Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant

Matthias Rolf and Jochen J. Steil

Abstract—We evaluate the use of continuum kinematics with constant curvature as a kinematic model for Festo’s “Bionic Handling Assistant” (BHA). We introduce a new, elegant, and parameterless method to deal with geometric singularities in stretched positions, which allows to capture pure elongations that are not naturally expressed by the toroidal deformations underlying the constant curvature assumption. The stability of the method is shown with numeric simulations. We evaluate how well this model describes the BHA by using real-world position measurements as quantitative ground truth and find a good match between model and real BHA, with only 1% relative error. The model provides a practical, and highly efficient tool for the simulation and experimentation with continuum robots and is available as free software library.

I. INTRODUCTION

Continuum robotic systems have gathered increasing interest in the last decade of robotics research. Several robot platforms have been introduced that are inspired by biological actuators like elephant trunks [1], octopus arms [2], snakes [3] or even squid tentacles [4]. These systems move without traditional revolute or prismatic joints, but are based on continuous deformations in shape, and are typically actuated by either hydraulic or pneumatic actuation.

The focus of this paper is the Bionic Handling Assistant (BHA) [5] which is a new, award-winning [6] continuum platform inspired by elephant trunks and manufactured by Festo (see Fig. 1). The robot is pneumatically actuated and made almost completely out of polyamide which makes it very flexible and lightweight (ca. 1.8 kg). The robot comprises three main segments, each with three pneumatic bellow actuators, a ball-joint as wrist, also actuated by three actuators, and a three finger gripper actuated by one bellow actuator. The overall length without actuation is 0.75 m. When the bellow actuators are supplied with pressure, they extend their length and can cause arc-like deformations as well as elongations of the trunk. Supplying full pressure on the actuators extends the overall length to 1.2 m [7]. The posture can be sensed with twelve length sensors, one connected to each actuator except to the gripper. The nine main segment actuators are connected to cable potentiometers that allow to measure the length from the trunk base to the end of the actuator, such that the differences encode the length of each individual actuator.

M. Rolf and J. J. Steil are with Research Institute for Cognition and Robotics, Bielefeld University, Germany mrolf,jsteil@cor-lab.uni-bielefeld.de

1 The software library including kinematics computation and 3D visualization is available at http://www.cor-lab.org/software-continuum-kinematics-simulation
of measurements match. The most frequently used class of deformation functions are circular shapes [10], [12], [13]. Jones et al. used a virtual 5-link structure with Denavit-Hartenberg parameters to express the circular shape transformation by means of rigid-link equations [12]. The same authors presented a numerical investigation in [10] which points out the central problem of circular approaches: they do not apply if a segment is not bent, but in a straight position. This is particularly delicate if the continuum robot has variable length, meaning that it can articulate elongations without bending (such as the BHA). Godage et al. [14] proposed a method to prevent this problem by choosing a more general class of functions. Their model successfully describes elongation, but results in a rather high-dimensional parameter estimation problem.

Circular shapes correspond to a constant curvature of the continuum robot and describe the energy minimum without gravitational or other external forces [15]. Depending on the particular robot platform, these constraints can lead to severe miss-estimates of the robot kinematics, such as 50% relative error for the OctArm V platform [16]. This problem can be tackled by approaches that consider a more detailed model of material physics and bending processes [16], [17]. These models do, however, not provide closed-form solutions. They require an iterative solution of (potentially high-dimensional) differential equation systems, which makes them computationally expensive.

B. Approach

This paper investigates in how far the efficient approach of circular shapes can be used to model the Bionic Handling Assistant. Firstly, we provide a modification of the circular shape equations derived in [10], [12]. We improve over the state of the art by showing how the resulting coordinate transformations can be used to express elongations in Sec. III and show its numerical stability. Secondly, the main contribution of this paper is to show that circular shapes do indeed provide an accurate model for the BHA in Sec. IV. We provide quantitative results, comparing the model prediction to real world movement data recorded with a motion tracking system. While other studies showed substantial mismatches between constant curvature models and real continuum robots, we find that the BHA is modeled with an excellent accuracy of 1% relative error. Due to its computational efficiency, the model thereby proves to be a useful and practical tool for experimentation with the BHA.

