Notes for Lecture 3-17

Toby Baker, Howard Zhang

March 18, 2020

1 Introduction

Soft Robotics is the field of robotics that deals with the use of elastic elements that attempt to mimic organic materials.

1.1 Difficulty of Soft Robotic Control

Soft robots are difficult to design controllers for because they have infinite dimensions. *Example:* If given a soft robotic arm, we can split it up into two rigid bodies, and then subsequently split up each of those into more rigid bodies. Take this process to an infinite number of rigid bodies, and then you have an accurate model for a soft robotic arm. This makes dynamics/kinematics very computationally difficult for a soft robot.

2 Theory

2.1 Piecewise Constant Curvature Model

The Piecewise Constant Curvature(PCC) model reduces the kinematics of soft robotics to many segments with constant curvature that are merged together to create a globally differentiable curve.

2.1.1 Kinematics of PCC

As seen in Figure 1, each 2-dimensional segment has its configuration S_i, and the homogeneous transformation that maps between each segment is T_i^{i+1}.

From Figure 2, we see how each transformation matrix T_i^{i+1} is constructed with the value q, which is the degree of curvature. The equation of for each T is as follows:

$$T_{i-1}^{i}(q_i) = \begin{bmatrix} cos(q_i) & -sin(q_i) & L_i\frac{sin(q_i)}{q_i} \\ sin(q_i) & cos(q_i) & L_i\frac{1-cos(q_i)}{q_i} \\ 0 & 0 & 1 \end{bmatrix}$$

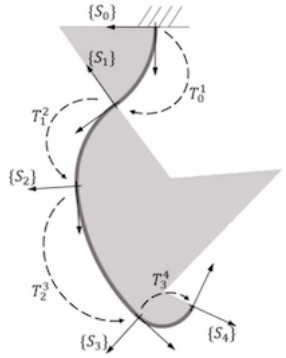


Figure 1: PCC Model with 4 Segments

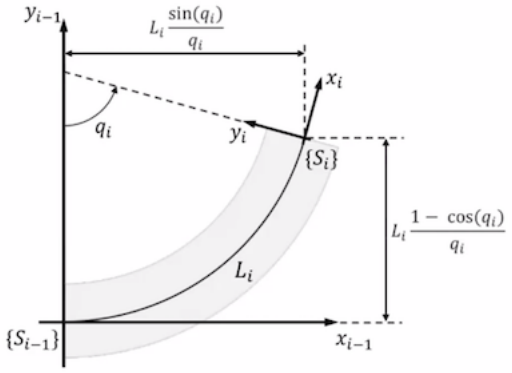


Figure 2: Kinematic Model of a Single Segment

2.2 Dynamics

2.2.1 Augmented Kinematics Mapping

We use the matrix mapping $m(q) : \mathbf{R}^n \rightarrow \mathbf{R}^{nm}$, which maps from a PCC modeled soft robot to a rigid body formulation. One of the more common mappings used by this paper is the revolute-prismatic-prismatic-revolute joint setup which preserves the center of mass to be in the middle of the segment shown in Figure 3. The columns of this mapping are as stated:

$$m_i(q_i) = \begin{bmatrix} \frac{q_i}{2} \\ L_i \frac{sin(\frac{q_i}{2})}{q_i} \\ L_i \frac{sin(\frac{q_i}{2})}{q_i} \\ \frac{q_i}{2} \end{bmatrix}$$

