Robust Extended Kalman Filter Based Sensor Fusion for Soft Robot State Estimation and Control

by

Kyle Stewart

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2022 by the
Graduate Supervisory Committee:

Wenlong Zhang, Chair
Spring Berman
Sze Zheng Yong

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Soft robots provide an additional measure of safety and compliance over traditional rigid robots. Generally, control and modelling experiments take place using a motion capture system for measuring robot configuration. While accurate, motion capture systems are expensive and require re-calibration whenever the cameras are adjusted. While advances in soft sensors contribute to a potential solution to sensing outside of a lab environment, most of these sensing methods require the sensors to be embedded into the soft robot arm. In this work, a more practical sensing method is proposed using off-the-shelf sensors and a Robust Extended Kalman Filter based sensor fusion method. Inertial measurement unit sensors and wire draw sensors are used to accurately estimate the state of the robot. An explanation for the need for sensor fusion is included in this work. The sensor fusion state estimate is compared to a motion capture measurement along with the raw inertial measurement unit reading to verify the accuracy of the results. The potential for this sensing system is further validated through Linear Quadratic Gaussian control of the soft robot. The Robust Extended Kalman Filter based sensor fusion shows an error of less than one degree when compared to the motion capture system.

ACKNOWLEDGEMENTS

There are a number of people I'd like to show my appreciation for with regards to my Thesis and overall education at ASU.

First and foremost I'd like to thank my wife, Leslie, for her support throughout my research and Master's degree. I appreciate her willingness to help out when I'm stuck working late at night or when I can't take care of our kids. She kept things fun and exciting for me even when the research wasn't working out the way I had hoped.

I'd like to thank my advisor Dr. Wenlong Zhang. His feedback and direction was very beneficial as I struggled to discover a research project I wished to pursue. The corrections he gave aided me in ensuring that I reached a desirable conclusion to my research Thesis.

I'd also like to thank the other students in the RISE lab. Thank you Reed for the many conversations we had on our way to and from the lab on how to improve our research and how to achieve success in our projects. Also for his help on the many homework assignments we worked on together. Thank you George for the mentorship you provided and for the many skills you taught me through out these two years. Your knowledge on dynamics and control became a much needed resource. Thank you to the remaining lab members, especially Karishma, Emiliano, and Weija for the experiences we shared.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

Soft robots have been gaining a lot of traction over recent years within the robotics community, because of the innate safety that a soft robot provides. Traditionally when rigid robots are used in the same workspace as humans, many safety precautions are taken to ensure the protection of the human. If contact is made between a rigid robot and something fragile, damage is likely to occur. Vasic and Billard (2013) carries out a survey classifying the danger cause by rigid robots. Soft robots are flexible, soft, and adaptable, allowing them to be used in environments not suitable for rigid robots. When contact occurs between a soft robot and its environment, the soft robot will bend accordingly without any harm or injury to the contacted object or person. Much like Bicchi *et al.* (2002) safe human interaction is the driving motivation behind a number of soft robot arms. Soft robot safety opens the door to new areas where robots could enhance our everyday lives. In this paper, steps are taken towards a soft robot capable of working in any environment with accurate sensing and control. This includes applying modeling techniques, creating a sensing system including fusion algorithms, and proving the controllability of the robot using the sensor fusion approach.

In Lee *et al.* (2017), a review of potential soft robot application is carried out. Some of the applications they find include human-machine interface and interaction, locomotion and exploration, manipulation, medical and surgical applications, and

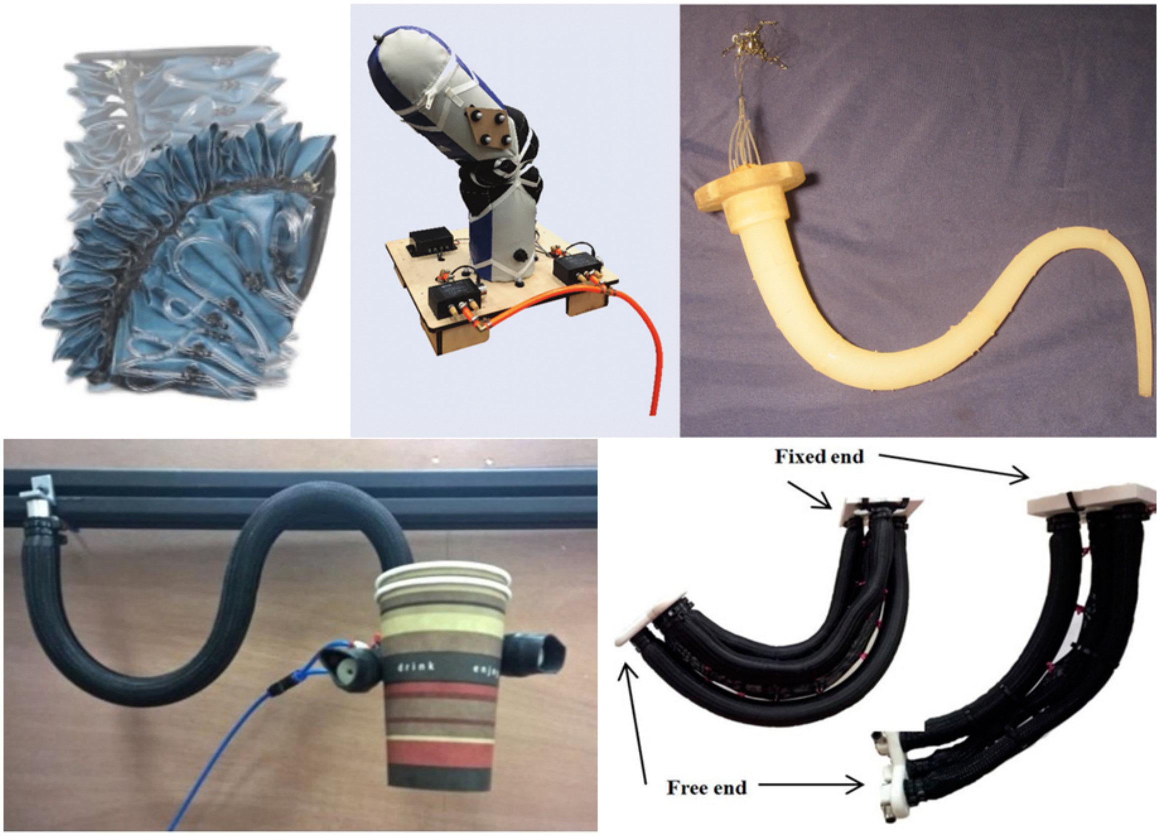


Figure 1.1: Five Soft Robots Manufactured with Unique Fabric and Polymer Materials. (Nguyen *et al.*, 2019), (Best *et al.*, 2015), (Al-Ibadi *et al.*, 2018a), (Renda *et al.*, 2014), (Al-Ibadi *et al.*, 2018b)

rehabilitation and wearable robots. The Octarm, an octopus inspired arm, manipulates objects and performs basic pick and place tasks both on ground and in the water (Grissom *et al.*, 2006). Initial concept design and testing for manipulation tasks with a soft arm in space has begun in Troise *et al.* (2021). In Best *et al.* (2015) a soft humanoid was designed and controlled for safer human-robot interaction. Visual servoing has been performed with a soft manipulator (Wang *et al.*, 2017). Soft robots may also be used in assisting the elderly or disabled with daily living tasks (Nguyen *et al.*, 2019). This project aims to assist in advancing the applications for a soft robotic manipulator.

1.2 Challenges

While the application of soft robots shows promise, the flexible nature of soft robots creates additional challenges that must be considered (Laschi, 2016). Modelling a rigid robot arm requires knowing the parameters of the links and joints of the robot. For a soft robot continuum arm, there is not a constant axis of rotation among the joints, thus traditional robot modelling techniques will fail without making additional assumptions. Obtaining accurate models and equations of motion that describe the robot as it moves through space requires novel techniques based on similar fields of research. These modelling approaches include applying mechanics of materials to describe the bending motion of the arm, making assumptions based on constant-curvature to incorporate traditional robot modelling techniques, dividing the robot into infinitesimally small masses or discs, or creating artificial neural networks and machine learning algorithms to learn the dynamics of a system (Armanini *et al.*, 2021). These modeling techniques require more formulation, calculation, simulation, or training than the traditional robot modeling approaches, thus research is done in this area to reduce the amount of modeling time and cost for a soft robot.

Before moving from dynamic models to controls, sensing must be considered. Again, when considering traditional rigid robots, sensing the state of the robot is simpler than it is for a soft robot, because of the use of encoders attached to the joints of the robot. Soft robots may not have specific joints and using a hard encoder would limit the softness of the robot. For this reason, sensing of soft robots has become another topic of interest within soft robotics. Sensing must consider both rotation about a certain frame or axis and elongation depending on the actuator design. Instead of only using angle and angular velocity as the states of the robot, some consider arc length, curvature, joint translation, or distance to the axis of rotation to be state variables. Many of the sensors in related research papers are manufactured by the researcher, which means the sensors aren't readily available for use in soft robot research. These sensors may be embedded into the system which make the manufacturability of a soft robot more difficult and complex. Section 2.2 covers related works on soft robot sensing.

Most soft robot research is done in a lab environment using some form of motion capture to sense the position of the robot as seen in figure 1.2. While extremely accurate, these systems are expensive, not portable, and take time to set-up and calibrate. Using motion capture outside of a lab environment is not practical, thus alternative state estimation techniques are necessary. Without use of motion capture, and due to the large uncertainty of a flexible soft robot, it is difficult to measure

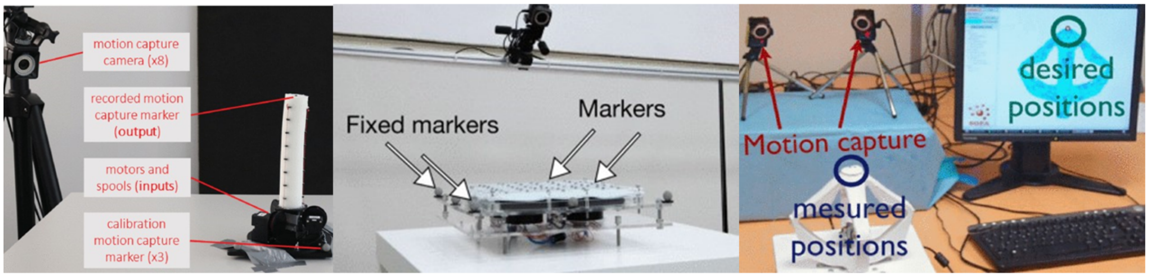

motion capture camera (x8)
recorded motion capture marker (output)
motors and spools (inputs)
calibration motion capture marker (x3)
Fixed markers
Markers
Motion capture
mesured positions
desired positions

Figure 1.2: Three Distinct Motion Capture Setups for Robot Sensing (Duriez, 2013), (Bern *et al.*, 2020), (Scharff *et al.*, 2021)

the state of a robot without some error in the readings. The material of the robot and sensors are subject to degradation over time which may cause sensor drift and inaccuracy. Currently soft robot manufacturing creates imperfections in the robot which makes state estimation more difficult. The motion of the robot is not always predictable meaning that sensor readings won't explain the exact motion of the robot. Filtering and estimation in addition to sensor readings may give more accurate results and should be considered as part of the sensing system. Apart from the accuracy of the sensor, using embedded sensors limits the transferability of the sensing system to be used on soft robots made with different materials. A brief overview of state estimation in addition to sensing is given in section 2.3. Further exploration of state estimation using multiple sensors is included.