II. VARIABLE LENGTH CONTINUUM KINEMATICS

In order to model the kinematic behavior of the BHA, we first consider isolated segments. Fig. 1 shows how the three actuators in each main segment cause a deformation between two rigid segment bases (shown in red). Lengths can be measured on the outer side of each actuator which gives three parameters for the reconstruction of pose per segment. In this section we derive how these three lengths can be used to estimate the coordinate transformation between two platforms, which can then be chained in order to get the complete forward kinematics from base to end effector.

A. Torus Segments as Geometric Model

As we have argued in the introduction, three length measurements are in general not sufficient to estimate a six dimensional coordinate transformation. We follow [10], [12] by using the additional constraint that deformations apply with constant curvature in each actuator. For each particular deformation in three dimensions, the segment can be seen as a torus segment, whereas lengths on the outer surface can be measured. Fig. 2a shows this relation with the overall torus in light green and the trunk segment in dark green. We assume that the base of the segment is centered in the x-y plane, and
that the segment itself has a radius $b$. Then, the deformation can be expressed by three geometric parameters: the radius $r$ (shown in red) of the torus deformation, and the angles (blue) $\theta$, expressing which part of the torus represents the segment, and $\phi$, expressing how the torus is oriented in the $x$-$y$ plane.

Using the lengths measurements $l_1$, $l_2$ and $l_3$ for the segment, the geometric parameters $r$, $\theta$ and $\phi$ can be reconstructed [10], [12]. We first introduce two intermediate sizes which correspond to mean length, and the irregularity between individual lengths:

\[
\bar{l} = \frac{l_1 + l_2 + l_3}{3} \tag{1}
\]

\[
g = \sqrt{l_1^2 + l_2^2 + l_3^2 - (l_1 l_2 + l_1 l_3 + l_2 l_3)} \tag{2}
\]

Then the geometric parameters can be computed by the following equations:

\[
\theta = \frac{2 \cdot g}{3 \cdot b} \tag{3}
\]

\[
r = \frac{3\bar{l} \cdot b}{2 \cdot g} = \frac{\bar{l}}{\theta} \tag{4}
\]

\[
\phi = \tan^{-1}\left(\sqrt{3} \cdot (l_3 - l_2) \over l_2 + l_3 - 2l_1\right) \tag{5}
\]

\[
= \text{atan2}\left(\sqrt{3} \cdot (l_3 - l_2), l_2 + l_3 - 2l_1\right) \tag{5}
\]

\[\hat{p}_0 = R_1^r \cdot \hat{p}_1 + d_0^l \tag{8}\]

The offset vector $d_0^l = (x, y, z)^T$ can be derived from the geometric parameters by simple trigonometry:

\[
x = 2r \cdot \sin\left(\frac{\theta}{2}\right) \cdot \cos\left(\frac{\pi - \theta}{2}\right) \cdot \sin\left(\phi - \frac{\pi}{2}\right) \tag{9}
\]

\[
y = -2r \cdot \sin\left(\frac{\theta}{2}\right) \cdot \cos\left(\frac{\pi - \theta}{2}\right) \cdot \cos\left(\phi - \frac{\pi}{2}\right) \tag{10}
\]

\[
z = 2r \cdot \sin\left(\frac{\theta}{2}\right) \cdot \sin\left(\frac{\pi - \theta}{2}\right) \tag{11}
\]

The rotation matrix can be found by considering an Euler-like composition of rotations. We first rotate around the $z$-axis by $-\phi$ to align the $x$-axis with the origin of $\phi$. In this pose, the $\theta$ rotation can be performed around the $y$-axis, after which the $-\phi$ is reversed:

\[
R_z^\phi = R_z(\phi) \cdot R_y(-\theta) \cdot R_z(-\phi), \tag{12}
\]

where $R_z(\alpha)$ is the rotation matrix expressing an angle $\alpha$ around the $i$-axis. The fully expanded form of this equation is shown in Eqn. 6.

### III. Elongation Case and Numerics

The weak point of the torus-segment formulation is how to express pure elongations, i.e. cases when all lengths vary with $l_1 = l_2 = l_3 = 1$. Pure elongation can not be modeled as torus deformation, so that limit cases need to be considered. We can observe the effect of pure elongation on the geometric parameters:

1) $\theta$ becomes zero.
2) $r$ becomes infinite.
3) $\phi$ is undefined.

