{\rho}_2$ relates to the Clarke coordinates, we can immediately conclude $\widehat{\rho}_1 = |\overline{\boldsymbol{\rho}}|$ and $\widehat{\rho}_2 = -|\overline{\boldsymbol{\rho}}|$. This observation aligns with the concept of virtual displacements [10], [5].

We gain more insight by looking into the planar representation utilizing the bending plane illustrated in Fig. 2, *cf.* Fig. 1. Assuming constant curvature, we can relate each arc length to each other, *e.g.,* $l - \widehat{\rho}_1 = (1/\kappa - d)\phi$. Solving for $\widehat{\rho}_1$ results in $\widehat{\rho}_1 = |\overline{\boldsymbol{\rho}}|$, where $\phi = l\kappa$ is used. Furthermore, we can identify that bending plane angle ϕ relates to the tip orientation, which is a known observation [11]. Note that constant curvature implicit assumes that the line associated with $\widehat{\rho}_1$ and $\widehat{\rho}_2$, respectively, is a fully constrained path, where, loosely speaking, the distance d to the backbone is constant. In fact, each line is a parallel curve, also known as an offset curve. A parallel curve depends only on the local curvature and its distance to the curve [12]. In Fig. 2, two parallel curves with distance d are depicted.

Using the formula for the arc length of a parallel curve stated by *Loria* [13], a property can be derived stating that the arc length of the curve is equal to the mean of the arc lengths of both parallel curves. In our case, using the quantities in Fig. 2 with sign convention depicted in Fig. 1 leads to

$$ l = ((l - \widehat{\rho}_1) + (l - \widehat{\rho}_2))/2 = l - (\widehat{\rho}_1 + \widehat{\rho}_2)/2. $$

Further algebraic manipulation results in $\widehat{\rho}_1 + \widehat{\rho}_2 = 0$ which is precisely the displacement constraint (3) for $n = 2$. The derivation using parallel curves shows that (3) is not a consequence of the constant curvature assumption. Figure 3 provides additional examples for visual aid. Moreover, since the displacement $|\overline{\boldsymbol{\rho}}|$ depends on the angle ϕ (obtained by integrating the local curvature along the curve) and the distance d, *cf.*, [13] and [11], we can write

$$ |\overline{\boldsymbol{\rho}}| = d\phi \tag{12} $$

instead of (10), where ϕ is the tip orientation and applicable beyond commonly used constant curvature assumption.

Moreover, the modulus of $\overline{\boldsymbol{\rho}}$ can be defined independently of the constant curvature assumption [7] and d. We can find

$$ |\overline{\boldsymbol{\rho}}| = \sqrt{2/n}\sqrt{\boldsymbol{\rho}^\top \boldsymbol{\rho}} \tag{13} $$

by equating the dot products of (4) and (6). Furthermore, the property $\boldsymbol{M}_{\mathcal{P}}^\top = (2/n)\boldsymbol{M}_{\mathcal{P}}^{-1}$ is used, which can be derived by inspection of (7) and (8), *cf.* [9].
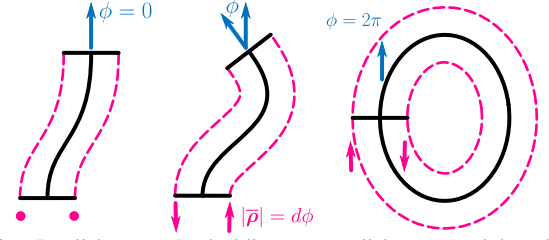

Fig. 3. Parallel curves. Dashed lines are parallel curves and the sold lines between them are the center-line. The displacement depends on the angle ϕ and the distance d. (left) 2 dof case of Cavalieri's principle, where the tip and base orientations are identical, resulting in $\widehat{\rho}_i = 0$. (middle) parallel curves resulting in $\widehat{\rho}_i = \pm d\phi$. (right) full circle, where the tip and base poses are identical, resulting in $\widehat{\rho}_i = \pm 2\pi d$.
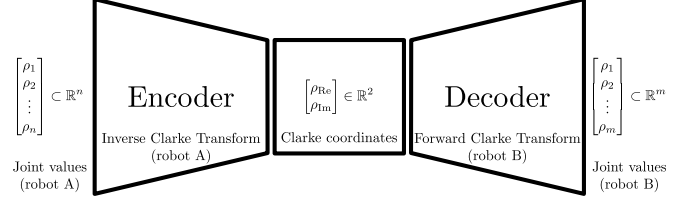

Fig. 4. Encoder-decoder architecture (image credit [8]). Joint values of one robot type (robot A) can be transformed into joint values of a different robot type (robot B). The latent space representation is encoded as Clarke coordinates. Note that the compression is lossless allowing joint values to be uniquely reconstructed from the Clarke coordinates.