1.3 Contributions

With the challenges of soft robot estimation and control in mind, this thesis proposes to find a practical solution to soft robot state estimation. This paper aims to generate a portable, transferable sensing system with a state estimator that can handle high nonlinearities and uncertainty in a robot model. These design focuses will save future soft robot designers time and money with no need for costly sensing techniques. The possibility for a sensing system that could be used on most if not all types of soft robots would be beneficial as well.

This Thesis proposes an improved state estimation approach that fuses three sensor measurements together with the dynamic model. The fusion takes place using a variation of the Extended Kalman Filter (EKF), the Robust Extended Kalman Filter (REKF), with the derived equations of motion. The proposed REKF filter provides a more robust estimator than other current estimation approaches. The contributions of this Thesis are as follows:

1. Realization of a portable, transferable sensor fusion system consisting of a wire draw sensor, accelerometer, and gyroscope.

2. Implementation of a Robust Extended Kalman Filter (REKF) for improved accuracy and robustness in state estimation.

3. Validation of the sensing system in control through implementation of a Linear Quadratic Gaussian (LQG) controller.

The remaining sections of this paper are divided into the following chapters: In chapter 2, an overview of the related works is given including the contributions to this paper. Chapter 3 describes the modeling approach used in this paper, the sensor selection and characterization, the state estimation algorithms, and the LQG control scheme. Chapter 4 describes the experiments and results for validating the sensor fusion algorithms. Finally, Chapter 5 sums up this work with future applications and extensions to the research completed.

Chapter 2

RELATED WORKS

2.1 Modeling

The challenge of modelling a compliant mechanism with large amounts of uncertainty and error has led to unique modeling methods that stray from using Denavit-Hatenburg (DH) parameters, forming homogenous transformation matrices and calculating traditional Jacobians for the robot arm. More evaluation on each joint individually results in much more complex models. Researchers have incorporated modeling techniques that draw from other fields of study commonly used for modeling compliant systems. In Armanini *et al.* (2021) soft robot modeling techniques are categorized into four groups: continuum mechanic models, geometric models, discrete material models, and surrogate models.

Continuum mechanic models are based on the mechanics of a material including the materials tendency to bend and stretch under a given force or torque. The manipulator is seen as having a backbone with compliant tendencies (Rao *et al.*, 2021). The material properties must be known for this approach. When multiple materials make up a soft robot or the materials don't naturally bend the continuum mechanic model will have uncertainty the model does not account for.

Geometric models make assumptions or model the shape of a robot based on its geometry. The most popular technique in this group is based on piecewise-constant curvature, a method used with continuum robots (Chawla *et al.*, 2018), (Robert J. Webster and Jones, 2010). Assuming the curvature of the rotation is consistent throughout the arm simplifies the geometric model. The assumptions made for this technique also create uncertainty within the robot, since the joints won't follow a constant curvature in all scenarios.

Discrete material models build off the other groups by discretizing the system. The robot arm will be separated into any number of discrete groupings sometimes infinitesimally small to simplify the amount of motion in each section. Finite element methods (FEM) sum up the results from all sections to create a wholistic model. The discrete material models approach improves the accuracy of geometric and continuum mechanic models, but the time required to carry out the FEM is significant and changes in the robot behavior may require additional formulation.

To better model the large uncertainty inherent in soft robots, machine learning approaches have been explored as they pertain to soft robots (Kim *et al.*, 2021a), (Wang *et al.*, 2021). Machine learning more accurately predicts all uncertainty caused by manufacturing error through large amounts of training. Much like the discrete material models, when anything in the environment changes, the system must be retrained which takes time. Further exploration into surrogate models includes order reducing algorithms.

2.2 Sensors

As mentioned previously, rigid encoders typically used in rigid robots are not practical for use in a soft robot. Encoders are too hard, too heavy, and there is no effective place to position them due to the variable axis of rotation of continuum joints. Researchers have turned to much less conventional sensing techniques that retain the compliant nature of the robot with slight changes to the impedance. Lee *et al.* (2017) describes the manufacturing of microfluidic sensors that measure changes in capacitance and inductance to describe the motion of the joint. This is put in practice in a silicon-elastomer based soft robot (Kim *et al.*, 2021b). Plastic optical fibers (POF) measure curvature in Lunni *et al.* (2018), by shining light into the fibers and measuring how much light reaches the end of the fiber. As the robot rotates, so do the fibers, which admit less light then in the straightened position. An embedded strain sensor attached throughout a joint also measures curvature assuming constant curvature (Loo *et al.*, 2019a). Hyatt *et al.* (2019) proposes a global measurement for orientation based on the Vive motion tracker and IMU sensors attached to the end of each joint. It is important to note that soft fabric based manipulators are more accommodating to rigid sensors than polymer based soft robots. The majority of these sensors are embedded in the robot meaning the sensors are positioned during manufacturing.

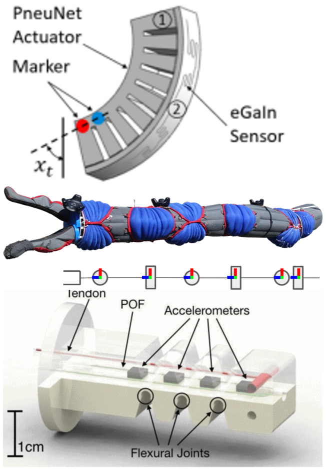


Figure 2.1: Top: Embedded Strain sensors. Middle: Vive Trackers. Bottom: Embedded POFs and Accelerometers.

2.3 State Estimation

Accurate sensing is difficult without the use of external sources such as a motion capture system. To improve the state estimation using on board sensors, signal filtering can be used to better estimate the state of the robot. A Bayesian network filters noisy sensor signals and incorporates hysteresis into the model for improved state estimation (Kim *et al.*, 2021b). Loo *et al.* (2019a) uses the EKF to combine a model, obtained through machine learning, with strain sensor measurements. Hyatt *et al.* (2019) proposes a new method for calibrating multiple unrelated sensor measurements for improved state estimation and global orientation estimation. An Adaptive Extended Kalman Filter integrates optical measurements with a simplified steady-state model (Lunni *et al.*, 2018). In Loo *et al.* (2019b) an H-infinity based EKF very similar to the REKF is used to simulate the state estimation for both bending and extension in a highly nonlinear soft robotic system. Fusing multiple sensors together into a single estimate may provide a more accurate sense of the where the robot is in space. This is one area of state estimation that has yet to be thoroughly explored. Sensor fusion is used however, to estimate the force and position of contact with capacitive sensors (Navarro *et al.*, 2020). Further exploration into the optimization of sensor placement and estimation accuracy can improve the sensing system results (Rupert *et al.*, 2021).

2.4 Control

As control is not the main contribution of this paper, an extensive overview of the control of soft robots is not given. Instead, a quick summary is offered. Due to the high uncertainty and potential for high non-linearity of a soft robot, a large number of control schemes have been attempted. These control schemes can be separated into three distinct groups, as shown in Wang and Chortos (2021), namely: open-loop control, closed-loop control and autonomous control. Open loop control methods don't require sensors. Model predictive control, impedance control, optimal linear quadratic regulator control, sliding mode control, are but a few commonly used closed-loop controllers. Autonomous controllers include machine learning, artificial intelligence and reinforcement learning control schemes. All these methods have shown suitable for soft robot control. This Thesis aims to provide more accurate state estimation leading to more reliable control in the future.

Chapter 3

METHODS

In this chapter, the modelling approach and sensor selection are described at length. This includes the derivations of the equations of motion, placing those equations in a compatible form to be used in the REKF and describing the equations used to sense rotation of the arm. The state estimation algorithms are then demonstrated including the alteration made for the REKF from the EKF. Lastly, an LQG controller based on the REKF is provided for further validation of the state estimation system.

3.1 Robot Design

This research uses an elephant trunk inspired soft robot joint as shown in figure 3.1, based on the work from Nguyen *et al.* (2019). The robot is made up of a fabric

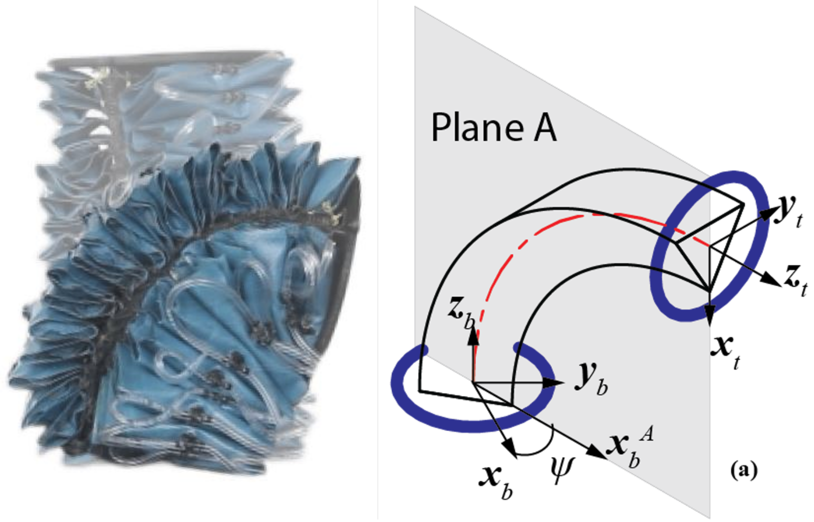


Figure 3.1: Elephant Inspired Soft Manipulator and Joint Frames for Describing Motion

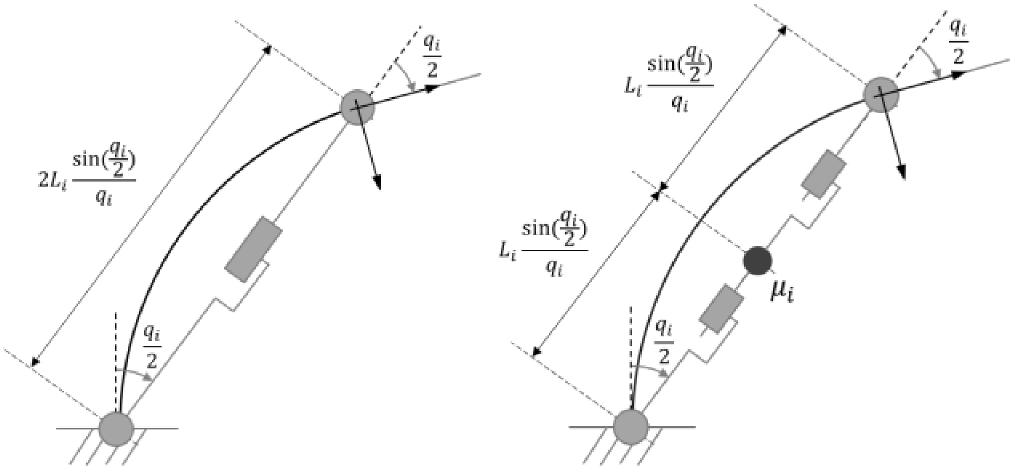
spine which is rigid enough to provide curvature without extension when actuated. The actuation is provided by inflating the blue woven thermoplastic polyurethane pockets in series forcing the rigid fabric into a bending motion. In future work, this robot joint can be combined with similar joints to create a soft robot arm with a large range of motion. All modeling, state estimation, and control derivations are formed with this actuator in mind.

