Development of an Automation Platform for Fabric Manipulation and Assembly

by

Calum Briggs

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Robotics Engineering by

August 2020

APPROVED:

Professor Marko Popovic, Advisor

Professor Michael Gennert, Committee Member

Professor Berk Calli, Committee Member

Abstract

Modern robotic automation technology has transformed manufacturing systems across many sectors. However, in the field of fabric assembly, which is one of the most labor-intensive industries in the world, there has been very little increase in automation over the past several decades. There are several challenges to contend with when automating fabric assembly tasks including the compliant material nature of fabric, the high frequency of task changes, and the feedback and adjustment required to successfully complete the tasks. A robotic fabric manipulation and assembly system is created based around a Yaskawa SDA10F industrial robot. Computer vision capabilities are integrated into the system to measure the process adjustments that are necessary when performing fabric assembly tasks. The system is designed to account for variability in material placement and composition and to be easily re-configurable to perform new tasks. The system's capabilities are evaluated by having it perform a proof of concept fabric assembly task with varied input material. Because the SDA10F robot has a high input command latency, a custom three degree of freedom wrist was constructed to perform corrections in real-time by adjusting the position and velocity of the end-effector. The accuracy of the wrist was independently tested to evaluate how it could improve the total system's functionality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The development of manufacturing automation has changed the nature of work for nearly every person on earth. Employment in direct industrial production has been decreasing, being replaced by jobs in service, education, and healthcare. While automation can displace workers and create short term readjustment problems in some communities, overall it creates more and better jobs, increases access to education, and raises standards of living [1].

Manufacturing automation has not affected all industries in the same way, the primary beneficiaries of this technology have so far been those manufacturing processes which require a high degree of power and precision. Processes such as chemical synthesis, metal and polymer casting, and the assembly of rigid components have been performed primarily by machines for several decades. Classic industrial automation involves creating a manufacturing line with custom machines that perform repetitive tasks at a high degree of precision, allowing for the creation of a high volume of near-identical products with a low labor cost.

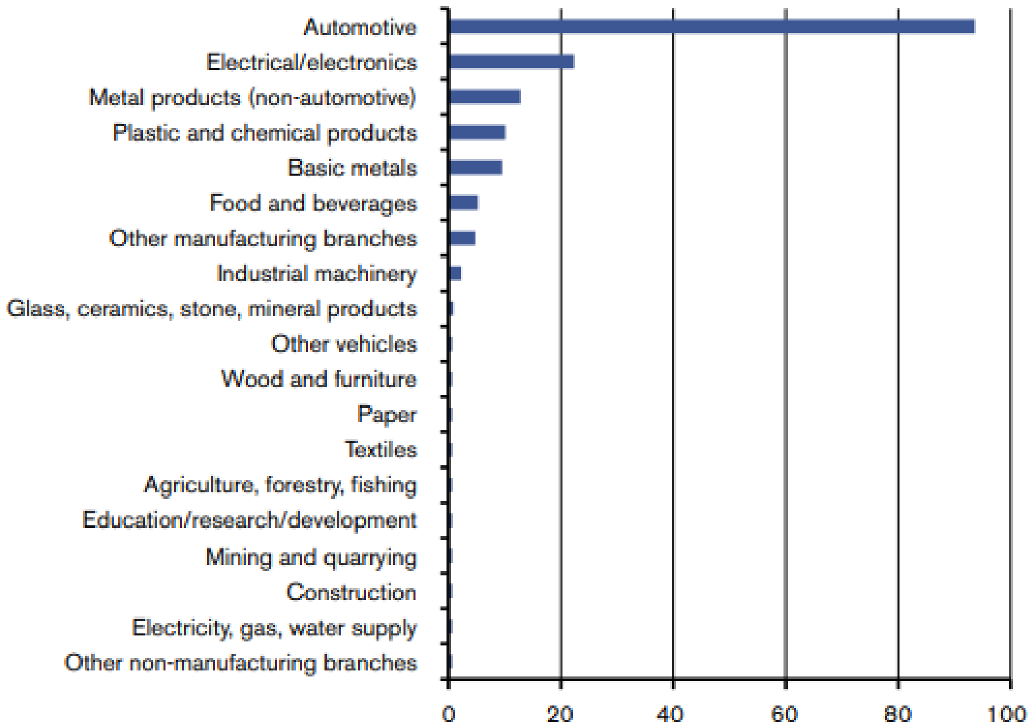

Figure 1.1: Robot Use by Sector, per 1000 Employees. [3]

Robotic automation processes are somewhat different. Instead of a custom machine to complete each process, general-purpose robotic manipulators are used, which are programmed to complete specific tasks. Industrial robotics has expanded the capabilities of manufacturing automation into high value and complex assembly operations, such as automotive and electrical manufacturing, which are today highly automated industries. One of the biggest manufacturing sectors which have seen little increase in automation in recent years is the textile and clothing industry. The manufacture of clothing and textiles makes up 6% of the worlds' exports and is worth more than $300 billion each year [2]. Despite the size and labor-intense nature of textile and clothing production, data from the World Trade organization indicates that the level of robotic automation in the sector is minimal. [3]

The demand for textile products across the world, especially in large developing countries such as India and China, has been expanding rapidly, is expected to increase considerably in the coming years due to rising incomes [4]. The increase in global demand for textile and clothing products has not been matched by an increase in productivity due to automation. Consequently, textile manufacturing has been expanding rapidly in areas with very low labor costs, especially in Southeast Asia. The increasing costs of labor in many of these countries further expands the need for production efficiency in textile manufacturing. Without the integration of automation into these production processes, it will become increasingly challenging to fulfill the demand for these products.

The least automated part of the textile manufacturing process is sewing assembly. Nearly all garment and fabric assembly tasks are performing manually using sewing machines. There is a limited amount of automation using programmable pattern stitchers, which can automatically stitch patterns into the fabric using string or to affix a piece of fabric to one placed on top of it. These processes are primarily decorative, however, and all sewing not done on a flat surface is performed manually. There are several barriers to increased automation in sewing assembly processes, which this paper explores.

1.2 Barriers to Automation

Apparel and other fabric assembly processes may involve between a dozen and several hundred steps to turn sheets of fabric, leather, and thread into final products. Some of these steps are very complex and require a degree of dexterity and control unavailable to modern robotic manipulators. The most complex assembly tasks involve sewing together two pieces of fabric that have been manually deformed to create a structural stitch which provides the shape of a garment. These operations require continual realignment and tension adjustment on the two pieces of fabric so that the final stitch line is exactly the right shape. Automating these tasks would likely require significant research and development on custom technology for a task that can be done quite efficiently with moderately skilled labor. An easier target for automation is those tasks that require less dexterity, only simple handling, and positioning of fabrics. There are many such tasks in modern garment assembly plants, which usually require moving material from one stage to another, arranging material on a flat surface to be processed by a machine, or affixing pieces together in a simple manner. Learning how the simpler tasks in the textile assembly process will start to build the capabilities necessary to automate the more complex ones. There are however barriers that have prevented almost any automation in the textile assembly process and which must be overcome to begin to automate even the simplest production tasks.