D. Encoder-Decoder Architecture

The encoder-decoder architecture first derived in our work [8] utilizes the fact that the Clarke coordinates $\overline{\boldsymbol{\rho}}$ can be seen as latent space representation. In short, we can use (5) to map $\boldsymbol{\rho}_{\text{(robot A)}}$ of *robot A* to the same $\overline{\boldsymbol{\rho}}$ that are obtained using (6) form $\boldsymbol{\rho}_{\text{(robot B)}}$ of *robot B*. This leads to

$$ \boldsymbol{\rho}_{\text{(robot A)}} = \underbrace{\boldsymbol{M}_{\mathcal{P}\text{(robot A)}}}_{\text{decoder}} \underbrace{\boldsymbol{M}_{\mathcal{P}}^{-1}{}_{\text{(robot B)}}}_{\text{encoder}} \boldsymbol{\rho}_{\text{(robot B)}}, \tag{14} $$

which we call *encoder-decoder architecture* [8]. Figure 4 visualizes this approach. A more general approach is presented in [8] as well.

III. FEASIBLE JOINT VALUES

We directly sample Clarke coordinates $\overline{\boldsymbol{\rho}}_{\mathcal{U}}$ using a rejection-free sampling method [5]. Considering the argument (11) and modulus (12) of $|\overline{\boldsymbol{\rho}}|$, we only need to sample

$$ |\overline{\boldsymbol{\rho}}_{\mathcal{U}}| = \phi_{\max} d \, \mathcal{U}[0;1] \quad \text{and} \quad \arg \overline{\boldsymbol{\rho}}_{\mathcal{U}} = \theta_{\max} \mathcal{U}[-1;1), $$

where $\phi_{\max}$ and $\theta_{\max}$ define the maximum value for the tip orientations, see Fig. 1 and Fig. 2. Each value is drawn from a uniform distribution $\mathcal{U}$. To match the application at hand, other distributions and ranges might be considered. Furthermore, if more samples are required, this sampling method can be vectorized.

Setting $\phi_{\max}$ is more convenient than defining a maximum curvature $\kappa_{\max}$. The convenience lies in relation to the tip orientation, recap Fig. 2, and the visualization using circles, where an angle describes a circle more intuitively than the corresponding curvature. Recap the relation to virtual displacement in Sec. II-C and Fig. 3, it also circumvents the relation to constant curvature, *i.e.,* $\phi_{\max} = l\kappa_{\max}$.

IV. TRAJECTORY GENERATION

In this paper, we focus on simple polynomial trajectories to lay out the basics of trajectory generation using Clarke coordinates. First, we establish a simple point-to-point trajectory for the displacement. Afterward, a point-to-point trajectory generator is proposed for Clarke coordinates. Finally, the connection between both trajectory generators is pointed out.

A. Trajectory Generation in Joint Space

Consider a $\mathcal{C}^3$ smooth path primitive $s(\tau) \in [0,1]$ and its derivative *w.r.t.* the time primitive $\tau \in [0,1]$ given by

$$
\begin{aligned}
s(\tau) &= -20\tau^7 + 70\tau^6 - 84\tau^5 + 35\tau^4, \\
s'(\tau) &= -140\tau^6 + 420\tau^5 - 420\tau^4 + 140\tau^3 \quad , \text{ and} \\
s''(\tau) &= -840\tau^5 + 2100\tau^4 - 1680\tau^3 + 420\tau^2,
\end{aligned}
$$

respectively [14]. The duration T of the trajectory is included in the definition of the time primitive $\tau = t/T$. Using the boundaries $s(1) = 1$ and $s(0) = s'(0) = s'(1) = s''(0) = s''(0) = 0$, the path primitive and its above derivative can be clipped to define trajectories for all t. For a given $\boldsymbol{\rho}_{\text{start}} \in \mathbf{Q}$ and $\boldsymbol{\rho}_{\text{goal}} \in \mathbf{Q}$, we can now define a simple trajectory as

$$
\boldsymbol{\rho}_{\text{d}}(t) = \boldsymbol{\rho}_{\text{start}} + \left(\boldsymbol{\rho}_{\text{goal}} - \boldsymbol{\rho}_{\text{start}}\right) s(t). \tag{15}
$$

It is important to show that, at every time step, a valid displacement is generated. For this, checking (3) results in

$$
\begin{aligned}
\sum_{i=1}^{n} \rho_{i,\text{d}}(t) &= \sum_{i=1}^{n} \left(\rho_{i,\text{start}} + \left(\rho_{i,\text{goal}} - \rho_{i,\text{start}}\right) s(t)\right) \\
&= \sum_{i=1}^{n} \rho_{i,\text{start}} + s(t) \sum_{i=1}^{n} \rho_{i,\text{goal}} - s(t) \sum_{i=1}^{n} \rho_{i,\text{start}} \\
&= 0,
\end{aligned}
$$

because $\boldsymbol{\rho}_{\text{start}} \in \mathbf{Q}$ and $\boldsymbol{\rho}_{\text{goal}} \in \mathbf{Q}$ obey (3). This simple check shows that $\boldsymbol{\rho}_{\text{d}}(t) \in \mathbf{Q}$ for all t.

For convenience, we define the maximal absolute value of all entries in $\boldsymbol{\rho}_{\text{goal}} - \boldsymbol{\rho}_{\text{start}}$ as

$$
\Delta\rho = \|\boldsymbol{\rho}_{\text{goal}} - \boldsymbol{\rho}_{\text{start}}\|_{\infty}, \tag{16}
$$

which is a common choice for trajectory generation of positions in Task space. The quantity $\Delta\rho$ is important to synchronize all n trajectories of (15).