This robot has infinite degrees of freedom allowing it to bend in all directions. This is explained in Figure 3.1 where the variable ψ measures the angle about the z_b axis from x_b to the projection of the direction of bending x_b^A on the base axis. The angle $theta$ measures the rotation angle of the actuator. Due to the complexity of this problem this work focuses on planar motion where the angle ψ is set to 0.0.

3.2 Modeling

Referring back to the four groups of modeling in section 2.1, namely continuum mechanic models, geometric models, discrete material models, and surrogate models, a modeling method was selected for this research based on formulation time while avoiding the direct use of material properties that made up the robot. A design that does not require a large amount of simulation, training, or calculation is beneficial in scenarios where the environment of the robot changes. With this known, the hybrid approach that focuses on the geometric curve of the joint is selected. This model develops an equation of motion based on the geometry of the robot, and some of the parameters are evaluated using a grey-box model approach. It's important to note that while material properties are not dealt with directly, the material properties effect the parameter estimation as is discussed in the next few paragraphs.

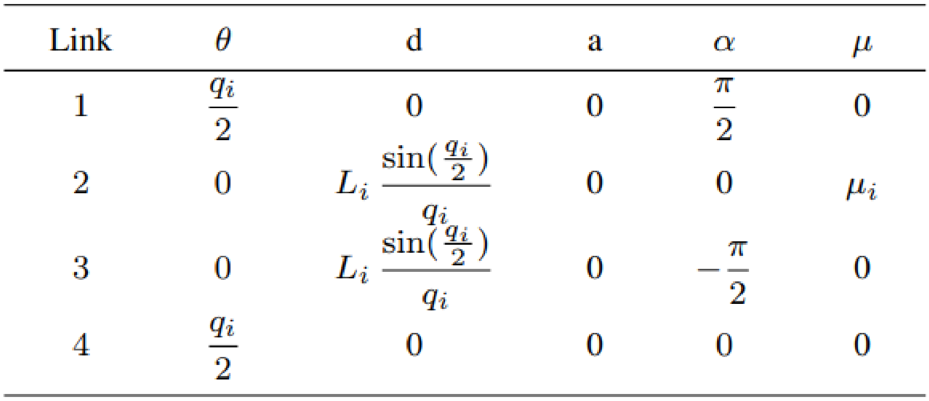
With the model design requirements noted, the piecewise constant curvature (PCC) geometric model was selected. This model uses the assumption of constant curvature to divide the curved joint into a system of joints that approximate the curve. The assumptions and background described for this modeling technique are clearly explained in Della Santina *et al.* (2018). While the constant curvature assumption does simplify the model, it also creates some amount of uncertainty in the projected motion of the arm. Figure 3.2 shows the simplification of a joint of constant curvature by taking the compliant joint and dividing it into three or four joints in series. The three joints are ordered as a revolute joint followed by a prismatic joint followed by an additional revolute joint (RPR). The four joints are ordered as a revolute joint, two prismatic joints with an additional revolute joint (RPPR). The four joint simplification is used in this paper. Table 3.1 shows the resulting Denavit Hartenberg (DH) parameters for a single soft joint where q represents the rotation angle and μ is the point mass.. The mass and inertial properties are estimated by


$2L_i \frac{\sin(\frac{q_i}{2})}{q_i}$
$\frac{q_i}{2}$
$\frac{q_i}{2}$
$L_i \frac{\sin(\frac{q_i}{2})}{q_i}$
$L_i \frac{\sin(\frac{q_i}{2})}{q_i}$
$\frac{q_i}{2}$
μ_i
$\frac{q_i}{2}$

(a) RPR **(b) RPPR**

Figure 3.2: Generalization of a Continuum Joint of Infinite Freedoms to a Three (a) or Four (b) Prismatic and Revolute Joint Approximation.

Table 3.1: DH-Table for the RPPR Approximation of a Single Soft Joint.



Link	θ	d	a	α	μ
1	$\frac{q_i}{2}$	0	0	$\frac{\pi}{2}$	0
2	0	$L_i \frac{\sin(\frac{q_i}{2})}{q_i}$	0	0	μ_i
3	0	$L_i \frac{\sin(\frac{q_i}{2})}{q_i}$	0	$-\frac{\pi}{2}$	0
4	$\frac{q_i}{2}$	0	0	0	0

assuming a point mass exists between the prismatic joints.

With the order of revolute and prismatic joints established and the DH-parameters evaluated, the kinematic relationships and dynamic equations of motion can be derived. The dynamics for planar motion of this soft robot are given by

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau - Kq - D\dot{q} \tag{3.1}$$

$$\tau = \alpha R(\phi)p \tag{3.2}$$

$$R(\phi) = \begin{bmatrix} -\sin(\pi/6) & -\sin(\pi/6) & 1 \\ \cos(\pi/6) & -\cos(\pi/6) & 0 \end{bmatrix} \tag{3.3}$$

where q, $\dot{q}$, $\ddot{q}$, are the states variables or bending states of the robot. M, C, and G are the inertia matrix, Coriolis and centrifugal matrix, and gravitational matrix respectively. A spring term K and damping term D are added to the dynamics due to the tendency for the robot to naturally return to the upright position when no pressure is applied. The torque caused by pressure *tau* is obtained through multiplying a pressure constant α to a transformation matrix $R(\phi)$. The unknown terms K, D, and α are found using the nonlinear system identification toolbox in MATLAB. The derivation of M, C, and G are shown in full in Cosimo's work (Della Santina *et al.*, 2018) and the equations are included in Appendix B.

3.3 Sensor Selection

One goal of this research project was to create a state estimation system that is transferable between different soft robot platforms. As such, the IMU and wire draw sensors proposed for this project can be placed on the exterior of any soft robot platform. The IMU shown in figure 3.3, containing an accelerometer, gyroscope, and magnetometer provides multiple sensor in one compact device. The draw sensor, also shown in figure 3.3, measures the length of the wire that has been extracted. The IMU sensor has proven to be beneficial for measuring orientation and rotation of a system. While each component contributes to the overall accuracy of the IMU sensor, the accelerometer and magnetometer produce errors depending on how and where the device is used. The accelerometer at times has an issue when linear accelerations occur. For now the device will be used in stationary environments, thus the accelerometer is included. Depending on the environment and electronics surrounding the IMU sensor, the electromagnetic readings may become distorted. Thus the magnetometer is excluded from the design.

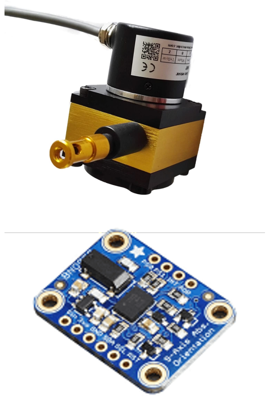


Figure 3.3: Top: Wire Draw Sensor. Bottom: Inertial Measurement Unit (IMU)

While each of the proposed sensors measures the rotation angle of a robot, there are inherent errors, which are minimized by the use of sensor fusion. The accelerometer and gyroscope are subject to drift and noise issues respectively. Over time, the error caused by the IMU will become large enough causing the sensor measurement to eventually be unusable. These sensors however are very accurate when measuring slow changes in orientation. The main source of accelerometer and gyroscope error shown in this project is primarily caused by alignment issues between the two IMUs used. The wire draw sensor is not subject to drift, but due to mechanical constraints is subject to timing delay (latency) errors (figure 3.4). This is caused from the internal friction of the recoiling mechanism and contact points of the wire. Further design adjustments alleviates this error in this paper, however when multiple actuators are joined together to form a long arm, friction is likely to delay the draw sensor readings. Sensor fusion creates a more consistent state estimation, because when the accelerometer and gyroscope experience drift, the draw sensor accounts for the error. Likewise, when the draw sensor has latency issues, the accelerometer and gyroscope counteract the delay. Sensor fusion

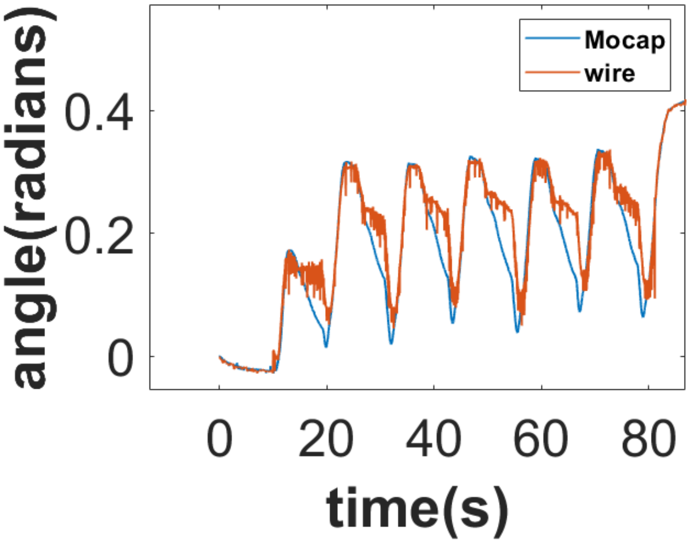


Figure 3.4: Latency Issues from the Draw Wire Sensor Due to Friction.

should portray more accurate estimates than each sensor individually.

Each sensor is manipulated to derive the angle of rotation. Therefore the vector z is

$$z = \begin{bmatrix} \theta_{accel} \\ \theta_{gyro} \\ \theta_{wire} \end{bmatrix} \tag{3.4}$$

The bno055 IMU sensor gives raw accelerometer and gyroscope data. Two IMU sensors, one on the base and on the robot, are used for accelerometer and gyroscope readings. For the accelerometer, the rotation about the x-axis of the robot is evaluated by subtracting the gravitational force on the IMU on the robot by the IMU on the base. This is given by

$$\theta_{x,base} = \arctan 2(-a_{y,base}, a_{z,base}) \tag{3.5}$$

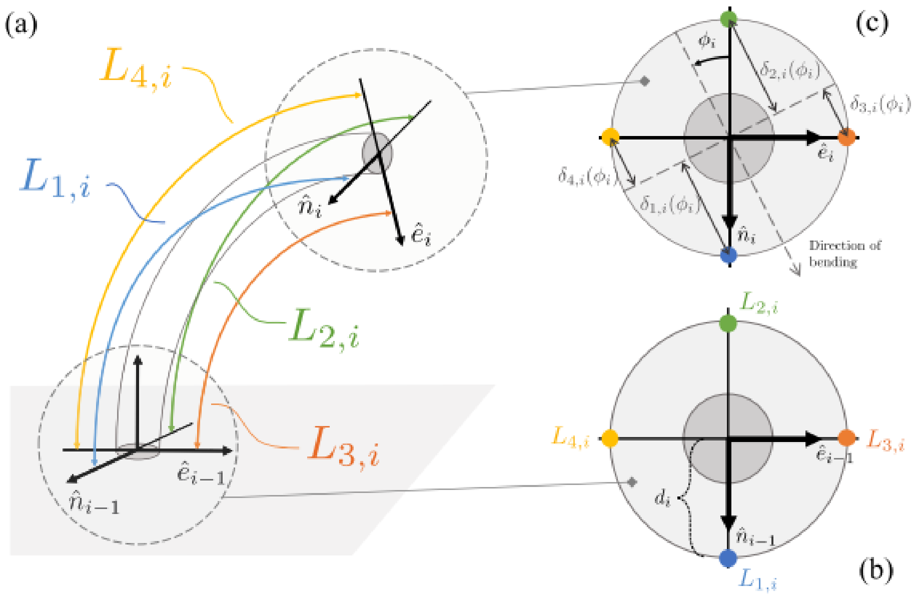


Figure 3.5: a) Wire Draw Sensor Location for Measuring Rotation in All Directions. b) Distance from Sensor to Bending Axis. c) Depiction of *delta*. (Not used in this work)

$$\theta_{x,robot} = \arctan 2(-a_{y,robot}, a_{z,robot}) \tag{3.6}$$

$$\theta_{accel} = \theta_{x,robot} - \theta_{x,base} \tag{3.7}$$

The rotation given by the gyroscope is acquired directly from the IMU. The python package used calculates rotation from rotation rate.