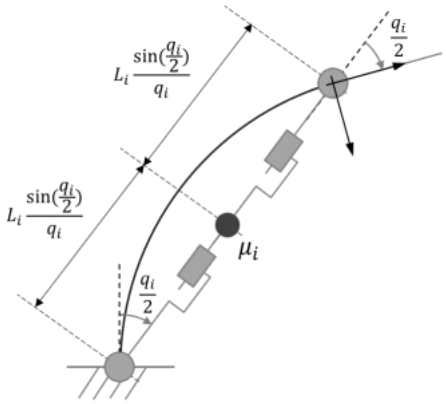


Figure 3: PCC Segment Modeled by 4 Joints

2.2.2 Dynamics with Augmented Kinematics

The initial dynamics mapping with the soft robotic PCC model is shown by Figure 4, where B, C, and G are the inertia, coriolis, and gravity values respectively, with torque and external forces on the right side of the equation. We plug in values for ξ (the configuration of the PCC segments) with the $m(q)$ mapping to define the equation in terms of q instead. We use the Jacobian (the derivative of the $m(q)$ mapping) and the mapping itself from Figure 5, add damping and stiffness terms D and K to reach Figure 6, our final dynamics model showing the change in q, the degree of curvature.

$$B_\xi(\xi)\ddot{\xi} + C_\xi(\xi,\dot{\xi})\dot{\xi} + G_\xi(\xi) = \tau_\xi + J_\xi^T(\xi) f_{\text{ext}}$$

Figure 4: Dynamics Equation 1

$$\begin{cases} \xi &= m(q) \\ \dot{\xi} &= J_{\text{m}}(q)\dot{q} \\ \ddot{\xi} &= \dot{J}_{\text{m}}(q,\dot{q})\dot{q} + J_{\text{m}}(q)\ddot{q} \end{cases}$$

Figure 5: Dynamics Equation 2

$$B\ddot{q} + (C + D)\dot{q} + G_{\mathrm{G}} + K\,q = \tau + J^T f_{\mathrm{ext}}$$

Figure 6: Dynamics Equation 3

2.3 Curvature Control

Curvature Control uses the equation from Figure 7 to control the torques. The I term is an integral gain term. Notice that after plugging this back into the dynamics equation derived earlier, we get a closed-loop control law that is PID, as seen in Figure 8. The stiffness term is the proportional gain, the damping term is the derivative gain, and the integral term is as before. The integral term is the only value that needs to be tuned in this case, and is responsible for lessening the global error from the inaccuracy of the model used. This closed loop control law helps the soft robot maintain its inherent softness when interacting with the surrounding environment.

$$\tau = K\bar{q} + D\dot{\bar{q}} + C(q,\dot{q})\dot{\bar{q}} + B(q)\ddot{\bar{q}} + G_{\mathrm{G}}(q) + I_{\mathrm{q}} \int (\bar{q} - q)$$

Figure 7: Curvature Control Equation

$$\begin{aligned} B(q)(\ddot{q} - \ddot{\bar{q}}) + C(q,\dot{q})(\dot{q} - \dot{\bar{q}}) - J^T(q) f_{\mathrm{ext}} \\ = K(\bar{q} - q) + I_{\mathrm{q}} \int (\bar{q} - q) + D(\dot{\bar{q}} - \dot{q}) \end{aligned}$$

Figure 8: Closed-Loop Curvature Control Equation

2.4 Cartesian Impedance Control

Cartesian impedance control is used to have the end-effector follow along the surface of an obstacle, while remaining compliant. Regulating the impedance at the contact is extremely important for robust and reliable interactions between the robot and its' environment.

In Figure 9 we can see all necessary terms that are later used for control. $n_{\perp}$ is the perpendicular vector to the surface tangent and $n_{\parallel}$ is the vector tangent to the surface. An artificial spring damper (K_c, D_c) system is used to to move the soft arm toward the goal, in a way that is compliant.

Throughout we assume:

- We know the point x_o which is the initial contact point on the target surface.

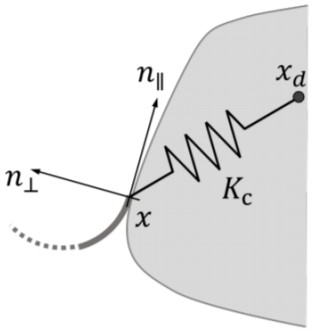


Figure 9: Diagram of the control model

- That we can detect when the arm is in contact with the obstacle, which can be done with force sensors or a motion capture rig.
- The target point x_t, (x_d in Figure 9 is on the surface.