To consider the kinematic constraints given by v_{max} for the maximal velocity and a_{max} for the maximal acceleration, each time at the maximal value of $s'(t)$ and $s''(t)$ is determined. By choosing a large enough duration T, the kinematic constraints can be considered. Therefore, the duration T is

$$
T = \max\left\{ \frac{35\Delta\rho}{16v_{\text{max}}}, \sqrt{\frac{84\Delta\rho}{5\sqrt{5}a_{\text{max}}}}, T_{\text{user}} \right\}, \tag{17}
$$

where T_{user} is defined by a user. For $T_{\text{user}} = 0$, the executed trajectory (15) always reaches one of the kinematic constraints. Furthermore, due to the consideration of (16) in (17), all trajectories in (15) are synchronized, recall $\tau = t/T$.

B. Trajectory Generation using Clarke Coordinates

Until now, all trajectories generated with geometric scaling using path primitives $s(t)$ have shown to be feasible. Mapping the trajectory (15) onto the manifold leads to

$$
\begin{aligned}
\overline{\boldsymbol{\rho}}_{\text{d}}(t) &= \boldsymbol{M}_{\mathcal{P}} \boldsymbol{\rho}_{\text{d}}(t) \\
&= \boldsymbol{M}_{\mathcal{P}} \left(\boldsymbol{\rho}_{\text{start}} + \left(\boldsymbol{\rho}_{\text{goal}} - \boldsymbol{\rho}_{\text{start}}\right) s(t)\right) \\
&= \boldsymbol{M}_{\mathcal{P}} \boldsymbol{\rho}_{\text{start}} + s(t) \boldsymbol{M}_{\mathcal{P}} \boldsymbol{\rho}_{\text{goal}} - s(t) \boldsymbol{M}_{\mathcal{P}} \boldsymbol{\rho}_{\text{start}} \\
&= \overline{\boldsymbol{\rho}}_{\text{start}} + s(t) \overline{\boldsymbol{\rho}}_{\text{goal}} - s(t) \overline{\boldsymbol{\rho}}_{\text{start}},
\end{aligned}
$$

where the last step results in the formulation

$$
\overline{\boldsymbol{\rho}}_{\text{d}}(t) = \overline{\boldsymbol{\rho}}_{\text{start}} + \left(\overline{\boldsymbol{\rho}}_{\text{goal}} - \overline{\boldsymbol{\rho}}_{\text{start}}\right) s(t) \tag{18}
$$

for a trajectory defined on the manifold. Note that (15) is a straight line in the joint space $\mathbf{Q}$ and (18) is also a straight line on the manifold. The derivation of (18) shows that it is consistent with a trajectory (15) defined in $\mathbf{Q}$.

To consider the kinematic constraints defined on the manifold, we can adapt the time T by substituting

$$
\begin{aligned}
\Delta\rho &\longleftarrow \Delta\bar{\rho}, \\
v_{\text{max}}\rho &\longleftarrow \bar{v}_{\text{max}} \quad , \text{ and} \\
a_{\text{max}} &\longleftarrow \bar{a}_{\text{max}}
\end{aligned}
$$

into (17), where the displacement $\Delta\bar{\rho}$ is defined by

$$
\Delta\bar{\rho} = \|\overline{\boldsymbol{\rho}}_{\text{goal}} - \overline{\boldsymbol{\rho}}_{\text{start}}\|_{\infty}. \tag{19}
$$

While the definition of the kinematic constraints defined in the joint space $\mathbf{Q}$ are meaningful due to the physical relation to the displacement-actuated continuum robot, the equivalent kinematic constraints used for (18), *i.e.*, $\bar{v}_{\text{max}}$ and $\bar{a}_{\text{max}}$ are obscure. Therefore, there is a need to establish the relation between those two sets of kinematic constraints.

C. Relation between the Trajectory Generators

The physical meaning of the kinematic constraints defined in $\mathbf{Q}$ depends on the used displacement-actuated continuum robot. For example, a tendon-driven continuum robot uses tendons that actuated with a winch or a threaded drum attached to a motor. Another exemplary displacement-actuated continuum robot is one with multiple secondary backbones actuated with linear actuators. In those cases, the physical meaning of v_{max} and a_{max} are straightforward. For a soft robot actuated with bellows, the relationship to v_{max} and a_{max} is slightly more complicated.