The draw wire sensors require the constant curvature assumption to calculate the angle given the arc length measurements from the wire as shown in Della Santina *et al.* (2020) (figure 3.5). Note that since the robot only experiences planar motion, only two draw wire sensors are needed to calculate the bending angle based on arc length. Thus the angle of rotation is given by

$$\theta_{wire} = (L_1 - L_2)/(2 * d_r) \tag{3.8}$$

where L_1, L_2 are the draw sensor readings, and d_r is the radial distance from the wire to the center of bending which is assumed to be at the center of the robot cross-section.

3.4 Extended Kalman Filter

Before discussing the EKF, the dynamic equations must be linearized, discretized, and placed in state space form. Placing equations (3.1)-(3.3) in state space form with $x_1 = q$ and $x_2 = \dot{q}$ gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ M^{-1}(-Cx_2 - G - Kx_1 - Dx_2 + \alpha R(\phi)p) \end{bmatrix} \tag{3.9}$$

Discretizing the equations results in

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = f(x_{k-1}) = \begin{bmatrix} x_{1,k-1} + x_{2,k-1}\Delta t \ x_{2,k-1} + M^{-1}(-Cx_2 - G - Kx_1 - Dx_2 + \alpha R(\phi)p)\Delta t \end{bmatrix} \tag{3.10}$$

The EKF requires a linearization of the state equations. Thus the linearization A is

$$A = \frac{\partial f(x)}{\partial x_{k-1}} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \tag{3.11}$$

$$A_1 = \frac{\partial x_{1,k}}{\partial x_{1,k-1}}\bigg|_{x=\hat{x}_{k-1}} = 1 \tag{3.12}$$

$$A_2 = \frac{\partial x_{1,k}}{\partial x_{2,k-1}}\bigg|_{x=\hat{x}_{k-1}} = \Delta t \tag{3.13}$$

$$\begin{aligned} A_3 = \frac{\partial x_{2,k}}{\partial x_{1,k-1}}\bigg|_{x=\hat{x}_{k-1}} &= M^{-1}\bigg(- \frac{\partial C}{\partial x_{1,k-1}}\bigg|_{x=\hat{x}_{k-1}} x_{2,k-1} - \frac{\partial G}{\partial x_{1,k-1}}\bigg|_{x=\hat{x}_{k-1}} - K\bigg)\Delta t \\ &+ \frac{\partial M^{-1}}{\partial x_{1,k-1}}\bigg|_{x=\hat{x}_{k-1}} \bigg(- Cx_{2,k-1} - G - Kx_{1,k-1} - Dx_{2,k-1} + \alpha R(\phi)p\bigg)\Delta t \end{aligned} \tag{3.14}$$

$$A_4 = \frac{\partial x_{2,k}}{\partial x_{2,k-1}}\bigg|_{x=\hat{x}_{k-1}} = 1 + M^{-1}(-\frac{\partial C}{\partial x_{2,k-1}}\bigg|_{x=\hat{x}_{k-1}} - C - D)\Delta t \tag{3.15}$$

where $\frac{\partial M^{-1}}{\partial x_{1,k-1}} = -M^{-1}\frac{\partial M}{\partial x_{1,k-1}}M^{-1}$

The Extended Kalman Filter (EKF) is chosen due to its low memory usage. Only values from the previous time step are necessary for the EKF. Since the equations of motion are non-linear, the Extended Kalman filter (EKF) is considered over the Kalman Filter. The discrete state space equations are now of the form

$$x_k = f(x_{k-1}, u) + w_{k-1} \tag{3.16}$$

$$z_k = h(x_k) + v_k \tag{3.17}$$

where x are the states, u is the input, f and h are the dynamics and observation terms, w, v are the system noise.

The EKF consists of a two step process. In the first step, a prediction is made using the dynamics from equations (3.1) - (3.3). The system covariance is also estimated for making corrections to this approach in the next step. These prediction equations are

$$\hat{x}_k^- = f(x_{k-1}) \tag{3.18}$$

$$P_k^- = A_{k-1}P_{k-1}A_{k-1}^T + Q \tag{3.19}$$

$\hat{x}_k^-$ and P_k^- are the state and covariance predictions using the previous values. A is the linearized matrix derived in equations (3.11) - (3.15). Q represents the process covariance and R, which will be used later on is the observance covariance.

The second step corrects the predictions using the Kalman gain L, sensor measurements z and observer covariance R. The updated state estimate $\hat{x}_k$ and covariance estimate P_k are now give by

$$L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \tag{3.20}$$

$$\hat{x}_k = \hat{x}_k^- + L_k(z_k - h(\hat{x}_k^-)) \tag{3.21}$$

$$P_k = (I - L_k H_k)P_k^- \tag{3.22}$$

H_k is the linearized matrix of $h(x_k)$. The observation equation is assumed to be linear, thus $h(x_k) = Hx_k$. I is the identity matrix. A condensed flow of the EKF is shown in Appendix A.

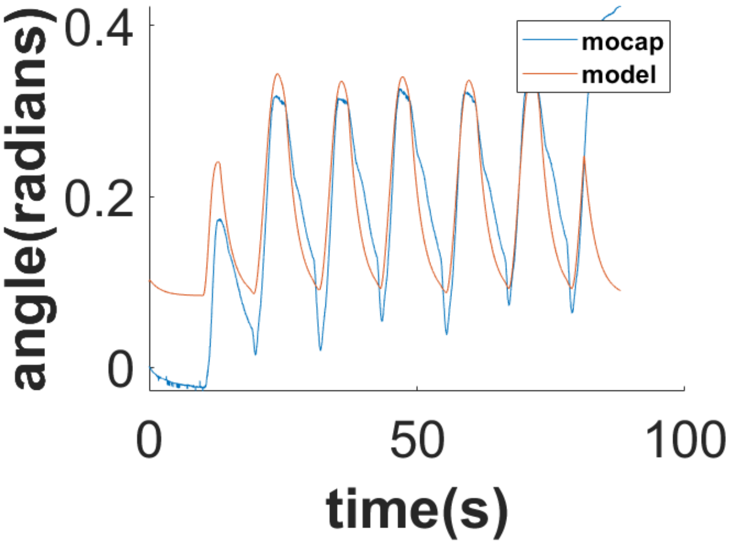
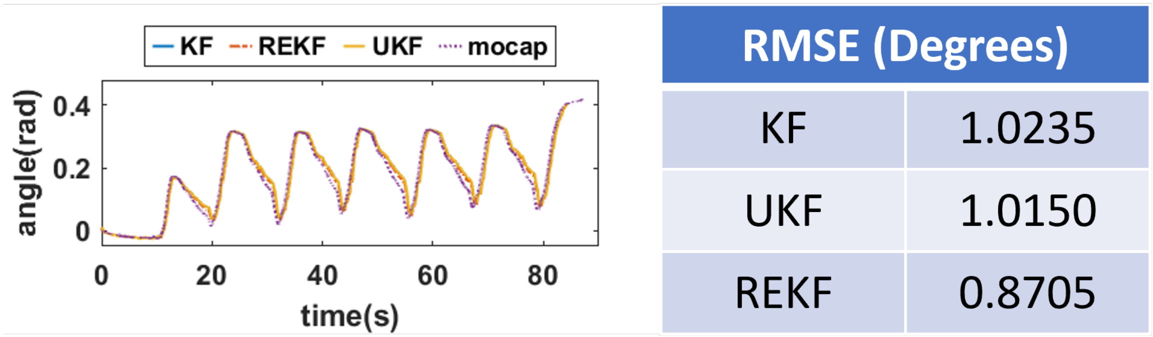


Figure 3.6: Validation of the Proposed Dynamic Model in Response to a Sine Wave Input

3.5 Robust EKF

The EKF, while beneficial in some scenarios, struggles to obtain estimates to highly nonlinear systems or systems with large uncertainty. The estimates diverge quickly due to the linearization of the nonlinear system. Such is the case with the soft robot used in this work, the model of which can be seen in Figure 3.6. As shown, errors occur when deflating the robot in the prediction model prove there exists uncertainty with the dynamic equations. The deflation is difficult to model due to no internal torque from the pressure forcing the soft robot back into the upright position. Instead the model estimates the stiffness and damping terms, which aren't as accurate. This uncertainty results in solutions that diverge immediately.

Instead of continuing with the EKF, variations to the EKF are considered. These variations have been designed to overcome the non-linearities of a dynamic system. Figure 3.7 shows a brief comparison experiment between two variations of the EKF:



RMSE (Degrees)	
KF	1.0235
UKF	1.0150
REKF	0.8705

Figure 3.7: Quick Comparison of Variations to the EKF Including Root Mean Square Error.

The Robust Extended Kalman Filter (REKF) and the Unscented Kalman Filter (UKF). The EKF results are not included, since the filter diverges immediately, but the KF based on kinematics is shown to prove the EKF variations improve the results. The error during deflation is due to the friction error described in section 3.3. This comparison does not constitute as proof that the EKF outperforms the UKF, but rather demonstrates that both filters are viable options. The UKF improves the accuracy of the filter through the use of multiple predictions, while the REKF improves the stability of each estimate through results from H-infinity control. The UKF derivation along with a comparison of each algorithm with the EKF is shown in Appendix C and A respectively. Since the EKF was seen as unstable for the robot presented in this paper, the Robust Extended Kalman Filter will be used for state estimation and control in this work. The UKF shows improvement, but the problem of divergence persists unless a sampling time of $< .0001$ is used which exceeds the operating speed of the raspberry pi described in chapter 4.