1.2.1 Production Scale

Replacing manual processes with automated robotic systems requires a significant upfront investment to maintain consistency in production throughput. Because garment manufacturing generally occurs in areas with very low costs of labor, there is little incentive to replace human workers with robotic automation unless it can be made very economical. The large production scale also poses a problem because of the "assembly line" nature of manufacturing. If a malfunction causes any single step in the process to be interrupted the whole manufacturing process is now bottle-necked. The tools of the fabric assembly system are designed to be robust to malfunction. Human workers and sewing machines can be easily swapped in and out to perform a wide variety of tasks as production requires. A viable automation system should be both robust to error and be possible to circumvent in the event of a malfunction. Automation systems must be developed with economy and ease of integration in mind. Optimally automation systems should make use of as much existing and low-cost hardware as possible.

1.2.2 Scope of Variation of Tasks

While each step in most fabric assembly processes is relatively simple, no two are the same. Each task requires slightly different tooling and motion. A typical fabric assembly factor may produce several products and modifications of products. This is especially true in garment assembly, where factories manufacture many different sizes and models with production shifting daily. Human operators can easily switch between different tasks but taking time reprogramming a robot delays production. A viable automation solution must utilize hardware able to perform many tasks and must be run on software that is quickly re-configurable to perform new operations.

1.2.3 Soft and Variable Material

Perhaps the most difficult barrier to overcome when trying to automate tasks in fabric assembly is dealing with the soft and variable nature of the material involved in the processes. For manufacturing involving rigid objects, once a robot has grasped a component, all of the motions of that component can be calculated for future manipulation. This makes task planning and precise actuation very straightforward. This is not the case for soft material, and especially for fabric which will move dynamically in a manner that is complex to model. There has been a large amount of research in the field of modeling soft materials to try and address this problem but is it far from resolved [5]. Furthermore, materials used in garment manufacturing, especially biologically derived products such as wool, cotton, and leather, may exhibit significant variation in physical characteristics between samples. A fabric assembly system must be able to account for the dynamic nature of the material and be robust to any material variability that may exist.

Chapter 2

Related Work

Much of the recent academic literature regarding fabric manipulation and assembly focuses on techniques for sensing and managing position and configuration in a relatively unknown environment. [5] Additionally, a lot of research focuses on exploring a specific task such as smoothing, folding, or stacking. There is also significant research into modeling the deformations of fabric and techniques and into techniques for smoothing out wrinkles in fabric to reduce the complexity of its configuration. This research is often not directly relevant to automation engineering, which deals in fixed inputs and outputs, and optimizes for speed, economy, flexibility, and high success rate.

There are also several recent commercial ventures which have advertised themselves as being textile assemble automation solutions. Evaluating these is not straightforward as limited technical information is generally available for commercial products and the company's claims cannot be taken at face value. This section discusses some of the most relevant research and commercial technology aimed at addressing the challenges in textile manufacturing.

2.1 Early Research

The first significant research into the automation of garment manufacturing was performed during the 1980s. During a time marked by an increase in automation in other sectors such as automotive, apparel manufacturers sought ways to decrease the labor volume in their operations. Automation issues that are presented by both the physical nature of the fabric as well as the scope and flexibility requirements of the manufacturing process were evaluated and discussed. This research was also motivated by the fact that at this time, manufacturing was moving to areas with lower costs of labor and was sponsored by both manufacturing groups and trade organizations such as NATO [7]. This early research led to the development of many robotic capabilities useful for fabric manipulation, such as stack separation [8], visual analysis [9], and tactile sensing [10].

A report by the American Apparel Manufacturers Association from 1987 [11] lauded the great improvement in "hard automation" or non-robotic labor-saving tools such as thread spinning and fabric weaving and was optimistic about the entrance of the newer robotic automation into the field. The report describes several advancements in robotic capabilities that would be necessary before there would be a large scale integration of robotics into the apparel manufacturing market. Today, all of the advancements listed in this report have been achieved, including the development of real-time vision processing, robust tactile sensing, increased robotic mobility, and general-purpose end effectors. Nevertheless, few of these developments ever became a part of commercial apparel manufacturing processes. The reason for this, in part, is that while narrow solutions for individual manufacturing applications can be developed, It is much more difficult to combine them into a feasible total manufacturing solution.

An analysis by Henry A. Seesselberg, Director of Advanced Technology operations at the Fashion Institute of Technology in New York City in 1990, concluded that any device which could account for the variability in shape and material properties as well as have the process flexibility required to begin to replace humans in the garment assembly process would be infeasible using traditional automation approaches [11]. History would seem to support Seesselberg's analysis, although there have been some commercial advances in garment manufacture, recent increases in efficiency have come primarily from ever-expanding globalized supply chains and increased access to low-cost labor markets. Access to sufficient low-cost labor has reduced the economic incentives to create automation solutions have also decreased significantly. However, due to the global increase in both cost of labor and demand for textile products, especially garments, this field has had a recent increase in research interest.

2.2 Softwear automation

One of the most ambitious commercial attempts to create a garment assembly automation solution is being performed by the company Softwear Automation.[13] The Softwear Automation solution consists of a large flat surface with small balls mounted on the surface which can spin in different directions, moving fabric placed on the surface. The technology is designed to be modular, with sewing machines and other processes devices distributed along the bed which performs operations on fabric in a fashion similar to an assembly line. This approach addresses some of the barriers to fabric assembly by circumventing the problem of soft material manipulation. On their table, the orientation of the fabric can be easily manipulated and it is possible to correct for folds and creases. Automatic rollers can be employed

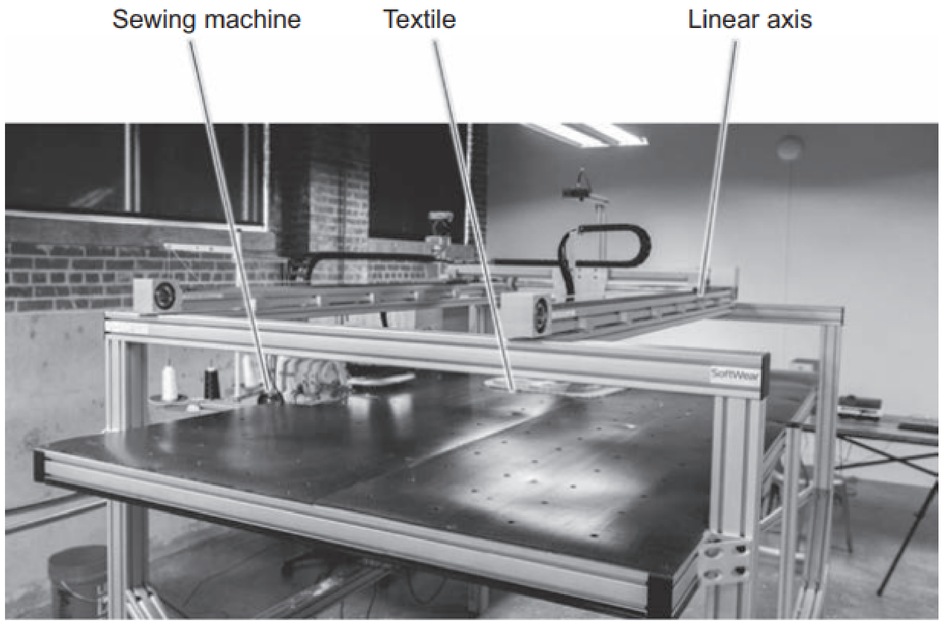

Figure 2.1: Softwear Automation Structure [12]

to create simple folds, and different pieces can be positioned relative to each other in order to create stitches. vacuum picking modules are used to re-position fabric more exactly than the rollers are capable.