To relate both sets of kinematic constraints, we find the extrema of time derivatives of (15) and (18), respectively. Since the trajectory is synchronized, it is sufficient to look at the ith entry of (15) and the real part of (18). We can find

$$
\begin{aligned}
\max \frac{\mathrm{d}}{\mathrm{d}t} \rho_{i,\text{d}}(t) &= \frac{\Delta\rho}{T} \frac{35}{16} = v_{\text{max}} \quad \text{and} \\
\max \frac{\mathrm{d}^2}{\mathrm{d}^2 t} \rho_{i,\text{d}}(t) &= \frac{\Delta\rho}{T^2} \frac{84}{5\sqrt{5}} = a_{\text{max}}
\end{aligned}
$$

for (15), whereas the extrema for (18) are

$$\max \frac{\mathrm{d}}{\mathrm{d}t}\rho_{\mathrm{Re,d}}(t) = \frac{\Delta\bar{\rho}}{T}\frac{35}{16} = \bar{v}_{\max} \quad \text{and}$$

$$\max \frac{\mathrm{d}^2}{\mathrm{d}^2 t}\rho_{\mathrm{Re,d}}(t) = \frac{\Delta\bar{\rho}}{T^2}\frac{84}{5\sqrt{5}} = \bar{a}_{\max}.$$

Afterward, T is equated and the resulting ratios are rearranged. The relation between the kinematic constraint is

$$\bar{v}_{\max} = v_{\max}\frac{\Delta\bar{\rho}}{\Delta\rho} \quad \text{and} \quad \bar{a}_{\max} = a_{\max}\frac{\Delta\bar{\rho}}{\Delta\rho},$$

respectively. However, the above relations rely on the evaluation of (16) and (19). To overcome this, we can exploit

$$\|\boldsymbol{x}\|_\infty \le \|\boldsymbol{x}\|_2 \le \sqrt{n}\|\boldsymbol{x}\|_\infty$$

for any vector $\boldsymbol{x}$. Now, we can utilize (13) and derive lower and upper bounds. The lower bounds are given by

$$\bar{v}_{\max} \ge \sqrt{\frac{2}{n}}\frac{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_\infty}{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_2}v_{\max} \ge \frac{\sqrt{2}}{n}v_{\max} \quad \text{and}$$

$$\bar{a}_{\max} \ge \sqrt{\frac{2}{n}}\frac{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_\infty}{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_2}a_{\max} \ge \frac{\sqrt{2}}{n}a_{\max}$$

Since lower values for $\bar{v}_{\max}$ and $\bar{a}_{\max}$ will satisfy the kinematic constraints in joint space, we set

$$\bar{v}_{\max} \leftarrow \sqrt{\frac{2}{n}}\frac{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_\infty}{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_2}v_{\max} \quad \text{and} \tag{20}$$

$$\bar{a}_{\max} \leftarrow \sqrt{\frac{2}{n}}\frac{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_\infty}{\|\bar{\boldsymbol{\rho}}_{\mathrm{goal}} - \bar{\boldsymbol{\rho}}_{\mathrm{start}}\|_2}a_{\max}, \tag{21}$$

allowing to define the kinematic constraints for (18).

V. CONTROL

The Clarke transform enables the synthesis of linear controllers that utilize the $2\,\mathrm{dof}$ manifold. A potential controller scheme could be a linear controller with two degrees of freedom. Grassmann *et al.* proposed two independent proportional feedback controllers with pre-compensation based on work by Morin & Samon [15]. In our work [8], two independent PD controllers are utilized, where the controller gains are defined as $\boldsymbol{K}_\mathrm{P} = k_\mathrm{P}\boldsymbol{I}_{2\times2}$ and $\boldsymbol{K}_\mathrm{D} = k_\mathrm{D}\boldsymbol{I}_{2\times2}$. Non-diagonal $\boldsymbol{K}_\mathrm{P} \in \mathbb{R}^{2\times2}$ and $\boldsymbol{K}_\mathrm{D} \in \mathbb{R}^{2\times2}$ are also possible.

Figure 5 illustrates a general controller scheme using parts of the encoder-decoder architecture (14). In this general framework, controllers beyond the PID controller can be used including non-linear controller schemes. In a framework solely based on Clarke coordinates, the encoder on the left side depicted in Fig. 5 can be omitted, where the desired Clarke coordinates are provided by a trajectory generator. However, it can be used as an interface for external frameworks outputting $\boldsymbol{\rho}$ and its time derivatives.
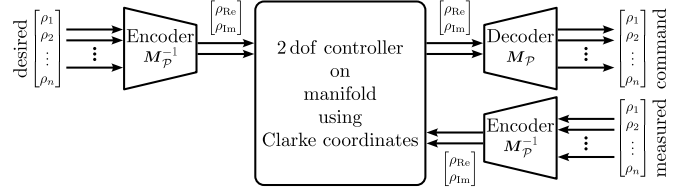

Fig. 5. Controller scheme utilizing Clarke transform. For the sake of compactness, we omit potential derivative terms such as velocity.

TABLE I
PARAMETERS OF THE FRAMEWORK.

Variable	Value	Unit	Description
$\phi_{\max}$	$2\pi/3$	rad	Related to the maximum curvature $\phi_{\max} = l\kappa_{\max}$
$\theta_{\max}$	π	rad	Symmetric maximum bending plane angle
$v_{\max}$	0.01	m/s	Maximum velocity of $\rho_i(t)$
$a_{\max}$	0.01	m/s^2	Maximum acceleration of $\rho_i(t)$
$\bar{v}_{\max}$	–	m/s	Maximum velocity of $\bar{\boldsymbol{\rho}}$ is predefined by $v_{\max}$ and (20)
$\bar{a}_{\max}$	–	m/s^2	Maximum acceleration of $\bar{\boldsymbol{\rho}}$ is predefined by $a_{\max}$ and (21)
k_P	10	–	Controller gain of proportional term
n	5	–	Number of joints
l	0.07	m	Segment length
ψ	$2\pi/n$	rad	Angular distance between adjacent joint location
d	0.01	m	Distant between center-line and joint location

VI. FRAMEWORK AND DEMONSTRATION IN SIMULATION

Now that the planner, trajectory generator, and controller have been defined by using Clarke coordinates, the framework can be assembled. Figure 6 illustrates the framework with additional encoders enabling the possibility to reuse components from existing frameworks.