At first sight the most severe problem is the computation of $r$, since it requires division by $\theta = 0$. $r$ only appears in the expressions for $d_0^l$, but not in $R_1^r$. In $d_0^l$ we can find the common expression $2r \cdot \sin(\theta/2)$ that connects all appearances of $r$. We can use the following identity:

\[
2r \cdot \sin\left(\frac{\theta}{2}\right) \equiv \bar{l} \cdot 2 \cdot \sin\left(\frac{\theta}{2}\right) = \frac{\bar{l}}{\theta} \cdot \frac{\theta}{2} \tag{4}
\]

This expression has a well-defined limit for $\theta = 0$ since

\[
\lim_{\alpha \to \pm0} \frac{\sin(\alpha)}{\alpha} = 1. \tag{13}
\]

The important question is when to insert this limit value. In the case of finite-precision machine arithmetics the decision can be delicate. A direct way to do this would be to test numerically for an exact $\alpha = 0$ match, but which is numerically problematic. Jones et al. [10] have designed a rather complex testing-mechanism to insert the limit case inside an in-equality bounded region with parametrized size around zero. Here, we introduce a more natural, parameter-less way: Since $|\sin(\alpha)| \leq |\alpha|$ (also guaranteed for numeric sin implementations that follow IEEE-754 (2008) standard recommendations [18]), we can distinguish three relevant numeric cases close to zero:

1) $\sin(\alpha) = \alpha = 0$: limit (13) needs to be inserted.
2) $\sin(\alpha) \neq 0 \land \alpha \neq 0$: the fraction $\sin(\alpha)/\alpha$ can be computed safely. Numeric division gives accurate results for arbitrarily small values.
3) $\sin(\alpha) = 0 \land \alpha \neq 0$: plain division results in zero which is a severe numeric error. Limit (13) needs to be inserted.

Hence, an exact and numerically safe test is to check for $\sin(\alpha) = 0$, requiring that $\alpha \in [-\pi; \pi]$ (i.e. that the sine is zero at the first pole, not at any other pole). This test includes all cases of $\alpha = 0$ and avoids the problem of the sine dropping to zero before $\alpha$ does. With this test we can define a replacement function for $\sin(\alpha)$:

\[
\sin X(\alpha) = \begin{cases} 
1 & \text{if } \sin(\alpha) = 0.0 \land |\alpha| < \pi \\
\sin(\alpha)/\alpha & \text{else}
\end{cases}
\]

This function is safe to evaluate numerically even close to zero since division is, in contrast to subtraction, a numerically safe operation. We can now insert it in $d_0^l = (x, y, z)^T$.
coordinate transformation expressions remain well-defined: evaluated.

point values. This increase of accuracy largely stabilizes the

(2) and (5)) which were evaluated with 96 bit floating

compiled with GCC 4.3. For all results we used 64 bit

program running on an Intel processor set,

the singular value of

shows how geometric model and coordinate transformation

Hence, the replacement of \( \sin X \)

and get

\[
\begin{align*}
x &= 7 \cdot \sin X \left( \frac{\theta}{2} \right) \cdot \cos \left( \frac{\pi - \theta}{2} \right) \cdot \sin \left( \phi - \frac{\pi}{2} \right) \quad (14) \\
y &= -7 \cdot \sin X \left( \frac{\theta}{2} \right) \cdot \cos \left( \frac{\pi - \theta}{2} \right) \cdot \cos \left( \phi - \frac{\pi}{2} \right) \quad (15) \\
z &= 7 \cdot \sin X \left( \frac{\theta}{2} \right) \cdot \sin \left( \frac{\pi - \theta}{2} \right). \quad (16)
\end{align*}
\]

The second problem is that \( \phi \) is undefined in pure elongation cases. This problem is, however, very mild compared to the singular value of \( r \). Since \( \phi \) is only undefined if \( \theta = 0 \), all coordinate transformation expressions remain well-defined:

1) \( z \) does not depend on \( \phi \).
2) \( x \) and \( y \) become zero, since \( \cos \left( \frac{\pi - \theta}{2} \right) = 0 \).
3) The rotation matrix evaluates to identity (see Eqn. 7).

Hence, the replacement of \( \sin X \) is sufficient to deal with the singularity of the geometric torus-segment model. Fig. 3 shows how geometric model and coordinate transformation evaluate for several segment length combinations including singular pure elongation cases.