Softwear Automation has successfully completed several basic sewing tasks using this system but it has plans for more ambitious operations such as full shirt and mattresses. However, they are yet to demonstrate assembly operations that surpass the capabilities of programmable stitching machines, which have been standard in the industry for more than thirty years. This technology may be able to replace some operations, such as decorative stitching and shoe upper preparation, which consist of many flat layers of fabric being stacked on top of one another and sewn together in 2D, but they have not demonstrated thus far that the technology could match the speed and efficiency of human operators. This system also represents a significant investment on the part of an assembly plant, as all of the production hardware integrated into the table and the entire production process must be based around the capabilities of the system.

2.3 SewBo

In 2016, the company SewBo developed a system that can fully assemble a shirt out of fabric pieces using a standard industrial manipulator, a sewing machine, and a vacuum pickup module. It was able to do this because, before the process, each of the pieces of fabric had been treated with a starch solution which made the fabric rigid and thus easily manipulable. This does not represent a major technological advancement in automation. As discussed earlier, it is very easy to manipulate rigid objects to perform simple motions such as sewing. There is also little information on the additional time it takes to starch and unstarch sewn components. The Sewbo robot does however present a more flexible and economical system than that put forward by softwear automation. Because the Sewbo system has few custom hardware components, it would be much easier to begin to integrate it into existing manufacturing systems if the technology proves effective for automating additional tasks.

2.4 Soft Matter Manipulation Research

Much of the recent work relevant to textile assembly automation focuses on soft material manipulation, which would be a vital component of a robust textile automation platform. Soft matter manipulation is a form of indirect position control, which is a control approach that for example would calculate how manipulating one part of a soft material held by an end effector can be used to control a different part of the same material. A system that is designed to deal with frequently changing shapes and materials must be able to determine optimal gripping points and control

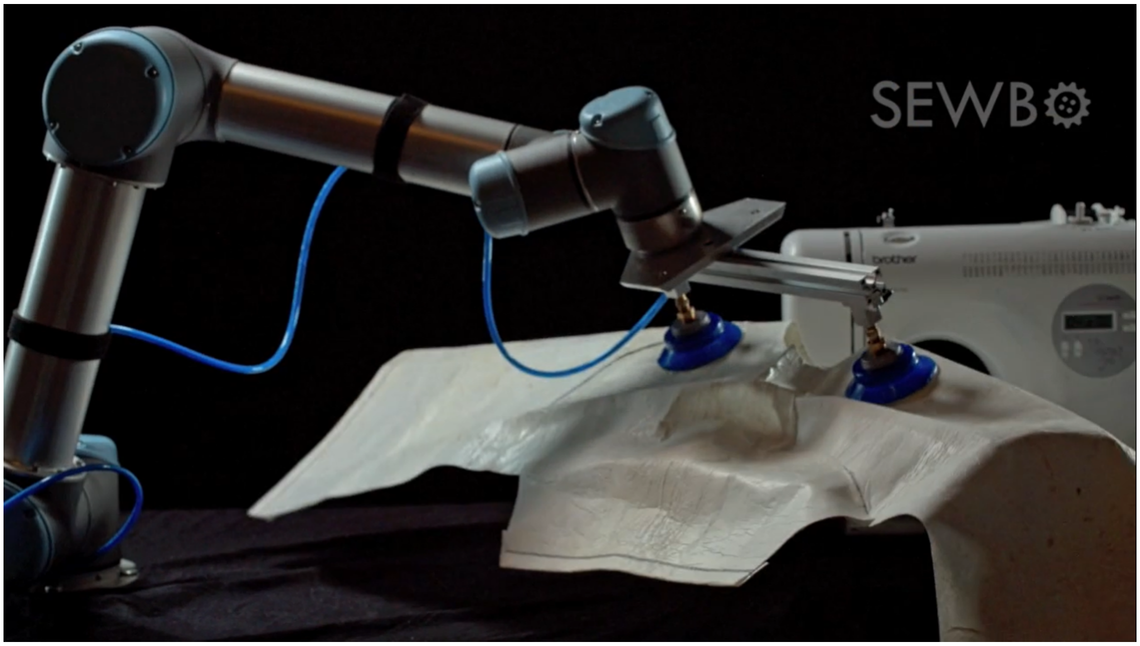

Figure 2.2: Sewbo System performing a sewing operation on starched fabric [14]

policies for performing a specific task.

Concept of Inverse position control on fabric illustrating how forces may be mapped to intended motion of particular part of fabric. [15]

In order to perform accurate inverse position control, one needs to generate a physical model of the material. There are several methods to do this, including modeling a sheet of soft material as a grid of masses and spring [16]. These include using a linearized controller [15] Jacobian estimation based on distance from the point to the end effector [17], and a PID based approach [18]. These attempts to accurately estimate, rather than to completely model the motion of soft materials may be more useful when it comes to automated manufacturing. The key adaptation these methods would need added to these methods to be useful in fabric assembly is the ability to be automatically generated from visual data. In that way, modeling software could be integrated automatically into fabric assembly systems, and would not need to be regenerated manually for different tasks.