The framework is parameterized by the boundaries of the sampler used as planner, the kinematic constraints of the trajectory generator, the controller gain of the controller scheme, as well as the kinematic design parameters for the encoder and decoder. Table I lists the parameters and their values. For the demonstration, the planner is the simple random joint generator providing feasible joint values as described in Sec. III. The used point-to-point trajectory generator is described in Sec. IV-B. For the controller, we use two independent P controllers with pre-compensation [15] acting on the difference between the measured and desired Clarke coordinates. The output of the controller is passed through a decoder to a displacement-actuated continuum robot in simulation.

For the system consisting of the framework and the coupled displacement-actuated continuum robot, the sampling time is set to $10\,\mathrm{ms}$. To simulate the displacement-actuated continuum robot, each actuator is modeled as an interdependent first-order proportional delay element (PT$_1$) system. The time constant for each PT$_1$ system is uniformly set to $200\,\mathrm{ms}$. We include additive noise, drawn from a
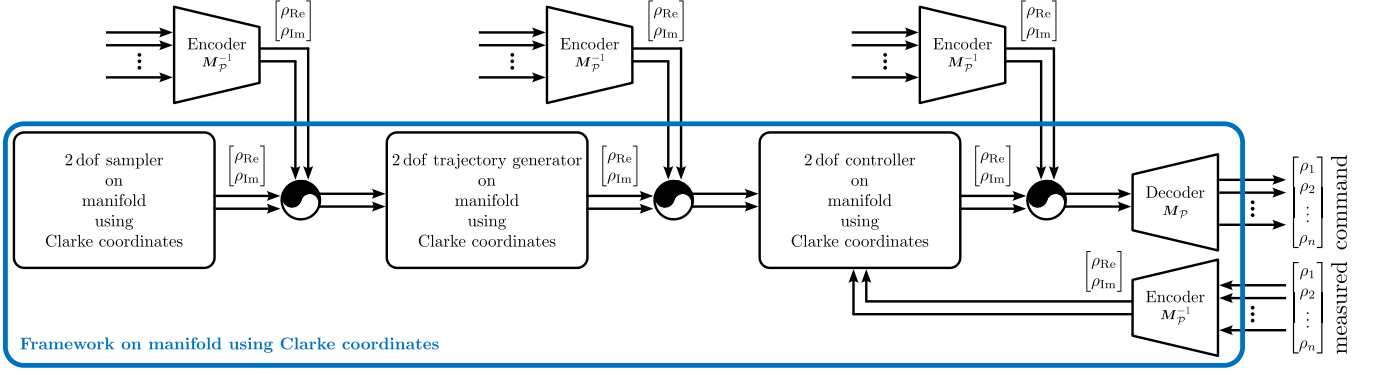
Fig. 6. Framework based on Clarke coordinates. Each of the components of the framework, *i.e.*, sampling, trajectory generation, and control, are solely defined in terms of the Clarke coordinates. Using an encoder as an interface, existing frameworks can be included and reused. A switch depicted as a circular symbol enables or disables external inputs. The right-hand side represents the interface with a displacement-actuated continuum robot.

uniform distribution with an amplitude of $0.1\,\text{mm}$, to the measure displacement-actuated joint values $\boldsymbol{\rho}_{\text{m}}$. The robot is mass-less as only the kinematic constant curvature model is considered for the visualization. To account for several segments, each segment is treated independently and has its own instance of the framework as depicted in Fig. 7, where all frameworks have the same parameters stated in Table I. To synchronize all trajectories across the frameworks, the maximum duration is computed and used.
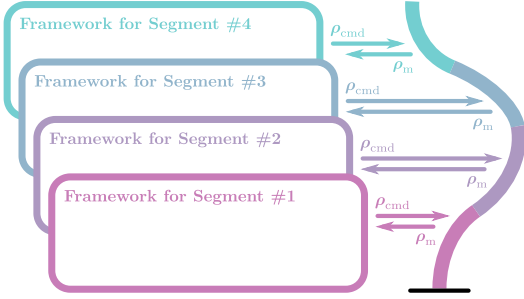

Fig. 7. One segment, one framework

Fig. 8. Planned trajectory using Clarke coordinates.

The parallelized framework illustrated in Fig. 7 provides displacements, which are sent to the displacement-actuated continuum robot. Figure 8 shows the Clarke coordinates and their time derivatives generated by the trajectory generator (18).

As can be seen from the image sequence in Fig. 10, a displacement-actuated continuum robot with four segments can move from a start configuration to a goal configuration. The course of the desired and executed trajectories are shown in Fig. 9.