The EKF attempts to achieve local optimality through the underlying Riccati equation. This equation however is not always stable, because there is no guarantee the solutions to the Riccati equation are positive definite. The Robust Extended

Kalman Filter improves the stability of the design by adding a positive definite term to the Riccati equation as first proposed in Einicke and White (1999). An update to the signal model (3.16)-(3.17) is given by

$$x_k = f(x_{k-1}, u) + B_t w_{k-1} \tag{3.23}$$

$$z_k = h(x_k) + D_t w_k + v_k \tag{3.24}$$

where B_t and D_t are the linearized process noise and observation noise matrices respectively. The change to the Ricatti equation results in the following adjustments to the prediction equations (3.18)-(3.19)

$$\hat{x}_k^- = f(x_{k-1}) + B_t Q D_t^T \bar{R_t}^{-1}(z - H_k f(x_{k-1})) \tag{3.25}$$

$$P_k^- = (A_k - B_t Q D_t^T \bar{R_t}^{-1} H_k) P_{k-1} (A_k - B_t Q D_t^T \bar{R_t}^{-1} H)^T + B(Q - Q D_t^T \bar{R_t}^{-1} D_t Q) B^T \tag{3.26}$$

where

$$\bar{R}_t = R + D_t Q D_t^T \tag{3.27}$$

Equations (3.20)-(3.22) remain the same for the REKF. These updated equations obtain a stable solution more often than the traditional EKF.

3.6 LQG Control

To further prove the validity of this state estimation system, the REKF is used in Linear Quadratic Gaussian (LQG) Control. LQG was selected as the controller due to its simplicity and ease of use with the EKF or REKF. This controller was not chosen to outperform the many currently presented in soft robotics work. The experiments explained in Chapter 4 will evaluate the accuracy of this state estimation approach and the accuracy of this approach enveloped in a controller. Thus in this section, the LQG formulations, using the REKF instead of the EKF are discussed.

Traditionally, the LQG controller combines the EKF with a Linear Quadratic Regulator (LQR) controller. In this LQG controller LQR is used with the REKF including the linearized equations as previously discussed. Such can be seen in figure 3.8. While this may not be the most accurate controller used within the soft robotics community, the implementation of this controller is simple and effective. The following discrete cost function is minimized to achieve the optimal solution:

$$J = x_N^T F x_N + \sum_{i=0}^{N-1} (x_i^T Q_i^* X_i + u_i^T R_i^* u_i) \tag{3.28}$$

with the gain F equal to zero for simplicity. The controller than uses the same equations for the REKF (3.20)-(3.22) and (3.25)-(3.26). The LQG controller is

$$e_k = x_{d,k} - \hat{x}_k \tag{3.29}$$

$$u_k = -K_k e_k \tag{3.30}$$

$$x_k = f(x_{k-1}) + B_t u + L_k(z - H_k \hat{x}_k^-) \tag{3.31}$$

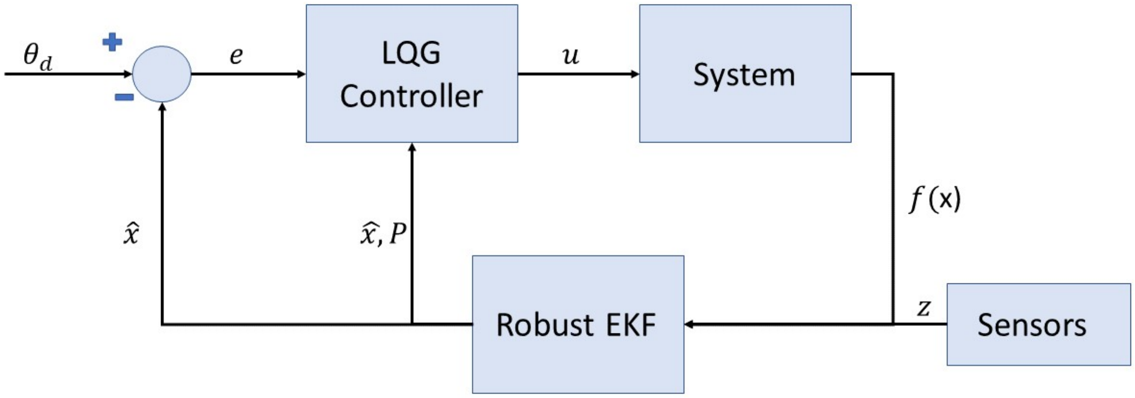


Figure 3.8: LQG Block Diagram with Robust EKF

e represents the error between the desired state $x_{d,k}$ and the current state x_k. The LQG controller gain K is evaluated by

$$K_k = (B_t^T S_{k+1} B + R^*)^{-1} B_t^T S_{k+1} A_{k-1}) \tag{3.32}$$

$$S_k = A_{k-1}^T (S_{k+1} - S_{k+1} B_t (B_t^T S_{k+1} B_t + R^*) B^T S_{k+1}) A + Q^*; \tag{3.33}$$

It is important to note the difference between the covariance matrices Q, R and LQG gains Q^*, R^*. The equation for S_k is derived by following the Riccati equation backwards in time. The gains were selected through trial and error beginning with identity matrices and increasing one matrix at a time.

Chapter 4

TESTS AND RESULTS

The goal of this paper is to provide a sensor fusion system comparable to that of motion capture. This allows soft robots to be used in all environments rather than only in a lab. To achieve this goal, as described in this chapter, tests were run to compare the results of the proposed sensor fusion estimation with motion capture readings. Results are also compared to the on board filtered IMU data to verify if the REKF estimate improves upon the IMU data alone. The purpose is not to create a sensing system more accurate than motion capture, but rather a system that comes close the accuracy of motion capture. To prove the adaptability of these algorithms, LQG control experiments are performed as explained in this chapter.

4.1 Experimental Setup

The two wire sensors are placed on opposite sides of the robot. The wire housing is secured to the base and the wire stretches up to the end of the robot as shown in figure 4.1. Note: the wire sensor attached to the back of the robot is not visible in this photo, and an additional IMU is placed at the base. The wires are secured to the body of the robot to ensure an accurate measurement of the arc length. One IMU sensor is placed on the base of the robot and the other is placed on the tip of the robot. The

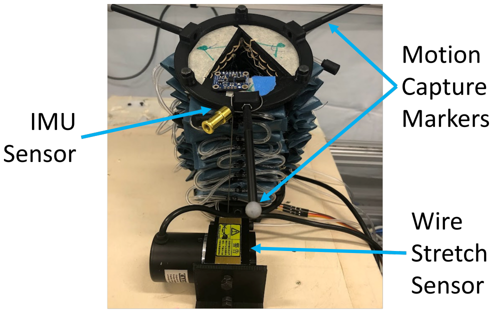


Figure 4.1: Sensor Placement

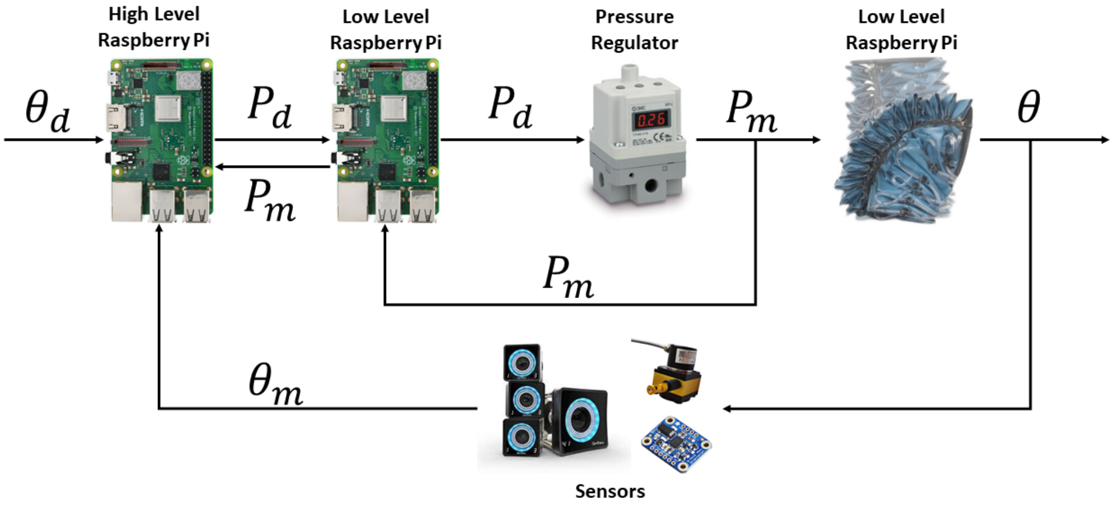


Figure 4.2: Physical Controller Components

orientations of the IMU sensors are matched at the origin to avoid positioning errors.

The robot as described in 3.1, is controlled using two raspberry pis. The first raspberry pi or the low-level controller sends the desired pressure values to three SMC ITV1000 series proportional valves. The low-level controller also measures the pressure in each actuator and sends this information to the high-level controller. The desired pressure readings are sent from the high-level controller to the low-level controller. This high-level controller receives the pressure readings, motion capture readings, and sensor measurements then performs all algorithms described in 3. The wire sensor readings are received through an ADS1263 analog to digital converter and the IMU sensors communicate the sensor measurements with the high-level controller through I2C communication. The communication between the raspberry pi's and the motion capture system takes place through Ethernet connections. Figure 4.2 shows the physical components that make up the controller.

There are two control options that are used in this research. The first is open-loop control where pressure values are controlled to follow a desired pressure trajectory. The cycle time for the raspberry pi for open-loop control is 0.123 seconds. The second option involves closed-loop LQG control where all controller calculations are carried out on the high-level raspberry pi. The closed loop configuration has a cycle time of 0.0997 seconds. The somewhat slow cycle time for the open and closed-loop methods is due to the large amount of calculation needed to linearize the equations each iteration. In both cases, the REKF is running on the high-level controller to estimate the state of the robot at all times. The open-loop option is used for REKF comparisons with the motion capture readings, and the closed-loop option is used for further validation that control is possible given a portable state estimation approach.