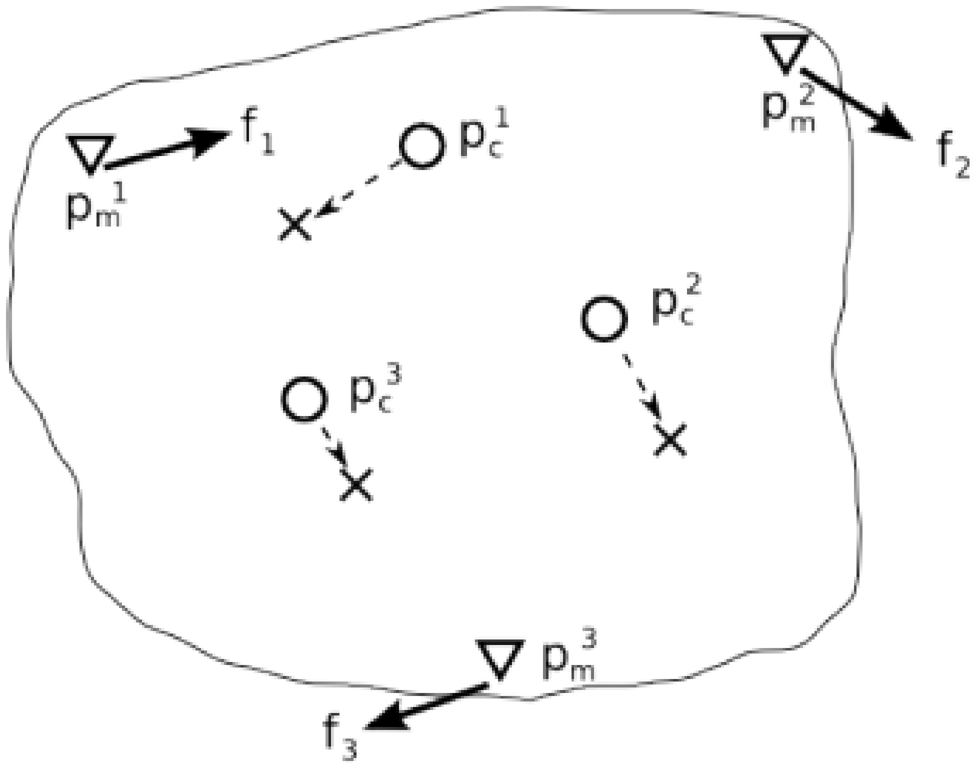

Figure 2.3: Concept of inverse position control on fabric illustrating how forces may be mapped to intended motion of particular part of fabric. [15]

Machine learning is another promising area for fabric assembly. Fabric assembly tasks are mechanically simple but difficult to manually program a robot to perform. There has been some success in teaching robots to perform fabric manipulation and assembly tasks such as using fuzzy logic to perform sewing [19] deep learning to perform bed making [20]. An issue with the machine learning approach for automation tasks is that they are very time critical, many of machine learning methods have lengthy training periods and may generate inefficient solutions . To utilize machine learning approaches for fabric assembly, they must be set up in such a way that they could accept different shapes, sizes, and materials as input parameters and still be able to perform their tasks accurately.

Chapter 3

Methodology

This section documents the design considerations and implementation of the automated alignment platform. The system is constructed using several provided pieces of hardware which serve as its primary robotic actuation components. Additional hardware and visual sensing systems are constructed in order to perform the alignment task. A ROS based communication framework is used by the system to translate visual feedback data into the required physical adjustments.

3.1 Platform Hardware

The automation platform developed in this work was built in part using hardware components provided to Popovic Labs in a partnership with New Balance Athletics. The provided hardware consists of a Yaskawa SDA10F 15 axis dual armed robot [21], two Robotiq 2F-85 two-finger grippers [22] attached to each army of the SDA10F, and an ORISOL B1510-OCS computerized stitching machine [23]. A computerized stitching machine clamps a piece of fabric to a flat surface then stitches a programmed path into it. The motors in the SDA10F are driven by a Yaskawa FS100 controller box [24].

Figure 3.1: Left to Right: Yaskawa SDA10F, Robotiq 2F-85, ORISOL ONS-1510 [22,23,24]

An Arduino Uno microprocessor [25] is wired into the stitching machine's manual pedal control system so that it can be software controlled. There is a 1280X720p ELP USB camera [26] mounted directly above and pointed at a table where the robot picks up material in order to provide visual data about the input material. There is a 3840x2160p USB camera [27] mounted to the stitching machine which is pointed down onto the surface where the stitching is performed in order to provide visual data when the material is in under the stitching machine.

3.2 Previous Contributions

This work is informed by the work of a 2017-2018 WPI Major Qualifying Project (MQP) by the author as well as Thomas Brown, Sarah O'Grady, William Sullivan, and Andrew Lewis [28]. That MQP attempted to use motion planning and tactile sensing to perform fabric handling and sewing tasks using the robot and gripper presented above and a standard sewing machine. The MQP concluded that in order to automate a sewing task, the robot performing the sewing must be able to control its motion velocity dynamically and have real-time feedback capability. These are necessary because the motion required to use a sewing machine requires continuously

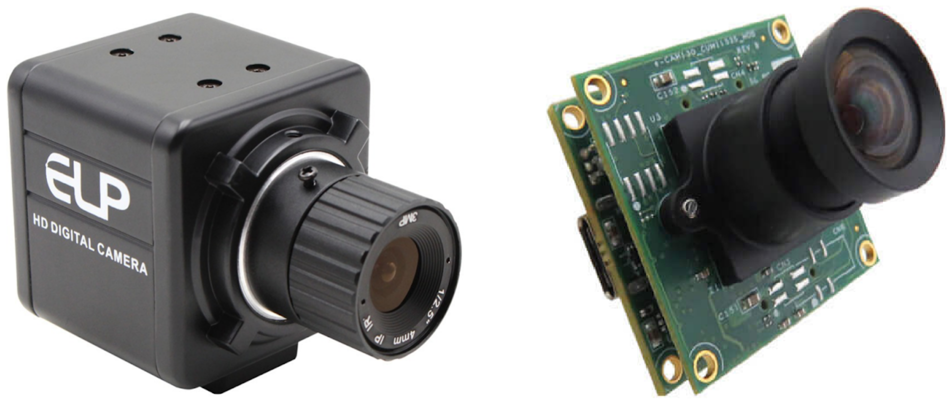

Figure 3.2: Left: ELP Camera, Right: e-CAM131 Camera [25,26]

changing speed and maintaining a particular tension on the fabric. It must also be able to account for errors caused by the variability of the material that may exist. However, the SDA10f, like many other industrial robots, is not designed to allow for dynamic velocity control or real-time feedback. Industrial robots are designed primarily to perform point to point motions to a very high degree of precision and to perform for long periods without malfunctioning. Because of this, all motion and path requests are processed by internal proprietary kinematics verification software to ensure that the motion will not damage the robot. Any new motion command will cause the robot to stop for a period of up to one second to check the validity of the new requested path before it is executed.