A. Numeric Stability

After fixing the singularity in the geometric representation, an obvious question is the numeric stability using finite precision machine arithmetics. We evaluated the equations with a C++ program running on an Intel processor set, compiled with GCC 4.3. For all results we used 64 bit floating point values except intermediate values that contain sums or differences of more than two values (Eqn. (1), (2) and (5)) which were evaluated with 96 bit floating point values. This increase of accuracy largely stabilizes the

\[
\begin{align*}
R_d &= \begin{pmatrix}
c(\phi)c(\theta)c(\phi) + s(\phi)s(\phi) & c(\phi)c(\theta)s(\phi) - s(\phi)c(\phi) & -c(\phi)s(\theta) \\
s(\phi)c(\theta)c(\phi) - c(\phi)s(\phi) & s(\phi)c(\theta)s(\phi) + c(\phi)c(\phi) & -s(\phi)s(\theta) \\
s(\theta)c(\phi) & s(\theta)s(\phi) & c(\theta)
\end{pmatrix} \\
\lim_{\theta \to 0} R_d &= \begin{pmatrix}
c(\phi)c(\phi) + s(\phi)s(\phi) & c(\phi)s(\phi) - s(\phi)c(\phi) & 0 \\
s(\phi)c(\phi) - c(\phi)s(\phi) & s(\phi)c(\phi) + c(\phi)c(\phi) & 0 \\
0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix} \quad (7)
\end{align*}
\]
otherwise delicate subtraction of values with similar size, in particular in Eqn. (2).

Fig. 4 shows results on the stability. We evaluated a segment with radius $b = 0.1\,m$ and fixated $l_1 = l_2 = 0.2\,m$. $l_3$ is varied around $0.2\,m$ with variable offset $l_3 = 0.2\,m \pm \Delta$, which corresponds to variations of the posture shown in Fig. 3a. For each value of $l_3$ we measured how far the coordinate transformation values in $d^l_3$ and $R^l_3$ deviated from their value at $l_3 = 0.2\,m$. For orientation, Fig. 4a shows how the values change on a rather global scale. We find that the overall results are very stable. In the range of $\Delta \sim 10^{-9}\,m$ (Fig. 4b) we can observe mild numeric irritations as the values begin to fluctuate. However, the fluctuations remain small in a range of $10^{-10}\,m$ which is by far better than the full degeneration of values reported in [10]. In a range of $\Delta \sim 10^{-10}\,m$ around the singularity, most values become constant. The only exception is the value of $z$, as shown for a range of $\Delta \sim 10^{-15}\,m$ in Fig. 4c. Even in this range, very close to the machine precision ($\sim 10^{-16}$ for 64 bit floats) the value of $z$ is extremely stable and even accurate which demonstrates the validity of the $\sin x$ replacement. In fact, the steps in the plot are isolated machine-epsilons. $z$ is the only value that is not affected by $\phi$, which allows to localize the minor fluctuations of all other transformation values in the computation of $\phi$.

The overall results show that the computation of the coordinate transformation can be done very stably even close to the singular pure-elongation case. Only minor fluctuations with an amplitude of $10^{-10}$ are observed which is clearly good enough for a practical kinematic simulation.

IV. Simulation of the Bionic Handling Assistant

In order to get a kinematic model of the Bionic Handling Assistant, we can now use the coordinate transformation and stack three segments on each other. The only free parameter of each segment is the base radius $b$. The geometric model as derived in the previous sections assumes that this radius is equal along one segment. This assumption does not hold exactly for the BHA, which has a slightly conic form. Therefore, we estimate those segment radii that give the best fit on observed data. After the third segment we introduce an offset of $0.148m$ in $z$-direction which corresponds to a position inside the gripper-palm.

We recorded ground-truth data from three different movements. During each movement, we drive the robot on a triangular path through the configuration space. Each segment starts with $l = (0.2, 0.2, 0.2)^T$, a straight posture at the lower end of the actuation range. The robot then elongates into $l = (0.3, 0.3, 0.3)^T$, also for each segment. A bending motion then goes into $l = (0.28, 0.2, 0.2)^T$ ($l_3$ stays elongated) in the first movement, whereas in the second and third movement $l_2$ and $l_3$ stay elongated. Afterwards the movement goes back to $l = (0.2, 0.2, 0.2)^T$ again. Thereby all segments bend in the same direction as shown in the first three images in Fig. 6. The three different movements then contain bendings in different directions. The reason for choosing this movement type with synchronous bending of all segments for our evaluation is twofold: Firstly, it allows to uncover cumulative modeling errors across the three segments, while the errors for different segments might compensate each other when moving along different axes. Secondly, they invoke a very strong overall bending of the robot, in which it is most strongly exposed to gravitational forces orthogonal to the robot’s main axis. Hence, prediction errors due to unmodeled gravitational shearing can be assessed.

