EECS 206B Lab 4 Report: Soft Robotics Modeling

Aakash Parikh, Federik Warburg, Jun Zeng

May 2019

1 Introduction

In this project, we perform system identification of a soft robotics finger. We will firstly model the finger using the standard model, then using an hyper elastic model. Finally we will compare the Ecoflex and Dragonskin 30 fingers.

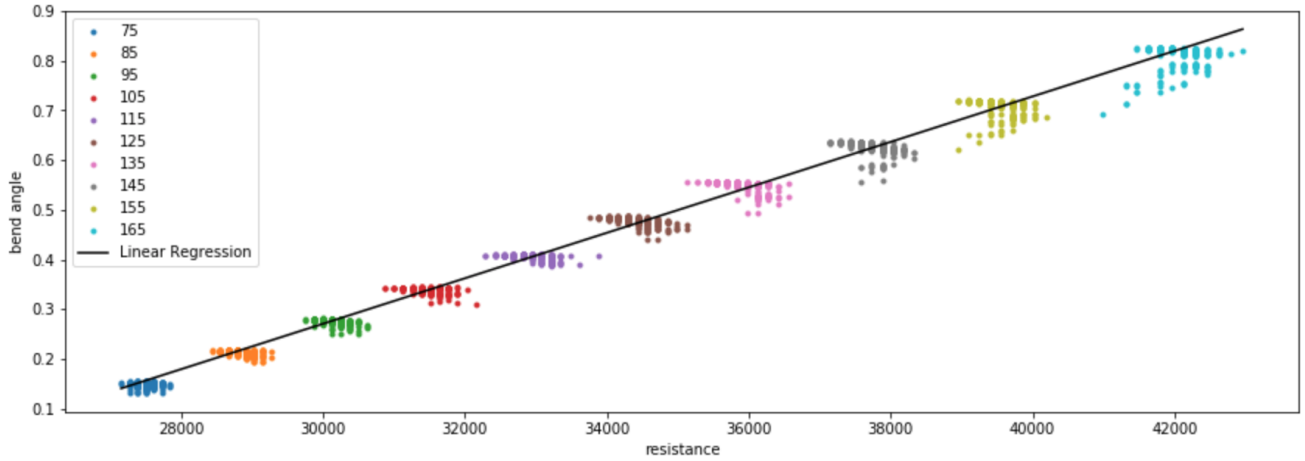
A video of us recording data can be found here. We include our GitHub repository here. Please note that initial data cleaning is done in python in **readcsv.ipynb** while the remaining system identification is performed in Matlab.

2 Methods and Results

2.1 Calibration of Sensor

The soft robot finger is equipped with a flex sensor. We receive voltage readings from the bend sensor. In order to calibrate this sensor, we want to compare it to the bend angle calculated by our camera measurements. The circuit for the bend sensor is a simple voltage divider across $5V$ with another $47k\Omega$ resistor. Thus the resistance across the sensor can be calculated as: $R = -47000 * (V-5)/V$ From the Rus paper, the bend angle can be calculated as: $q = 2 * tan^{-}1(\delta y/\delta x)$

We perform a linear regression on voltage and sensor readings from results of 10 different PWM inputs to calculate $q = a * R_{sensor} + b$. The least squares result is: $a = 4.574e^{-5}, b = -1.101$ with an r^2 score of 0.992. Note that we measure resistance in $k\Omega$, which is why the coefficient is so small.




```
intercept =  [-1.10122715] coef =  [4.57358337e-05] r^2 score =  0.9921836245365793
```

Figure 1: Resistance vs Bend Angle

2.2 Standard Model EcoFlex 30

First we identify the system as a standard rigid manipulator. This models the system as: $M(q)\ddot{q} + (C(q,\dot{q}+D)\dot{q} + G(q) + Kq = B\tau$. Ihis requires us to solve for system parameters: K, α, γ, D

2.2.1 Alpha and K analysis

We can derive initial estimates of α and K by using the steady state values. We use data from the time period where q_i and u is stabilized. This means we assume that $\dot{q}_i, \ddot{q}_i, \dot{\tau}, t\ddot{a}u = 0$. Thus we can use the simplified model: $G(q) + Kq = \alpha u$ This regression is performed in Matlab, using a **lsqlin** which allows us to constrain α and K values to be non negative. The psuedocode is as follows:

Result: Estimate α, K
$q = 2 * tan^{-}1(\delta y/\delta x)$
u = data.left_pwm
G = G_gen(q)
$C = (u, -q)$
$A = (-1, -1)$
$b = -0.01$
$((\alpha, K), resnorm)$ = lsqlin(C,d, A, b);

From our regression we get the following values: $\alpha = 3.4977e^{-4}, K = 0.0997, resnorm = 0.0017$
They will be used as initial values to help derive the remaining parameters.