3.3 Software Structure

The software for this system is based on ROS [29], a communication framework for robotics that allows for simple and robust cross-platform communication. It also

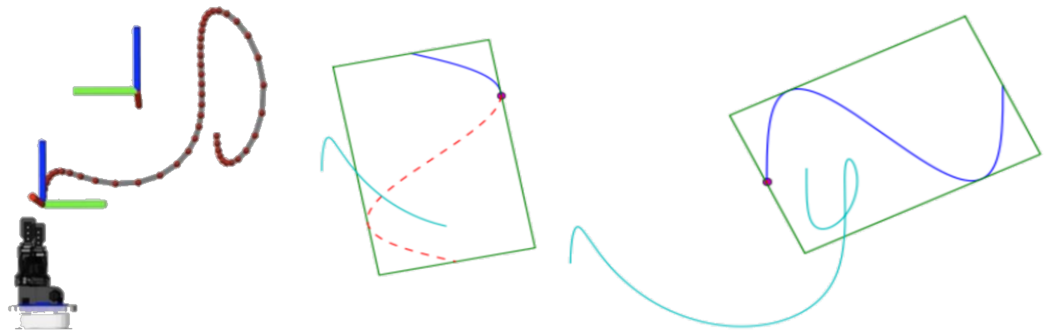

Figure 3.3: Left: Visualization of end effector path required to sew a sine wave into a piece of material, each red dot represents an equally spaced point in time. Center and Right: Simulation of the robot sewing a sine wave into a piece of fabric from a different starting position [28]

provides many software tools which are useful in robotics research. This system utilizes the MoveIt [30], ROS Industrial [31], and MotoROS [32] libraries. MoveIt and ROS-Industrial are open-source community maintained robotics motion planning libraries. MotoROS is a software package developed by Yaskawa and the Southwest Research Institute to allow Yaskawa Robots to be integrated with ROS. The MotoROS software package includes a 3D model, kinematic data, and MoveIt integration for the SDA10F.

ROS operates as a decentralized node-based structure, where nodes communicate with one another over channels using a "publisher-subscriber" model. Any node may publish data to a specific channel, and other nodes may subscribe to that channel to receive the information. There is a central "main" node that processes information from camera and robot and sends commands to the robot, gripper, and stitching machine. The data from the cameras above the stitching machine and input table are processed by individual ROS nodes. These use OpenCV [33] tools to interpret the images, transform local positional data into the robot's kinematic frame, and transmit that data to the main node. The microprocessor which controls

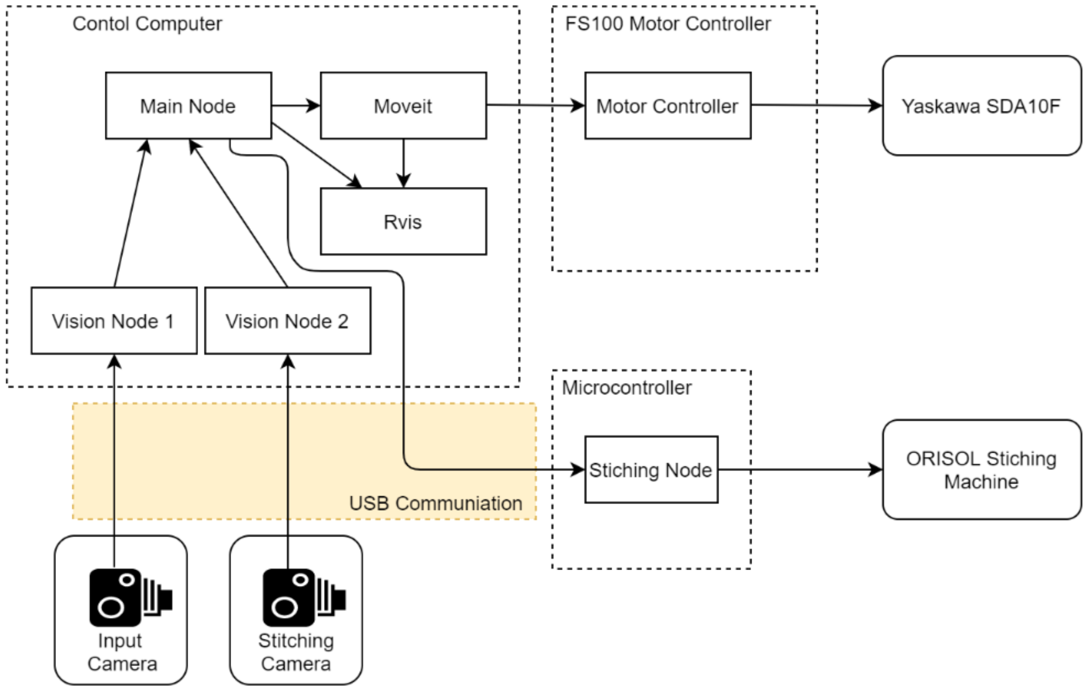

Figure 3.4: Functional diagram of the ROS software system

the stitching machine is also a ROS node and takes commands from the main node over USB using the rosserial library. The FS100 Controller which drives the robot motors is another ROS node that takes trajectory commands as inputs from the main node via the MoveIt library. Data from the main node and the robot are ready by a ROS component called rvis, which allows the user to see the current position and trajectory of the robot and monitor all activity in the ROS network.

3.4 Fabric Alignment Automation System

The primary goal of this work is to create an assembly automation platform and to use that platform to perform a proof of concept assembly task. The chosen task is to pick up a piece of flat material and move it to a processing machine, in this case, the ORISOL stitching machine, and align the material for processing. This is a common class of assembly tasks, where a worker aligns one or two pieces of material on a flat surface to be processed by a machine. Processes which fall into this class of tasks include cutting, stamping, affixing of a logo or design, and automatic stitching. The operation is relatively simple and widely applicable and therefore serves as a good starting point for examining how to begin automating garment assembly processes.

The additional design requirement for this system is that it must address the barriers and challenges to fabric assembly automation discussed earlier. The system must be able to account for deviation and error in the material composition and placement. The system must also be able to perform its task on different sizes and shapes of input material with little or no alteration to the system. Leather was chosen as the primary test material because it displays a moderate stiffness and high degree of variation in material properties between samples.

3.4.1 Fingertip Design

Custom gripping fingers affixed to the end of the Robotiq two-finger gripper were developed to optimally pick up and manipulate fabric. There are several considerations when designing the gripping attachments including ease of manipulation, passive compliance, and gripping quality.

The gripper must be able to manipulate the fabric satisfactorily inside the workspace and to perform corrective alignment once the fabric is placed in the stitching machine. The 45 degree offset in the gripper provides two key features. It allows the robot to move the bottom of the finger across a flat surface without the hand and robot obstructing its path. The offset also reduces the radius of rotation in the upper arm while rotating a grasped object. A “T-shaped” design of the finger is used so that it does not collide with the guide clamp on the stitching machine, which sits 3 cm above the surface of the fable and clamps the fabric in place after

Figure 3.5: Fabric Alignment System

the alignment is performed.

The finger must be rigid enough so that it is able to perform accurate alignment, however, there are advantages to some amount of compliance. Compliant fingers can be pressed down onto a surface to create a flush surface with the table, allowing for precise pickup and placement operations without needing to perform as precise of a motion. This simplifies complex path planning that may need to be done in real-time and adds acts as a passive form of error tolerance. Because of the grippers flat wide design, it is much more compliant vertically than horizontally. It gains the benefit of compliance in the vertical direction without being horizontally flexible which may impact alignment accuracy.