During the robot’s movements, we recorded

1) the actual lengths $\vec{l} \in \mathbb{R}^3$ of all main segments, and
2) the ground-truth effector position of the BHA $x^* \in \mathbb{R}^3$,
are shown in Fig. 5. The three different movements are shown from different perspectives, showing the ground-truth BHA movement and the estimate of the kinematic model. Both are very close, with an average distance of 0.0102 m. Related to an approximate (average) robot length of 1 m this corresponds to a relative error of 1%. Note that the erratic form of the trajectories on top of the movement (particularly visible in the x/y view) does not reflect modeling errors, but a physically erratic movement during the data recording. The physical movement has been executed very slowly during the experiment in order to obtain enough data to average out sensory noise. The irritations in the movement are caused by the physically unstable mechanics in the stretched inverted pendulum position.

There are more systematic errors when the BHA bends downwards (low z value). Obviously, gravity starts to break the circular shape assumption in this region. It is, however, noticeable that not all three movements have equally strong deviations, which indicates that at least a portion of the deviation is caused by remaining calibration errors of the length-sensors. The calibration is indeed a delicate process and due to the morphology of the BHA, calibration (or generally sensing) errors of 1 mm in a length sensor can already account for end effector deviations of 1 cm.

V. Discussion

We have shown a formulation of continuum kinematics that can deal with circular bendings and elongations. Numerical evaluation shows that it gives stable results even close to the singularity, in which the geometric model of torus segments does not apply. Thereby our formulation improves the original one in [10], [12] by dealing with the singularity in a simple, elegant, and parameterless manner.

We have applied this formulation to the Bionic Handling Assistant and showed a good match of the simulated kinematics with respect to ground truth motion data. Obviously, the model can not perfectly describe the continuous robot behavior. Neither the assumption of circular shapes, nor the assumption of equal radius within segments hold exactly on the real robot. However, it is noteworthy to see that the model reaches 1% relative error, while constant curvature models have dramatically failed for other robots, and even expensive, “geometrically exact” models have only reached 1.5-5% relative error [16], [17]. Arguable reasons for this good fit are that (i) the BHA has a rather low weight which reduces the impact of self-gravitational forces and (ii) that the segments are rather compact, and therefore mechanically stable, as opposed to thin and long cannula robots investigated in [17].

Our experience shows that the controllable accuracy of the BHA lies in the same range as the prediction accuracy of the presented model. Substantial delays and noise, nonstationarities and physically slow equilibration processes do typically not allow for a reliable positioning of the end effector with less than 1 cm error. For practical purposes it would be pointless to use a computationally expensive model for prediction if it is “more accurate” than the physical robot can be. While we do not neglect the importance of measured with a Vicon motion tracking system [19]. In order to eliminate the impact of sensory noise (in particular in the length sensing), we averaged all values over 20 successive measurements. On this data set we assessed the accuracy of the kinematic model by evaluating the euclidean distance of the simulated kinematics and the measured effector position. We performed a grid-search over the segment radii to determine the best fit. The best parameters were found at $b_1^{(1)} = 0.1$ for the first segment, $b_1^{(2)} = 0.093$ for the second segment, and $b_1^{(3)} = 0.079$ for the third segment, which lies in a plausible range with respect to the physical geometry of the BHA. Results for this configuration
more physically grounded models for continuum robotics in general, it appears that the constant curvature approach is ideal for the BHA. The prediction accuracy is good enough for practical purposes and the model is very fast due to its closed form solution. We measured that the full BHA kinematics with three segments can be evaluated with 47.900Hz on a single Intel Q9550 core, which is more than fast enough for real-time computation and allows to perform otherwise infeasible offline calculations as they are for instance necessary in planning problems. In particular it is useful for online 3D visualization during experimentation with the robot, which allows to display various side information in relation to the 3D shape of the robot.

This paper did not directly address the kinematic control of the Bionic Handling Assistant. Jacobian controllers can, however, be derived from this formulation correspondingly to other formulations in literature [12], [13]. An attractive issue for future work is also to incorporate learning [20], [21] into simulation and control, in order to avoid the remaining, inherent mismatches between model assumptions and real world continuum robot.

REFERENCES