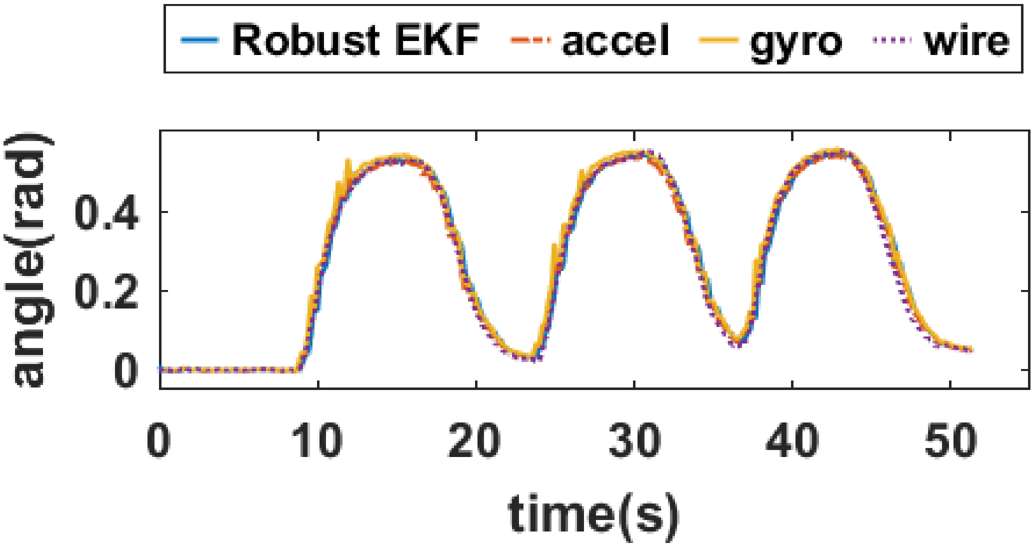


Figure 4.3: Fusion of Each Sensor with Respect to a Step Response

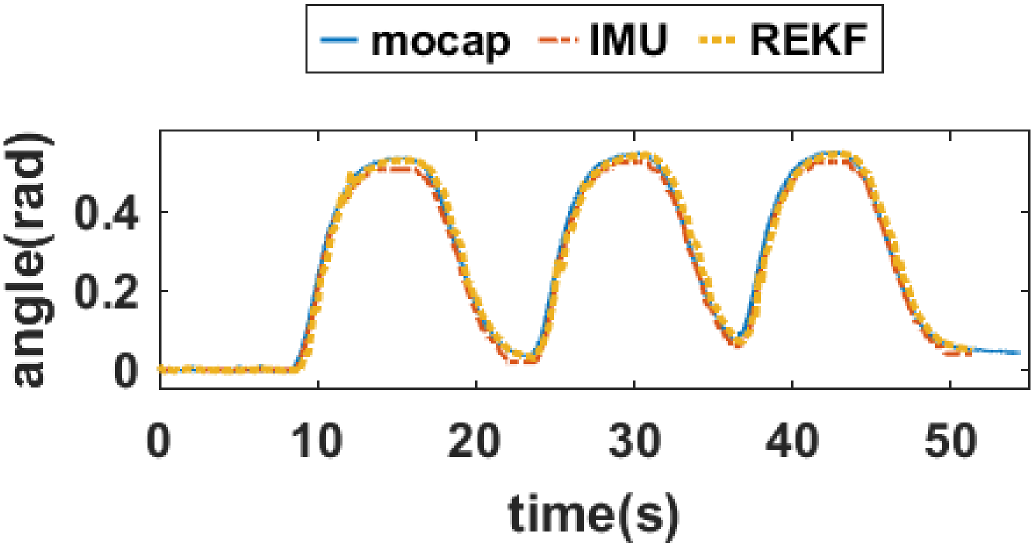


Figure 4.4: State Estimate Compared to Motion Capture (Ground Truth) and Filtered IMU Measurements in Response to a Square Wave.

4.2 Sensor Fusion

The first experiment aims to explore the accuracy of the REKF. A simple, open-loop, square wave and sine wave are implemented while the motion capture and REKF estimates are recorded. The low-level controller commands the pressure to a maximum amplitude of 25 psi in both experiments. Figure 4.3 shows the fusion results from the step input. Each sensor is displayed individually with the REKF Fusion represented as well. All sensors give the same relative measurement to one another as can be seen by the wire plot covering the other measurement plots for the peaks and valleys of the response. The wire draw sensor gives a noisy reading due to the large resolution needed to convert the analog signal to a digital signal for the raspberry pi to read. There are some minor jumps recorded for the accelerometer and gyroscope, but these jumps are minimized by the REKF.

The square wave comparison between the Motion capture set up, IMU and Robust

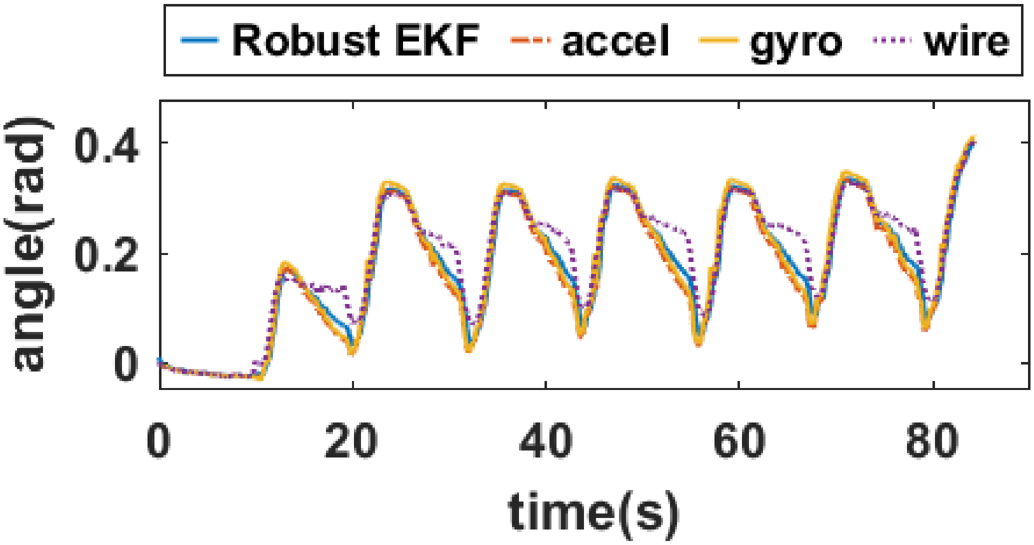


Figure 4.5: Fusion of Each Sensor with Respect to a Sine Response

EKF are shown in figure 4.4. The REKF sensor fusion estimate outperforms the IMU filtered data and follows the step response very closely. The noise in the signal is likely caused by resampling issues to match each time-step between sensors. Regardless of the minor noise, the sensor fusion estimate has a root mean square error (RMSE) of less than 1 degree as seen in table 4.1.

The response to the sine wave input (figures 4.5-4.6) is similar to the square wave except for the friction shown by the wire sensor. On this particular test, the wire was required to make a 90 degree turn immediately on leaving the housing. On future tests this design was improved to avoid this additional friction. Despite the delay in the wire readings, it is interesting to note how the inaccuracy shown by the wire sensor does not translate to the REKF results as indicated in the RMSE values from table 4.1. This proves the benefit provided by sensor fusion over any single sensor. When the wire sensors are inaccurate, the sensors from the IMU correct those inaccuracies. Notice that the sensor fusion approach outperforms the IMU filtered measurements

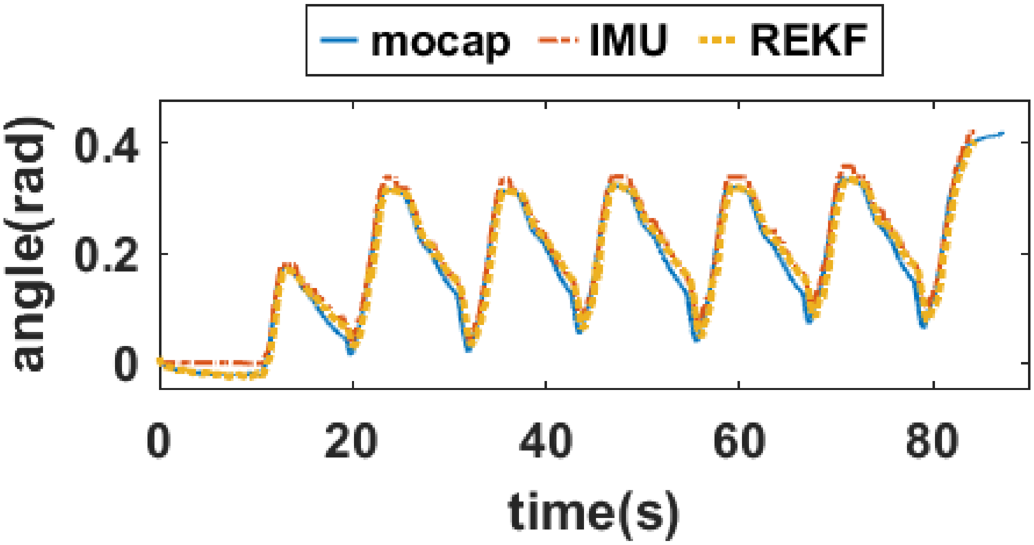


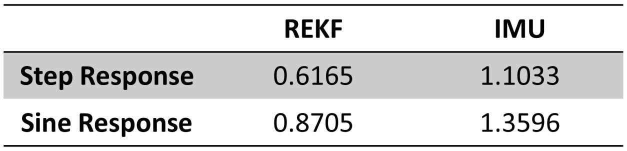
Figure 4.6: State Estimate Compared to Motion Capture (Ground Truth) and Filtered IMU Measurements in Response to a Sine Wave.

in both tests. This test also demonstrates how the uncertainty of the model (figure 3.6, especially during deflation, is overcome by the sensor measurements and REKF.

4.3 REKF Limitations

The next experiment intends to find where the REKF struggles to obtain an accurate estimate. This requires exploring the extremes of the robot such as the maximum bending angle and the highest operating frequency. This particular actuator reaches it's maximum bending angle at or near 35 psi. Due to the slow nature of pneumatically driven soft robots, the maximum frequency tested is .5 Hz. Figure 4.8 shows

Table 4.1: RMSE Values in Degrees for the Square Wave (Step Response) and Sine Wave (Sine Response).



	REKF	IMU
Step Response	0.6165	1.1033
Sine Response	0.8705	1.3596

the results from various open-loop configurations with pressure amplitudes ranging from 10 psi to 35 psi and frequencies ranging from .05 to .5 Hz.

The major takeaway from figure 4.8 on page 35 was discovered when the frequency increases. As the frequency increases, the REKF estimate is generally below the motion capture measurements. Besides the highest frequency case, the REKF is only inaccurate when approaching the maximum amplitude of the desired response. Since the increase in frequency is the limiting factor for the sensor fusion method proposed, a BODE plot would be helpful in future work to designate the exact frequency when the REKF is no longer sufficient for estimating the state. Even with the inaccuracies shown, the REKF estimates are fairly accurate compared to the motion capture system.

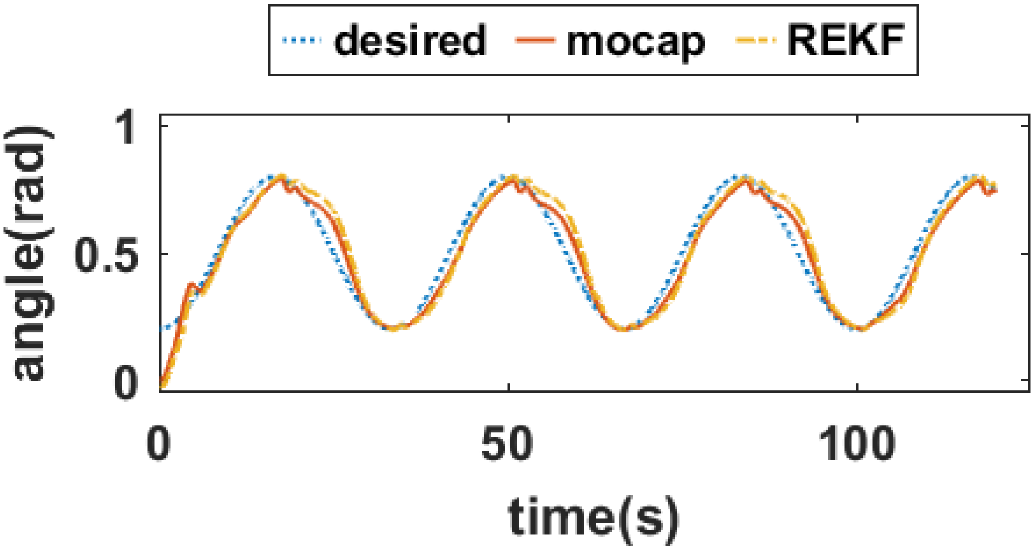


Figure 4.7: LQG Control of a Sine Wave Input with a Frequency of .05 and Maximum Amplitude of 0.8 Radians.