VII. DISCUSSION AND FUTURE WORK

The general approach of the proposed framework is to construct components of the framework using Clarke coordinates. The components of this framework are a planner, a trajectory generator, and a controller. Each of them can be switched to a different component utilizing Clarke coordinates. Furthermore, using an encoder (5), the framework can bypass components to incorporate approaches that are not based on Clarke coordinates. Currently, the framework is limited to kinematics, the independence of each segment,
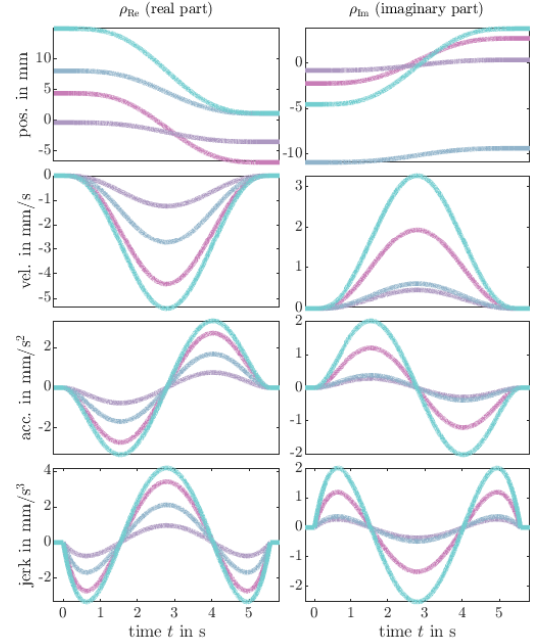
and the utilization of relatively simple components. This can be overcome by incorporating dynamic models and more sophisticated approaches for each component.

Furthermore, the only component with feedback is the controller. However, a large variety of different approaches with feedback from the robot and environment exist in the literature. For instance, collision avoidance for static and dynamic environments. For the former, the formulation of C-space obstacles [16] on the manifold is an interesting direction. For the latter, adapting potential fields [17] on the manifold is worth pursuing. Due to the low dimensionality of the $2\,\text{dof}$ manifold, it is expected to have the potential to be more computationally efficient.