Using the above assumptions we can leverage the dynamics to generate a control law as in Equation 1. The Jacobian is used to map the cartesian control terms into the jointspace of the soft robot. The first term in the equation is used to simulate the spring damper system and the integral term projects the error into the $n_{\parallel}$ direction and reduces the steady state error due to approximations made in the modelling. The remaining terms are the dynamics of the system. Throughout q, $\dot{q}$ are referencing the curvature and its derivative.

$$\tau = J^T(q)(K_c(x_d - x) - D_c J(q)\dot{q}) + C(q,\dot{q})\dot{q} + G_G(q) + Kq + I_c J^T(q) n \parallel \int n_{\parallel}^T (x_d - x) \tag{1}$$

In practice a δ term is used to ensure contact between the surface and the end-effector in the direction of $-n_{\perp}$. The algorithm used can be seen in Figure 10. The first section approaches the surface to reach the point x_0, during this time the integral term is not used. Following this we track along the surface to our target position and the integral term is used.

3 Results

The arm used during experiments has 5 bi-directional segments (can curve both ways) and is supported by roller balls to move in a frictionless level plane. A motion capture rig is used to provide real-time measurements and feedback to the system.

Algorithm 1 High level control

while isInContact() == False **do** ▷ Approaching
    $n_{\parallel} \leftarrow [0 \;\; 0]^{\mathrm{T}}$
    $n_{\perp} \leftarrow [0 \;\; 0]^{\mathrm{T}}$
    $x_{\mathrm{d}} \leftarrow x_0$
end while
while $||x - x_{\mathrm{d}}||_{n_{\parallel} n_{\parallel}^T} > \epsilon$ **do** ▷ Exploring
    $n_{\parallel} \leftarrow$ readParallelDirection()
    $n_{\perp} \leftarrow$ readPerpendicularDirection()
    $x_{\mathrm{d}} \leftarrow x_{\mathrm{t}} - n_{\perp}\delta$
end while

Figure 10: Algorithm used for Cartesian Impedance Control

Before tests began several parameters relating to the dynamics were identified by passed pulse-train inputs to the pneumatic actuators and observing how the system behaved.

3.1 Curvature Control

The input to the pistons is generated by first computing the torque and then passing it through a second order linear filter and the desired curvature trajectory can be seen in Equation 2. In Figure 11 we can see the reference trajectory as well as the trajectory of the curvature of each segment. In Figure 12 we see the same plots but the control was done with an integral gain which reduced the l_2 norm of the error from 0.1311 *rad* to 0.935 *rad* but does modify the torque profile over the evolution.

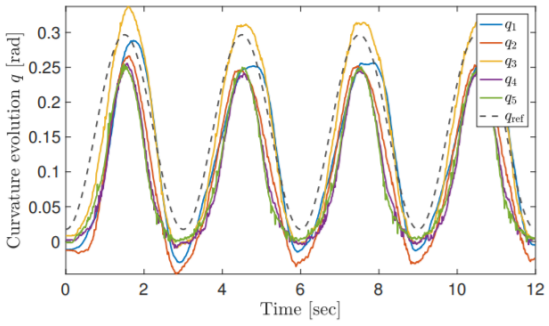
$$\bar{q}_i(t) = \frac{\pi}{20} - \frac{\pi}{24} \cos \frac{2}{3} \pi t \tag{2}$$

3.2 Cartesian Impedance Control

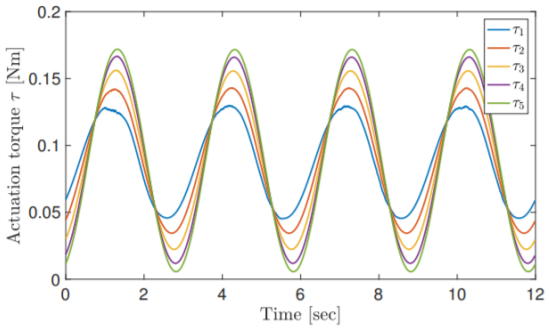
The input to the pistons is the output of a second order filter which takes the computed torque as an input. The control was tested in a scenario similar to that described in Algorithm 1 where the end effector tracks the surface of an obstacle. The cartesian states are assumed to be decoupled so diagonal gain matrices are used. The trajectory that was tracked can be seen in Figure 13.