Another issue that arises when gripping fabric is that the surface of the material

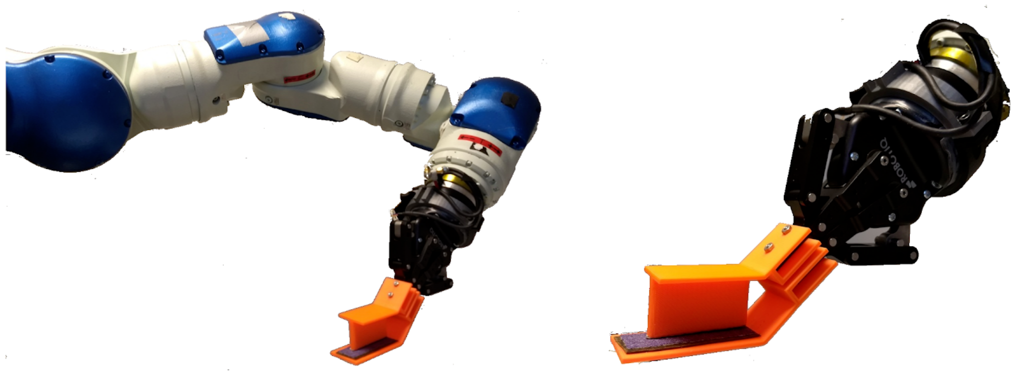

Figure 3.6: The finger attachment on the Robotiq gripper and Yaskawa robotic arm

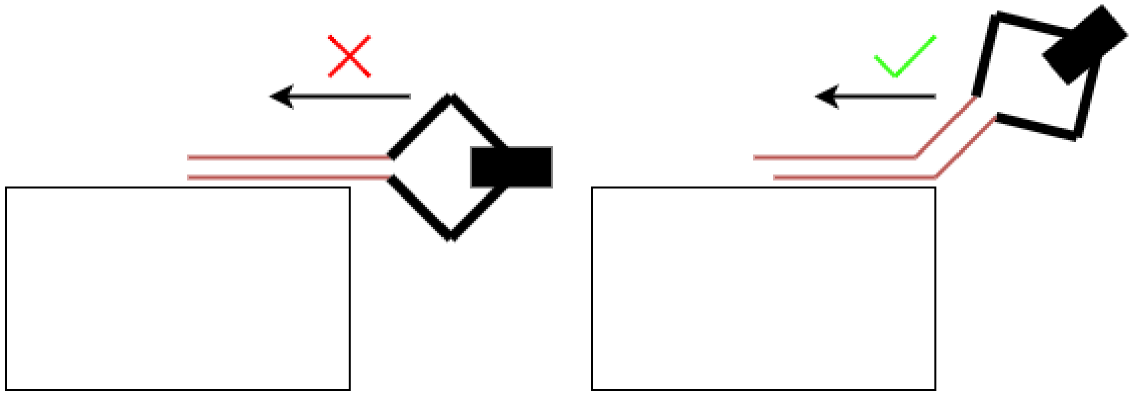

Figure 3.7: The angle in the fingers allow the hand to slide on top of surfaces without colliding with them

may be uneven, varying in thickness and surface characteristics such as friction. This can result in a grip that only makes partial contact and a loose grip. This can be exacerbated by a compliant gripper, as applying additional force to compensate for variation in grip quality can warp the fingers, leading to an even lower quality grip. The solution to this problem was to introduce a slight angle to the top finger so that the fingers are slightly closer together at the end than at the base. The difference in angle is only about 2 degrees, but when the gripper closes, it first makes contact at the end of the finger, bending as it closed until the whole finger was in contact.

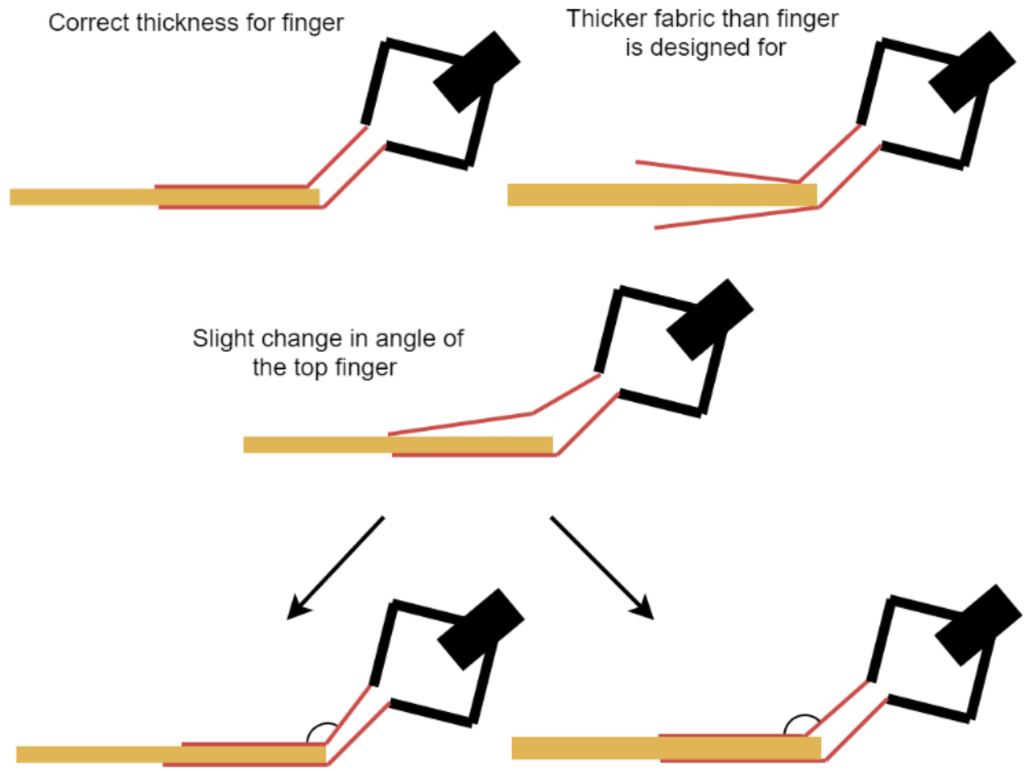

Figure 3.8: Visualization of slightly thicker that intended fabric warping a flat compliant gripper, by introducing a slight angle to the gripper, a wider variety of fabrics can be gripped.

This is not a perfect solution, with enough force, the same undesired warping will occur, but was sufficient to solve the problem for this application.

3.4.2 First Alignment Step

To begin the process, a piece of fabric with a paper pattern affixed to it is placed on the input area. The paper pattern serves as the processing target for the task. The goal of the process is to align the processing target with the processing point on the stitching machine. The input area of the system consists of a table with two raised bars onto which are placed the fabric to be processed by the system. Above the

Figure 3.9: Photo of input area with fabric in the start position. The white background and yellow bars have maximum contrast with the fabric. Shaded lamp provides diffuse light for computer vision processing

table, a camera looks directly down onto the tabletop. The camera takes a photo of the fabric input and uses OpenCV contour identification tools to determine the geometry and pose of the fabric and processing target. The robot then lifts the fabric into the air so that it naturally droops and deforms. The camera takes a picture of its new geometry to determine how much the fabric has deformed. Different samples and even the same sample run through the process multiple times will differ in how they deform once picked up.