4.4 LQG Control

As mentioned previously, the purpose of exploring the capabilities of sensor fusion on control doesn't focus on creating a controller that outperforms the numerous, currently accepted, soft robot control techniques. Rather the hope is this section provides further exploration on the capabilities and applications of the proposed sensor fusion approach. A desired sine function is input into the LQG controller and the results can be seen in figure 4.7. The controller proves that the REKF state estimation approach works in succession with an LQG controller, and the controller reaches the desired trajectory as shown by the motion capture plot.

A drawback to soft robot control is the inflation and deflation time of the robot. In this control setup there is no vacuum to aid in the deflation of the arm. This results in slower descents than expected by the dynamic model predictions and controller. This is represented in the response of the controller after approaching the maximum amplitude of the sine wave. In this test the controller tracks a trajectory with a frequency of less than .1 Hz. To increase the frequency of the response with successful control, improvements need to be made to the cycle time of the raspberry pi. Currently the real time calculation of the linearized dynamics is too computationally expensive to achieve good results at a higher frequency. This is left to future work. With the limitations stated, the results from the LQG control experiment represent a step towards full proprioception of a soft robot arm for improved control in any environment.

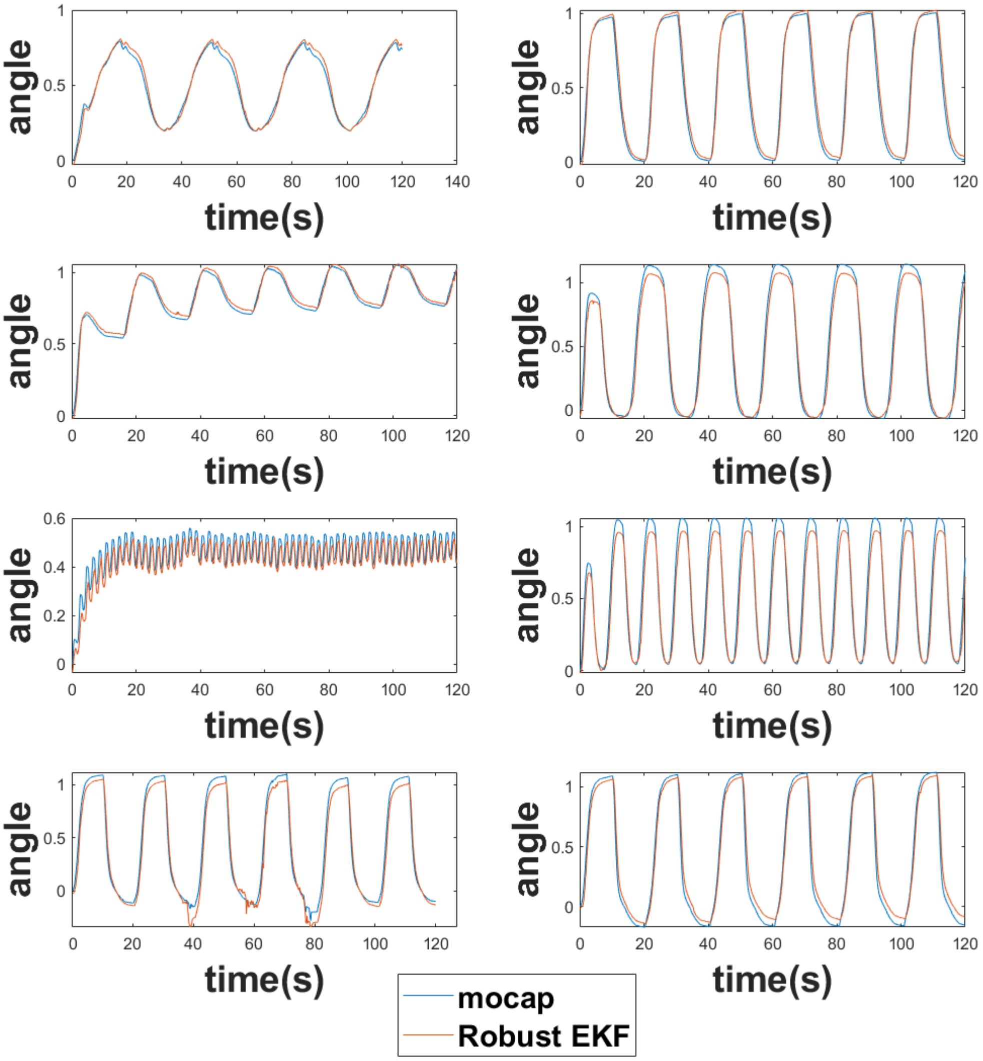


Figure 4.8: Tests that Vary in Amplitude and Frequency. Angle is Measured in Radians, and Time in Seconds

Chapter 5

CONCLUSION

This paper has proposed a sensor fusion approach for state estimation of soft robot arms based on the Robust Extended Kalman Filter (REKF). A model was derived following the piecewise constant curvature assumption. The sensing system incorporated is cost-effective, portable, and adaptable to most soft robot platforms. The REKF was compared to a motion capture system and results in error of less than one degree. The accuracy of the REKF proves the potential for this system to be used outside of a lab environment. The REKF is tested with a LQG controller demonstrating the practicality for this state estimation approach in control.

The robustness of the REKF was shown by an RMSE of less than 1 degree despite the imperfect, nonlinear dynamic equations. This approach proposes a more robust solution than similar state estimation filters used in soft robotics. The positive definite REKF equations ensure that the solution will not diverge from the desired trajectory.

Overall, the work performed in this Thesis has produced a robust, portable, and accurate REKF based sensor fusion approach for state estimation. This system increases soft robot proprioception enabling the robots use in environments where human-robot interaction is beneficial. This research takes a step towards fully controllable soft robots assisting people with dangerous, rehabilitative, or assistive tasks.

5.1 Future Work

While the results are promising, there are improvements that are left to future work. First of all the soft robot model shows error on deflation and with initial inputs.

Further augmentation or filtering would improve the sensor fusion results. Exploring more accurate modeling techniques that don't require a large amount of calculation or simulation time should be considered. Improvements could be made to the REKF as well. The REKF focuses on stability and may lack optimality. Thus experimenting with iterative or adaptive alteration could improve the accuracy of the filter.

The transferability of the sensing system shows promise for estimating the state of most if not all soft robot manipulators so long as the IMU sensor and wire draw sensor can be attached to the robot. One extension to this project would require testing this system on several robots and comparing the results. Such a research project would mostly consist of exploring the necessary changes required to update the dynamic models for each new arm. This work also sets a foundation for adding additional joints and exploring non-planar motion, because the sensors are capable of measuring bending in all directions with the addition of two more wire draw sensors. Improvements to the control system should also be considered, or rather incorporating the sensor fusion algorithms with a proven soft robot controller for further proof of soft robot use in any environment.

REFERENCES

Al-Ibadi, A., S. Nefti-Meziani and S. Davis, "Design, kinematics and controlling a novel soft robot arm with parallel motion", Robotics **7**, 2, URL `https://www.mdpi.com/2218-6581/7/2/19` (2018a).

Al-Ibadi, A., S. Nefti-Meziani, S. Davis and T. Theodoridis, "Novel design and position control strategy of a soft robot arm", Robotics **7**, 4, URL `https://www.mdpi.com/2218-6581/7/4/72` (2018b).

Armanini, C., C. Messer, A. T. Mathew, F. Boyer, C. Duriez and F. Renda, "Soft robots modeling: a literature unwinding", CoRR **abs/2112.03645**, URL `https://arxiv.org/abs/2112.03645` (2021).

Bern, J. M., Y. Schnider, P. Banzet, N. Kumar and S. Coros, "Soft robot control with a learned differentiable model", in "2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)", pp. 417–423 (2020).

Best, C. M., J. P. Wilson and M. D. Killpack, "Control of a pneumatically actuated, fully inflatable, fabric-based, humanoid robot", in "2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)", pp. 1133–1140 (2015).

Bicchi, A., G. Tonietti and E. Piaggio, "Design, realization and control of soft robot arms for intrinsically safe interaction with humans", in "Proc. IARP/RAS Workshop on Technical Challenges for Dependable Robots in Human Environments", pp. 79–87 (2002).

Chawla, A., C. Frazelle and I. Walker, "A comparison of constant curvature forward kinematics for multisection continuum manipulators", in "2018 Second IEEE International Conference on Robotic Computing (IRC)", pp. 217–223 (2018).

Della Santina, C., 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 **5**, 2, 1001–1008 (2020).

Della Santina, C., R. K. Katzschmann, A. Biechi and D. Rus, "Dynamic control of soft robots interacting with the environment", in "2018 IEEE International Conference on Soft Robotics (RoboSoft)", pp. 46–53 (2018).

Duriez, C., "Control of elastic soft robots based on real-time finite element method", in "2013 IEEE International Conference on Robotics and Automation", pp. 3982–3987 (2013).

Einicke, G. and L. White, "Robust extended kalman filtering", IEEE Transactions on Signal Processing **47**, 9, 2596–2599 (1999).

Grissom, M. D., V. Chitrakaran, D. Dienno, M. Csencits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn and I. Walker, “Design and experimental testing of the OctArm soft robot manipulator”, in “Unmanned Systems Technology VIII”, edited by G. R. Gerhart, C. M. Shoemaker and D. W. Gage, vol. 6230, pp. 491 – 500, International Society for Optics and Photonics (SPIE, 2006), URL https://doi.org/10.1117/12.665321.

Hyatt, P., D. Kraus, V. Sherrod, L. Rupert, N. Day and M. D. Killpack, “Configuration estimation for accurate position control of large-scale soft robots”, IEEE/ASME Transactions on Mechatronics **24**, 1, 88–99 (2019).

Kim, D., S.-H. Kim, T. Kim, B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kown, H. Lee, J. Bae, Y.-L. Park, K.-J. Cho and S. Jo, “Review of machine learning methods in soft robotics”, PLoS ONE 16(2): e0246102 URL https://doi.org/10.1371/journal.pone.0246102 (2021a).

Kim, D., M. Park and Y.-L. Park, “Probabilistic modeling and bayesian filtering for improved state estimation for soft robots”, IEEE Transactions on Robotics **37**, 5, 1728–1741 (2021b).

Laschi, C., “Soft Robotics: from scientific challenges to technological applications”, in “Micro- and Nanotechnology Sensors, Systems, and Applications VIII”, edited by T. George, A. K. Dutta and M. S. Islam, vol. 9836, pp. 383 – 390, International Society for Optics and Photonics (SPIE, 2016), URL https://doi.org/10.1117/12.2223873.

Lee, C., M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim and S. Kim, “Soft robot review”, International Journal of Control, Automation and Systems **15**, URL https://doi.org/10.1007/s12555-016-0462-3 (2017).

Loo, J. Y., K. C. Kong, C. P. Tan and S. G. Nurzaman, “Non-linear system identification and state estimation in a pneumatic based soft continuum robot”, in “2019 IEEE Conference on Control Technology and Applications (CCTA)”, pp. 39–46 (2019a).