First the arm is commanded to reach the point x_o on the surface of the obstacle and once contact is made, moves along the surface. In Figure 14 we can see the applied torques and desired x, y positions of the end effector.

Although both of these examples are planar the authors state that both of these control algorithms can be generalized to work with systems in higher dimensional spaces.

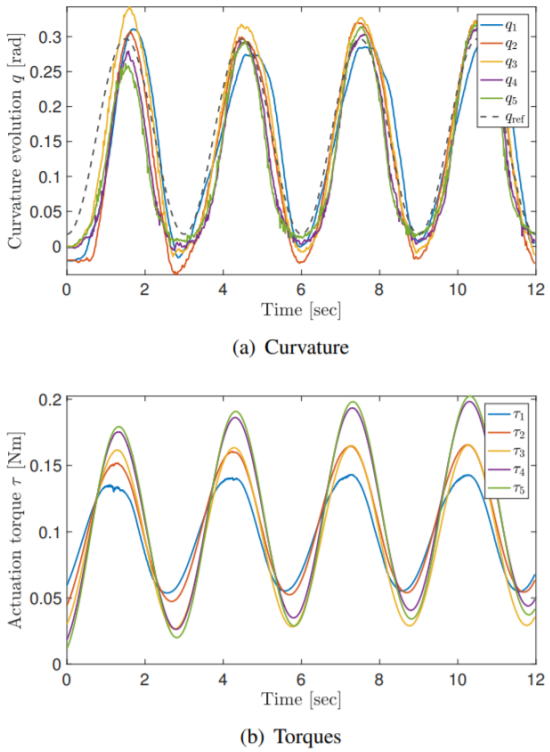


(a) Curvature



(b) Torques

Figure 11: Evolution of the curvature and input torque without integral gain present



(a) Curvature

(b) Torques

Figure 12: Evolution of the curvature and input torque with integral gain present

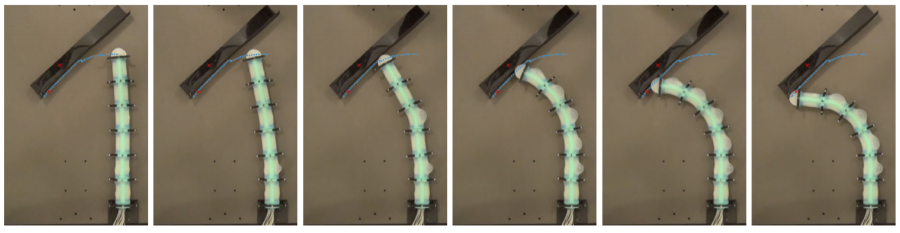


Figure 13: Algorithm used for Cartesian Impedance Control

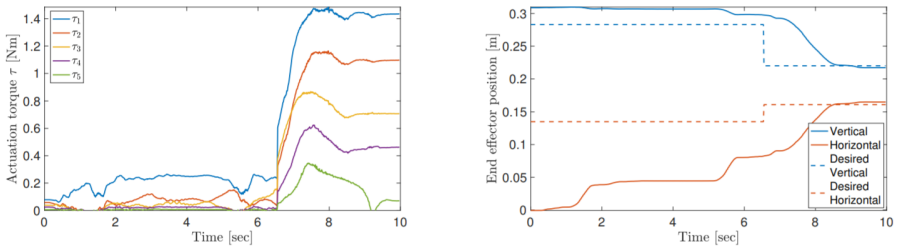


Figure 14: Algorithm used for Cartesian Impedance Control