In general, each of the components is exchangeable. As shown in Fig. 6, each component has input and output defined as Clarke coordinates. Moreover, due to the use of Clarke transform [5], it inherits the potential benefits of being computationally efficient, interpretable, closed-form, and compact in the formulation. One might say that the components are defined in their canonical formulation since
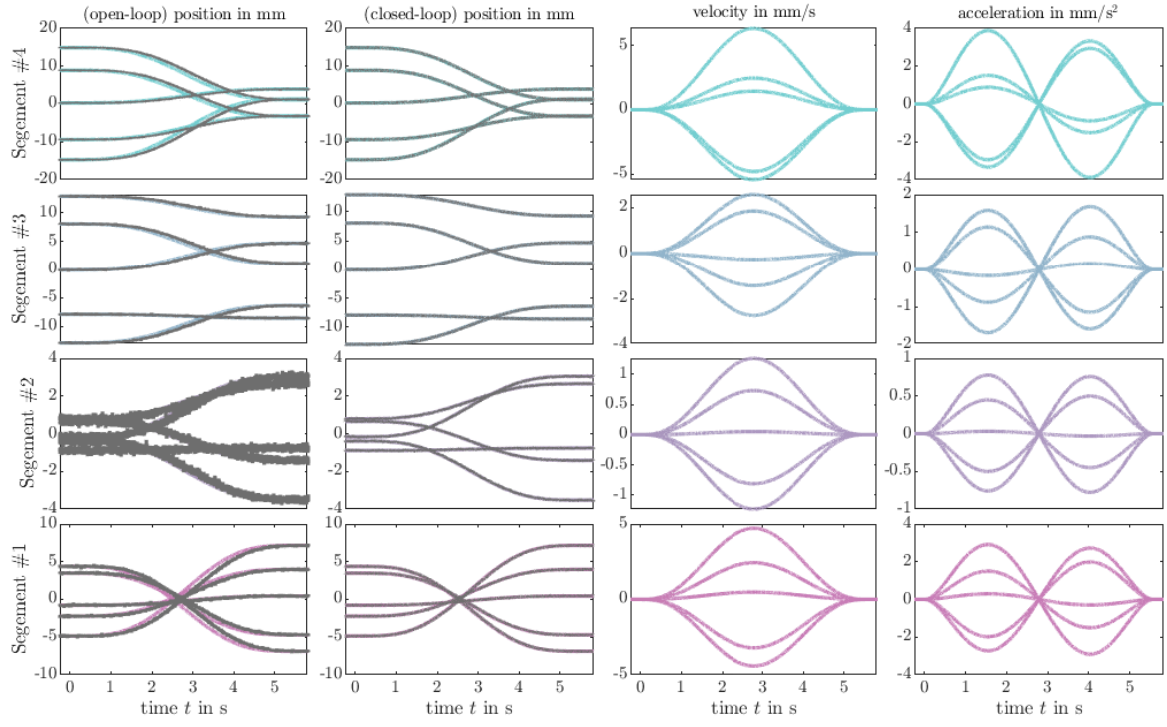
Fig. 9. Displacement-control. (1st column) Desired path *vs.* open-loop behavior of the PT$_1$ with noisy measurements. The noise is more pronounced for shorter displacements amplifying the noise-to-signal ratio, *cf.* segment #2 and segment #4. (2nd column) Desired path *vs.* closed-loop behavior. (3rd column) Velocity of the desired path. (4th column) Acceleration of the desired path that shows continuous acceleration profile, *i.e.*, $\mathcal{C}^3$ smoothness. Measured velocity and acceleration are not used due to the high noise. Observe that the kinematic constraints $v_{max} = 10\,\text{mm/s}$ and $a_{max} = 10\,\text{mm/s}^2$ are complied with.
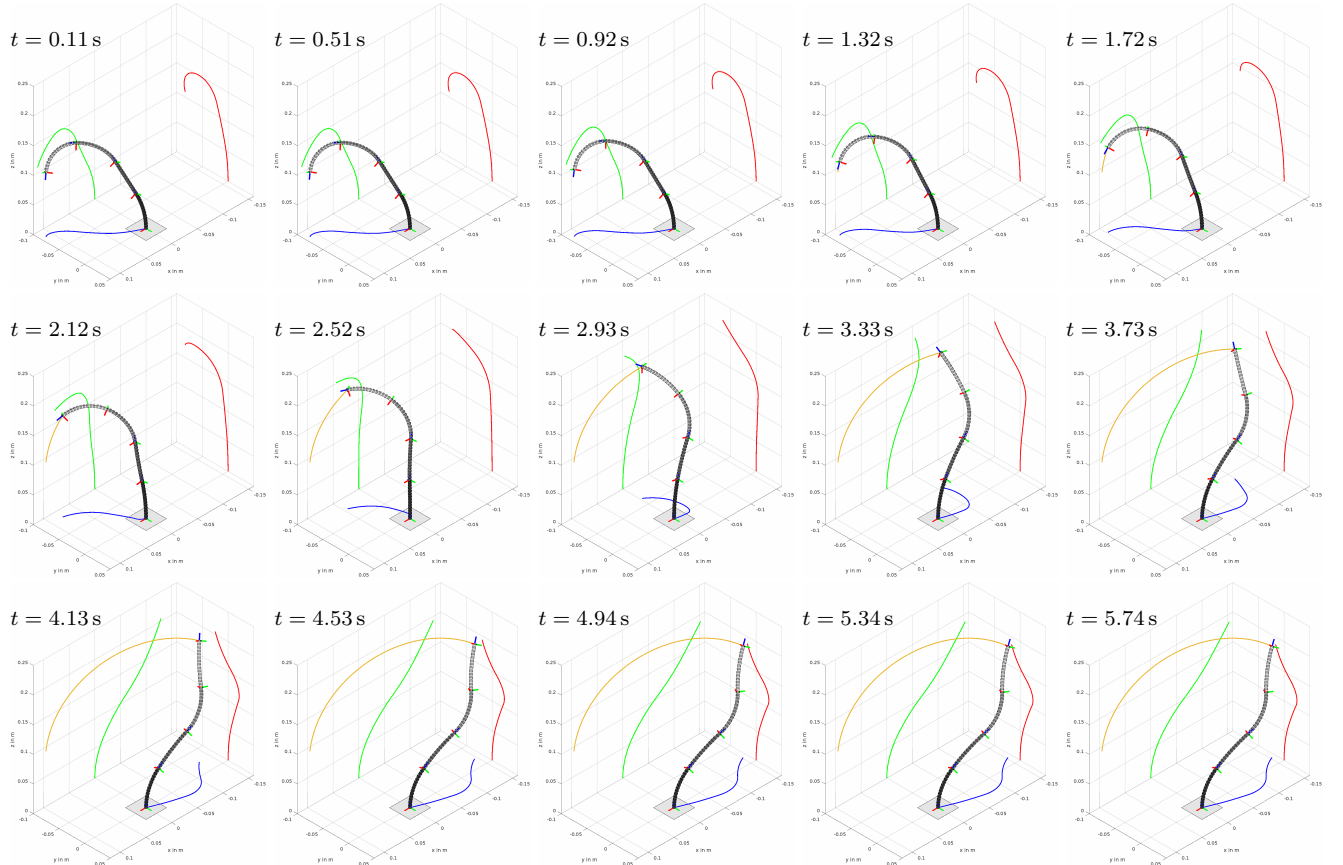

Fig. 10. Image sequence of a motion. The outputs of the controllers are fed to a displacement-actuated continuum robot with four segments. For the visualization, the CRVISTOOLKIT is used.

Clarke coordinates are the unification and generalization of improved state representations. However, the cost of time and effort might not justify the reimplementation of an existing framework. To reap the benefits of particular components, the encoder-decoder architecture [8] presented in Sec. II-D allows to include components based on Clarke transform into an existing framework not formulated with Clarke coordinates. Figure 5 shows this for the controller component. This is akin to using quaternions in certain parts of a bigger framework mainly relying on rotation matrices.