After the robot picks up the fabric, it moves it over to the stitching machine. Using information about the original shape and deformed shape of the fabric, the robot slides the fabric under the clamp of the stitching machine. Using the data

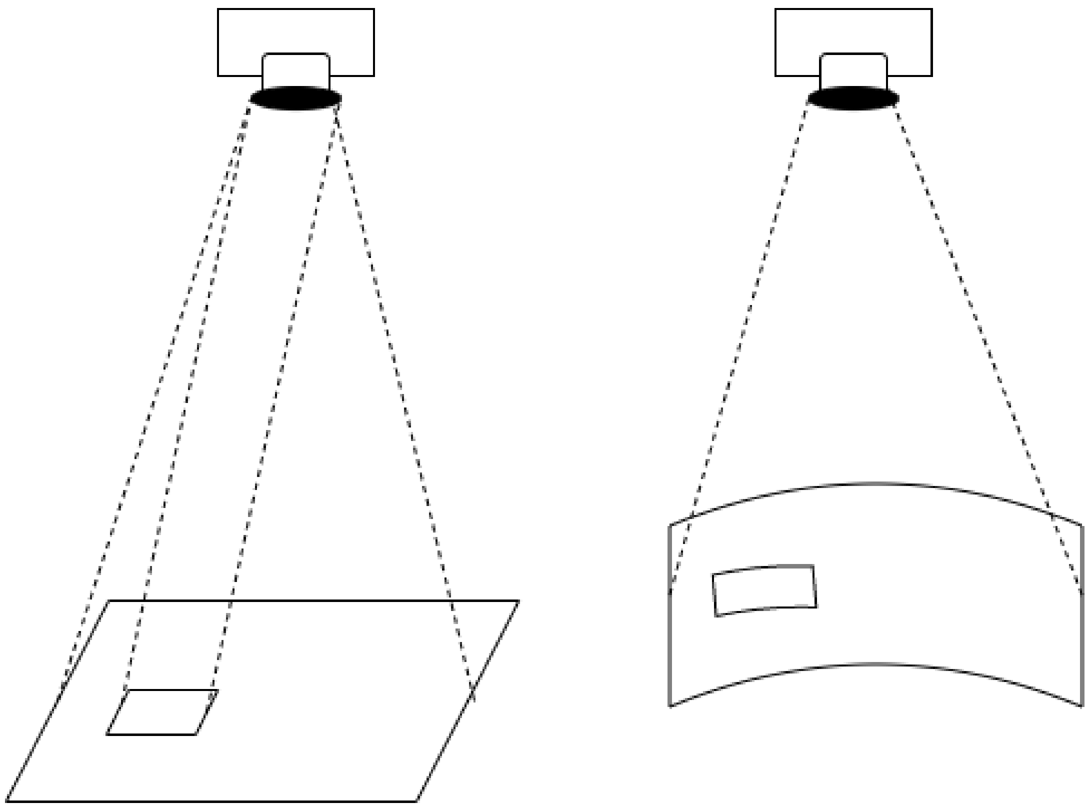

Figure 3.10: Visualization of initial computer vision algorithm. Algorithm records the geometry of the fabric and position of the target area. After the robot lifts the fabric into the air, the degree of deformation is measured.

from the pre-lift and post-lift photos, the fabric is placed so that the drooping edge makes contact with the table between the edge of the table and the edge of the clamp. The robot then moves the fabric down to be flat on the table while pulling it slightly away from the stitching machine. If the robot moves the fabric down but not backward, it can fold over itself, as is shown in figure 3.11 under error 1. If the friction between the material and the table is too high or the fabric is too soft or too long, the material may bunch up as shown in figure 3.11 under error 4. This error is sometimes correctable in the second alignment step, but if not, the process would need to be changed so that the fabric is pulled rather than pushed across the surface. This rarely occurs in leather, as it is quite stiff compared to other fabrics.

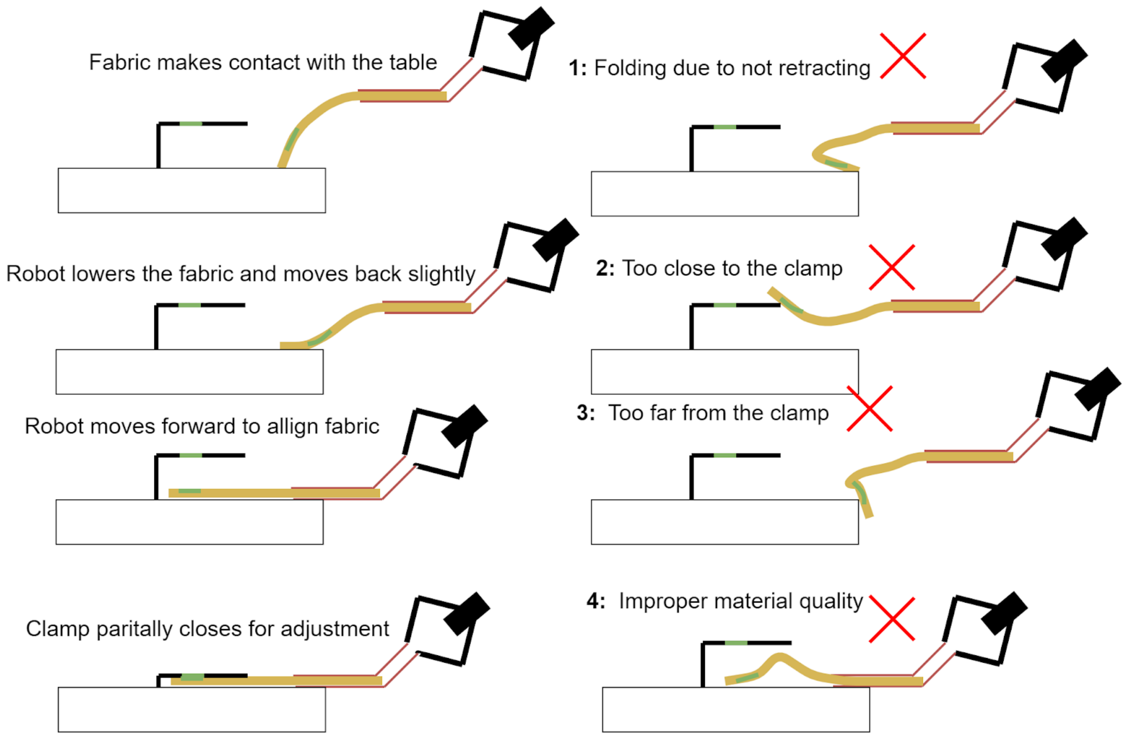

Figure 3.11: Left: Proper process of alignment, Right: Several error states that can occur