2.2.2 Pressure and γ analysis

Additionally, we can model pressure to be proportional to our torque, τ. That is $p = \beta * \tau$. Since τ is modeled as a second order filter, we can solve the differential equation as: $\gamma^2 \ddot{\tau} + 2\gamma\dot{\tau} = \alpha u$. For this we approximate $\dot{\tau}(t_i) = (\tau(t_{i+1}) - \tau(t_{i+1}))/(t_{i+1} - t_{i-1})$ and $\ddot{\tau}(t_i) = (\dot{\tau}(t_{i+1}) - \dot{\tau}(t_{i+1}))/(t_{i+1} - t_{i-1})$ In contrast to part 1, we use a non linear solver **lsqnonlin** to help solve for γ. The psuedocode is as follows:

Result: Estimate γ, K
gammacost = @(x) gamma_cost_function$(x(1), x(2), P, \dot{P}, \ddot{P}, u)$
((gamma, beta) ,resnorm) = lsqnonlin(gammacost, (1, 1))

Result: gamma_cost_function$(\gamma, \beta, P, \dot{P}, \ddot{P}, u)$
error = $\beta * u - (\gamma^2 * \ddot{P} + 2 * \gamma * 2 * \dot{P} + P)$

We get results: $\gamma = -0.0022, \beta = 0.8876, resnorm = 1.3378e^4$

2.2.3 Solution for remaining parameters

Using the above solutions helps us find starting points that will help us solve the overall optimization problem. We are trying to minimize our actual q, versus the $\hat{q}$ that minimizes the error in $M(q)\ddot{q} + (C(q, \dot{q} + D)\dot{q} + G(q) + Kq = B\tau$.

Result: Estimate parameters
cost = @(x) cost_function$(x(1), x(2), x(3), x(4), t, u, q, q_0, \dot{q_0}, \tau_0, \dot{\tau_0})$
$((K, D, \alpha, \gamma)$, resnorm) = lsqnonlin(cost, $(K, \delta, \alpha, \gamma)$)

Result: cost_function$(K, D, alpha, gamma, t, u, q, q_0, \dot{q_0}, \tau_0, \dot{\tau_0})$
tau = find_tau$(u, t, \alpha, \gamma, \tau_0, ta\dot{u}_0)$
$\hat{q}$ = find_q$(\tau, t, K, D, q_0, \dot{q_0})$
error = $\hat{q} - q$
We get results: $k = 12.8994, D = 16.277, \alpha = 0.1643, \gamma = 0.0014, resnorm = 0.4913$

2.3 Hyper Elastic Model

Next, we note that rubber is actually considered to be hyper elastic, so we must use a fourth order stiffness model. Here $K = \sigma/\lambda$ where $\lambda = q/\sin(q)$ and $\sigma(\lambda) = ((\lambda^4 - 1)/\lambda^3)(2C_1 + 4C_2(\lambda - 1/\lambda)^2$ This requires us to calculate new parameters: C_1, C_2. We do this using a nonlinear solver as before.

Result: Estimate C_1, C_2
cost = @(x) hyper_elastic_C_cost$(x(1), x(2), \alpha, u, G, q)$ $((C_1, C_2)$, resnorm) = lsqnonlin(cost, (0, 0))

Result: hyper_elastic_C_cost$(C_1, C_2, \alpha, u, G, q)$
$\lambda = q/\sin(q)$
$\sigma = ((\lambda^4 - 1)/\lambda^3)(2C_1 + 4C_2(\lambda - 1/\lambda)^2$
$K = \sigma/\lambda$
error = $\alpha * u - Kq - G$

The remainder of the problem is the same as above, with the exception that **find_q** uses a new dynamics function involving C_1, C_2. However, we are unable to find initial conditions that allow the objective function to continue.

2.4 DragonSkin 30 vs EcoFlex 30 Comparison

This lab had two soft robotic fingers made from different materials. The discussion above was for the Ecoflex material. We have also conducted the analysis for the DragonSkin, and present the comparison here.

2.4.1 Steady State Comparison

Parameter	Ecoflex30	Dragonskin30
α	$3.4977e^{-4}$	$5.442e^{-4}$
K	0.0997	0.0995
resnorm	0.0017	$5.562e^{-5}$

2.4.2 Using pressure to find γ

Parameter	Ecoflex30	Dragonskin30
γ	-0.0022	0.0241
β	0.8876	1.147
resnorm	$1.3378e^4$	$1.3160e^4$

2.4.3 Overall system identification

Parameter	Ecoflex30	Dragonskin30
K	12.8994	2.7446
D	16.277	5.2404
α	0.1643	0.0810
γ	0.0014	0.0007
resnorm	0.4913	1.6916

3 Discussion of Results

- We found our steady state analysis to give us exceptionally small results for α for both fingers. This means that an increase in PWM input has very little impact on the bend angle. We believe that the issue with this linear regression comes from the fact that G(q) is not linearly related to either q or u. This means that a linear model would be a poor fit for the data
- Secondly, we attempted to calculate γ by modelling pressure to be proportional to τ. This provides incredibly high error, which makes it unusable
- We attempted to calculate our parameters by using a gridding approach to vary the starting values of γ and D, however we found that regardless of the initial value, we converged to the same result.
- Although we wrote out the model for the hyper elastic functions, we were unable to find appropriate starting values for the optimizer.
- Comparing the two systems, we find that for the γ calculation and the overall identification, the error is too high to provide a meaningful comparison. We do see from the steady state comparison, that the values are very similar. On a visual analysis, an equivalent PWM value results in more of a bend for the ecoflex material.