Regarding the generation of trajectories, loosely speaking, the trajectory generator (15) for the joint space is Clarke transformed onto the manifold to form the trajectory generator (18) for Clarke coordinates. While (15) is transformed using (7), the kinematic constraints for (18) are not simply obtained. Furthermore, the used point-to-point trajectory generation as defined for (15) and (18) are commonly used for independent positions in Task space or independent joint values of a serial kinematic rigid robot. Therefore, those constraints are not directly related to the joints. For future work, we will investigate the use of virtual displacement.

To show that the framework is not limited to constant curvature assumption [7] and to derive necessary relations improving the understanding of sampling and kinematic constraints, we delved into the virtual displacement [10], [5] and displacement constraint (3). We establish in Sec. II-C that the displacement constraint (3) is not a result of constant curvature. Further connections to parallel curves are made that reframe geometric insights by Simaan *et al.* [11] and Burgner-Kahrs *et al.* [18]. In the case of displacement-actuated continuum robots, the displacement constraint (3) and parallel curves imply that the centerline needs to be sufficiently smooth and the tip orientation ϕ is related to displacements. In contrast and in general, parallel curves for a piecewise linear curve akin to a serial-kinematic robot do not have this property, *i.e.*, (3).

More importantly, the geometric insights on parallel curves imply that no displacement, *i.e.*, $\rho = 0$, is detected if the tip orientation ϕ is the same as the base orientation, *i.e.*, $\phi = 0$. This is depicted in Fig. 3. This has implications for the design of high-level planners, controllers, and sensor placement. For instance, a proprioceptive sensor based on displacement-actuated joints might not register an interaction with the environment, *e.g.*, unwanted collation, or desired manipulation. However, non-constant joint location along the structure, *e.g.*, non-straight tendon routing, will create a non-zero displacement constraint, *cf.* (3), in general. To investigate this challenge, future work will include the extension to more general joint locations varying along the structure of a displacement-actuated continuum robot.

VIII. Conclusions

We lay out a principle to formulate a complete framework using Clarke coordinates. Besides benefiting from the unified formulation on the $2\,\mathrm{dof}$ manifold, this framework has two key features. First, each of the components is interfaced using the Clarke coordinates allowing for modularity that comes with the associated advantages. Second, an encoder acts as an interface to allow the integration of external components that are not formulated using Clarke coordinates. Those features allow both in-house and external developers from different labs to integrate, among others, task planners, control schemes, and machine learning algorithms. This can accelerate innovation and drive the adoption of components that are built for displacement-actuated continuum robots. We anticipate that this paradigm shift of creating frameworks enables knowledge transfer between research communities to accomplish complex tasks for real-world application with these robot systems.

References

[1] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1001–1008, 2020.

[2] T. F. Allen, L. Rupert, T. R. Duggan, G. Hein, and K. Albert, "Closed-form non-singular constant-curvature continuum manipulator kinematics," in *IEEE International Conference on Soft Robotics*, 2020, pp. 410–416.

[3] Y. Cao, F. Feng, Z. Liu, and L. Xie, "Closed-Loop Trajectory Tracking Control of a Cable-Driven Continuum Robot With Integrated Draw Tower Grating Sensor Feedback," *Journal of Mechanisms and Robotics*, vol. 14, no. 6, p. 061004, 2022.

[4] S. Dian, Y. Zhu, G. Xiang, C. Ma, J. Liu, and R. Guo, "A Novel Disturbance-Rejection Control Framework for Cable-Driven Continuum Robots With Improved State Parameterizations," *IEEE Access*, vol. 10, pp. 91 545–91 556, 2022.

[5] R. M. Grassmann, A. Senyk, and J. Burgner-Kahrs, "Clarke Transform – A Fundamental Tool for Continuum Robotics," *arXiv preprint arXiv:2409.16501*, 2024.

[6] P. E. Dupont, N. Simaan, H. Choset, and D. C. Rucker, "Continuum robots for medical interventions," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 847–870, 2022.

[7] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[8] R. M. Grassmann and J. Burgner-Kahrs, "Clarke transform and encoder-decoder architecture for arbitrary joints locations in displacement-actuated continuum robots," *arXiv preprint*, 2024.

[9] ——, "Clarke Transform and Clarke Coordinates – A New Kid on the Block for State Representation of Continuum Robots," in *40th Anniversary of the IEEE International Conference on Robotics and Automation*, 2024.

[10] M. Firdaus and M. Vadali, "Virtual Tendon-Based Inverse Kinematics of Tendon-Driven Flexible Continuum Manipulators," in *International Conference on Advances in Robotics*, 2023, pp. 1–5.

[11] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *International Conference on Robotics and Automation*, vol. 1. IEEE, 2004, pp. 351–357.

[12] B. Pham, "Offset curves and surfaces: a brief survey," *Computer-Aided Design*, vol. 24, no. 4, pp. 223–229, 1992.

[13] G. Loria and F. Schütte, *Spezielle algebraische und transscendente ebene Kurven: Theorie und Geschichte (German)*. BG Teubner, 1902, vol. 5.

[14] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[15] P. Morin and C. Samson, "Motion control of wheeled mobile robots," *Springer handbook of robotics*, vol. 1, pp. 799–826, 2008.

[16] T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers*, vol. 32, no. 2, pp. 108–120, 1983.

[17] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[18] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.