Loo, J. Y., C. P. Tan and S. G. Nurzaman, “H-infinity based extended kalman filter for state estimation in highly non-linear soft robotic system”, in “2019 American Control Conference (ACC)”, pp. 5154–5160 (2019b).

Lunni, D., G. Giordano, E. Sinibaldi, M. Cianchetti and B. Mazzolai, “Shape estimation based on kalman filtering: Towards fully soft proprioception”, in “2018 IEEE International Conference on Soft Robotics (RoboSoft)”, pp. 541–546 (2018).

Navarro, S. E., S. Nagels, H. Alagi, L.-M. Faller, O. Goury, T. Morales-Bieze, H. Zangl, B. Hein, R. Ramakers, W. Deferme, G. Zheng and C. Duriez, “A model-based sensor fusion approach for force and shape estimation in soft robotics”, IEEE Robotics and Automation Letters **5**, 4, 5621–5628 (2020).

Nguyen, P. H., I. B. Imran Mohd, C. Sparks, F. L. Arellano, W. Zhang and P. Polygerinos, "Fabric soft poly-limbs for physical assistance of daily living tasks", in "2019 International Conference on Robotics and Automation (ICRA)", pp. 8429–8435 (2019).

Rao, P., Q. Peyron, S. Lilge and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance", Frontiers in Robotics and AI **7**, URL https://www.frontiersin.org/article/10.3389/frobt.2020.630245 (2021).

Renda, F., M. Giorelli, M. Calisti, M. Cianchetti and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables", IEEE Transactions on Robotics **30**, 5, 1109–1122 (2014).

Robert J. Webster, I. and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review", The International Journal of Robotics Research **29**, 13, 1661–1683, URL https://doi.org/10.1177/0278364910368147 (2010).

Rupert, L., T. Duggan and M. D. Killpack, "Improved continuum joint configuration estimation using a linear combination of length measurements and optimization of sensor placement", in "Front Robot AI", vol. 8 (2021), URL https://doi.org/10.3389/frobt.2021.637301.

Scharff, R. B. N., G. Fang, Y. Tian, J. Wu, J. M. P. Geraedts and C. C. Wang, "Sensing and reconstruction of 3-d deformation on pneumatic soft robots", IEEE/ASME Transactions on Mechatronics **26**, 4, 1877–1885 (2021).

Troise, M., M. Gaidano, P. Palmieri and S. Mauro, "Preliminary analysis of a lightweight and deployable soft robot for space applications", Applied Sciences **11**, 6, URL https://www.mdpi.com/2076-3417/11/6/2558 (2021).

Vasic, M. and A. Billard, "Safety issues in human-robot interactions", in "2013 IEEE International Conference on Robotics and Automation", pp. 197–204 (2013).

Wang, H., B. Yang, Y. Liu, W. Chen, X. Liang and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller", IEEE/ASME Transactions on Mechatronics **22**, 1, 41–50 (2017).

Wang, J. and A. Chortos, "Control strategies for soft robot systems", Advanced Intelligent Systems **n/a**, n/a, 2100165, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202100165 (2021).

Wang, X., Y. Li and K.-W. Kwok, "A survey for machine learning-based control of continuum robots", Frontiers in Robotics and AI **8**, URL https://www.frontiersin.org/article/10.3389/frobt.2021.730330 (2021).

APPENDIX A

EKF FIGURE

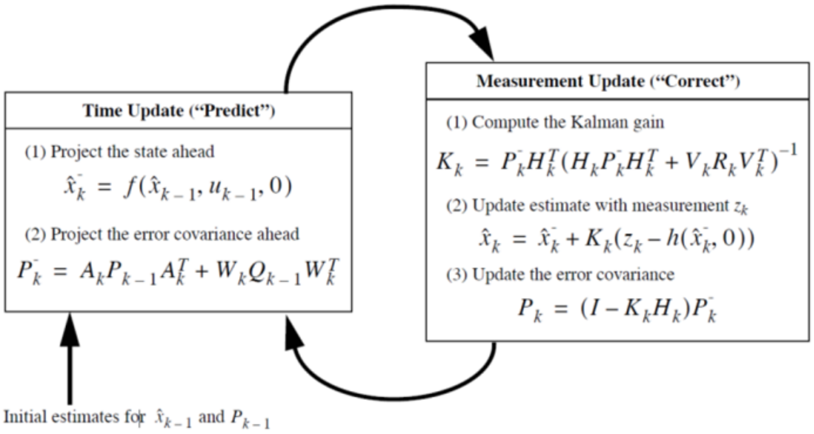

Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$
(2) Project the error covariance ahead
$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$
(3) Update the error covariance
$P_k = (I - K_k H_k) P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure A.1: Diagram Depicting the Flow of the EKF

Algorithm 1 EKF

```
**Initialize:**
$x_{k-1} \leftarrow x_0$
$P_{k-1} \leftarrow P_0$
**Predict:**
$\hat{x}_k^- \leftarrow f(x_{k-1})$
$P_k^- \leftarrow A_{k-1} P_{k-1} A_{k-1}^T + Q$
where: $\bar{R}_t = R + D_t Q D_t^T$
**Correct:**
$L_k \leftarrow P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
$\hat{x}_k \leftarrow \hat{x}_k^- + L_k(z_k - h(\hat{x}_k^-))$
$P_k \leftarrow (I - K_k H_k) P_k^-$
```

Algorithm 2 REKF

```
Initialize:
$x_{k-1} \leftarrow x_0$
$P_{k-1} \leftarrow P_0$
Predict:
$\hat{x}_k^- \leftarrow f(x_{k-1}) + B_t Q D_t^T \bar{R_t}^{-1}(z - H_k f(x_{k-1}))$
$P_k^- \leftarrow (A_{k-1} - B_t Q D_t^T \bar{R_t}^- 1H)P(A_{k-1}$
$- B_t Q D_t^T \bar{R_t}^- 1H)^T + B(Q - Q D_t^T R_t D_t Q)B^T$
where: $\bar{R}_t = R + D_t Q D_t^T$
Correct:
$L_k \leftarrow P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
$\hat{x}_k \leftarrow \hat{x}_k^- + L_k(z_k - h(\hat{x}_k^-))$
$P_k \leftarrow (I - K_k H_k) P_k^-$
```

Algorithm 3 UKF

```
Predict:
$\hat{x}_k^- \leftarrow \sum W_j^a x_j$
$P_k^- \leftarrow \sum W_j^c (x_j - \hat{x}_k^-)(x_j - \hat{x}_k^-)^T$
Correct:
$\hat{z} \leftarrow \sum W_j^a z_j$
$\hat{S}_k \leftarrow \sum W_j^c (s_j - \hat{x}_k^-)(z_j - \hat{z})^T$
$C_{sz} \leftarrow \sum W_j^c (s_j - \hat{x}_k^-)(z_j - \hat{z})^T$
$L_k \leftarrow C_{sz} \hat{S}_k^- 1$
$\hat{x}_k \leftarrow \hat{x}_k^- + L_k(z_k - \hat{z})$
$P_k \leftarrow P_k^- - K_k \hat{S}_k K_k^T$
```

APPENDIX B

EQUATION OF MOTION MATRIX DERIVATION

The following equations show the M, C, G matrices respectively along with the parameters used in table B.1.

$$M = \frac{1}{4}I_{zz} + \frac{1}{2}m(\cos(\frac{q}{2})(b - L/q)) + (L\sin(\frac{q}{2})/q^2)^2 + \frac{1}{4}(m0\sin(\frac{q}{2})^2(b - L/q)^2) \tag{B.1}$$

$$C = -\frac{1}{2q^5}(mL\dot{q}(2\sin(\frac{q}{2}) - q\cos(\frac{q}{2}))(2L\sin(\frac{q}{2}) - Lq\cos(\frac{q}{2}) + bq^2 cos(\frac{q}{2}))) \tag{B.2}$$

$$G = -\frac{1}{2q^2}(mg(L\sin(q) + bq^2\cos(q) - Lq\cos(q))) \tag{B.3}$$

Table B.1: Parameters Used in the Derivation of the Equations of Motion.

m	0.35 /(kg/)
g	9.8 m/s^2
L	0.17 /(m/)
I_{zz}	0.1 kg/m^2
k_0	.4897
d_0	.8616
α_0	1.2634

APPENDIX C

UNSCENTED KALMAN FILTER

The formulation of the Unscented Kalman Filter (UKF) is given here. To begin, a certain number of weights and sigma points are necessary to more accurately predict the current state. The number of weights and sigma points is determined by the equation:

$$N = 2n + 1 \tag{C.1}$$

with n equal to the number of dimensions of the system, in this case 2. The weights are now determined by

$$W_0 = \frac{1}{n + \lambda} = 0.4 \tag{C.2}$$

$$W_i = \frac{1}{2(n + \lambda)} \text{ for } i = 1, ..., 2n \tag{C.3}$$

The sum of the weights is equal to one. In the formulation of the filter two weights could be used, W^a for state predictions and W^c for covariance predictions. In this case both W weights are the same. Next the sigma points are found from the following:

$$\sigma_0 = x_k - 1 \tag{C.4}$$

$$\sigma_1 = x_k - 1 + \sqrt{\frac{n}{1 - W_0} * P_{k-1,j}}, j = 1 \tag{C.5}$$

$$\sigma_2 = x_k - 1 + \sqrt{\frac{n}{1 - W_0} * P_{k-1,j}}, j = 2 \tag{C.6}$$

$$\sigma_3 = x_k - 1 - \sqrt{\frac{n}{1 - W_0} * P_{k-1,j}}, j = 1 \tag{C.7}$$

$$\sigma_4 = x_k - 1 - \sqrt{\frac{n}{1 - W_0} * P_{k-1,j}}, j = 2 \tag{C.8}$$

where j is the column of P.

With the weights and sigma points established the prediction equations are given as the following sums

$$\hat{x}_k^- = \sum_{j=0}^{N} W_j x_j \tag{C.9}$$

$$P_k^- = \sum_{j=0}^{N} W_j (x_j - \hat{x_k^-})(x_j - \hat{x_k^-})^T \tag{C.10}$$

$$\hat{z} = \sum_{j}^{N} z_j \tag{C.11}$$

with

$$z_j = h(\sigma_j) \text{ for } j = 1, ..., N \tag{C.12}$$

The updates are then given by the equations

$$\hat{S}_k = \sum_{j=0}^{N} (z_j - \hat{z}_k^-)(z_j - \hat{z}_k^-)^T \tag{C.13}$$

$$C_{sz} = \sum_{j=0}^{N} (\sigma_j - \hat{x}_k^-)(z_j - \hat{z}_k^-)^T \tag{C.14}$$

$$L_k = C_{sz} \hat{S}_k^{-1} \tag{C.15}$$

$$\hat{x}_k = \hat{x}_k^- + L_k(z_k - \hat{z}) \tag{C.16}$$

$$P_k = P_k^- - L_k^T \hat{S}_k L_k^T \tag{C.17}$$

with $\hat{S}$ and C_{sz} are covariance sums and the remaining variables remain the same as discussed in the REKF algorithms.