3.4.3 Second Alignment Step

The second stage of alignment occurs after the robot places the fabric underneath the clamp on the stitching machine. Once the fabric is moved under the clamp during the first alignment step, the clamp partially closes so that it can serve as guide geometry for the vision alignment program. The camera placed directly above the clamp takes a photo of the fabric and generates a model of the target area and guide geometry through the following process. The Process refers to points and distances listed in Figure 3.13

1. Convert the image to grayscale.
2. Normalize the image brightness.
3. Perform a binary threshold to change to black and white, separating the back-

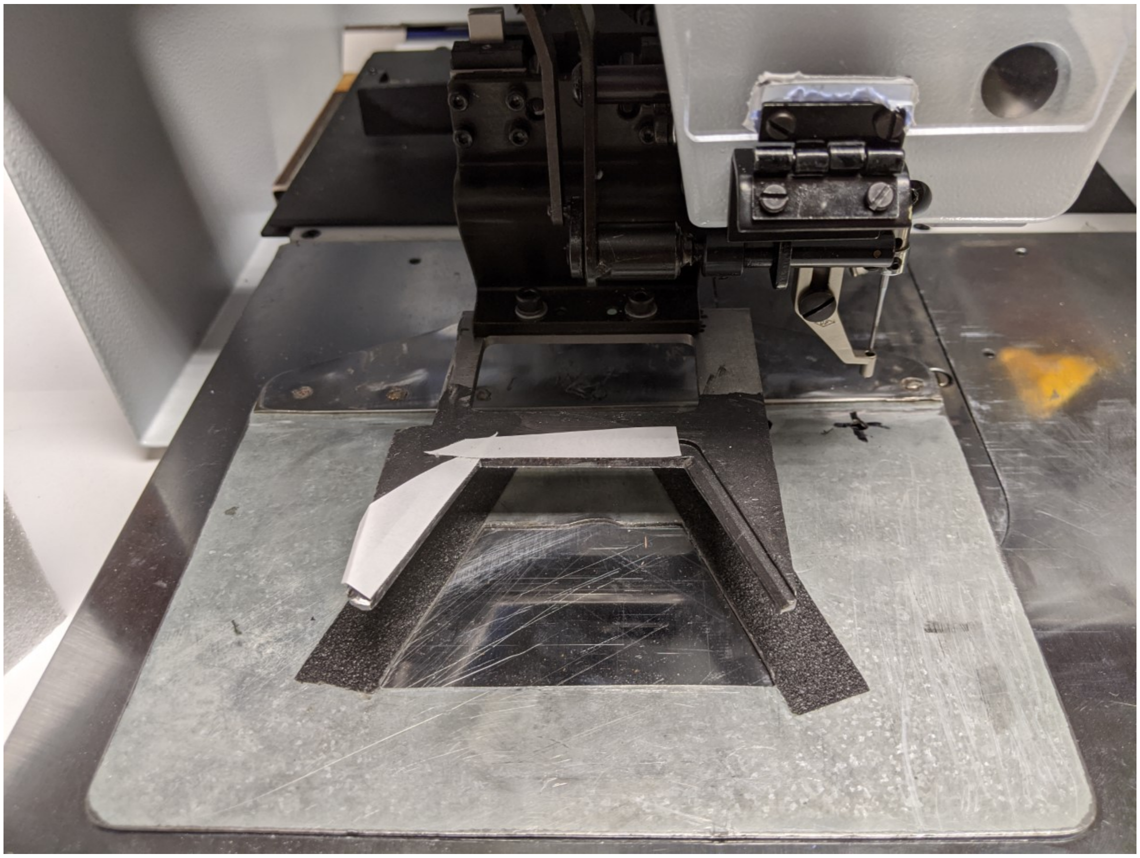

Figure 3.12: Clamp on programmable stitching machine in its partial closed position, the white markers act as guides for the fabric alignment

ground from the target area.

4. Remove specular noise from the image by performing a logical ”and” operation on several consecutively taken images.

5. Approximate the “center” of the pattern using the OpenCV contour tool

6. Generate the two points, i and j, at a 90° angle and same distance from the center.

7. From points i and j, generate the distances A and B, between the bottom of the pattern and edge of the clamp which serves as an alignment guide.

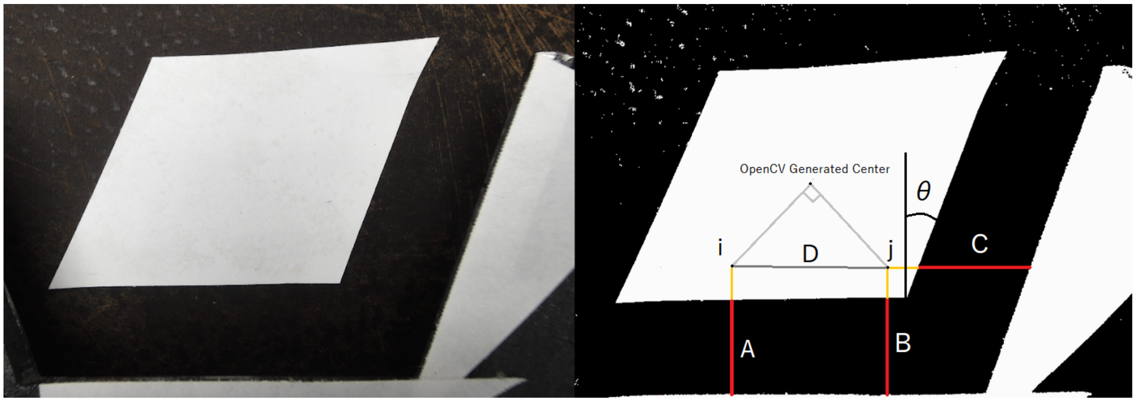

Figure 3.13: Visualization of the second stage vision processing. The generated distances A, B, and C are used along with known values D and θ to generate the necessary alignment

8. Determine the angular error in the alignment using the formula:

$$AngularError = \phi = arcsin(D(B - A)) \tag{3.1}$$

9. Rotate the material around the center of the pattern by the amount of the angular error so that the edge of the pattern are parallel with the edges of the guide on the clamp.

10. Generate the horizontal and vertical error using the following formulae:

$$\begin{aligned} VerticalError = y = \frac{A + B}{2} \\ HorizontalError = x = (C - (y)tan(\theta)) \end{aligned} \tag{3.2}$$

11. Move the material so that the processing pattern is aligned with the guide geometry on the clamp.

12. Fully close the clamp.

Figure 3.14: Example before and after images of the second stage alignment process

3.5 Independently Controlled Wrist

This section describes the addition of an independently controlled wrist capably of low latency control in order to expand the capability of the total robotic system. The SDA10F robot is capable of very high precision, power, and speed, but has limited dynamic control and high input latency. The SDA10F is driven by the FS100 control box which takes the intended trajectory from the control software via ROS as its input. When a new trajectory command is received by the FS100 stops all current motion, verifies the integrity of the intended trajectory, and then executes the new trajectory. This process can take up to a second, severely limiting the capability for adjustments based on sensory feedback. The SDA10F does however transmit its joint

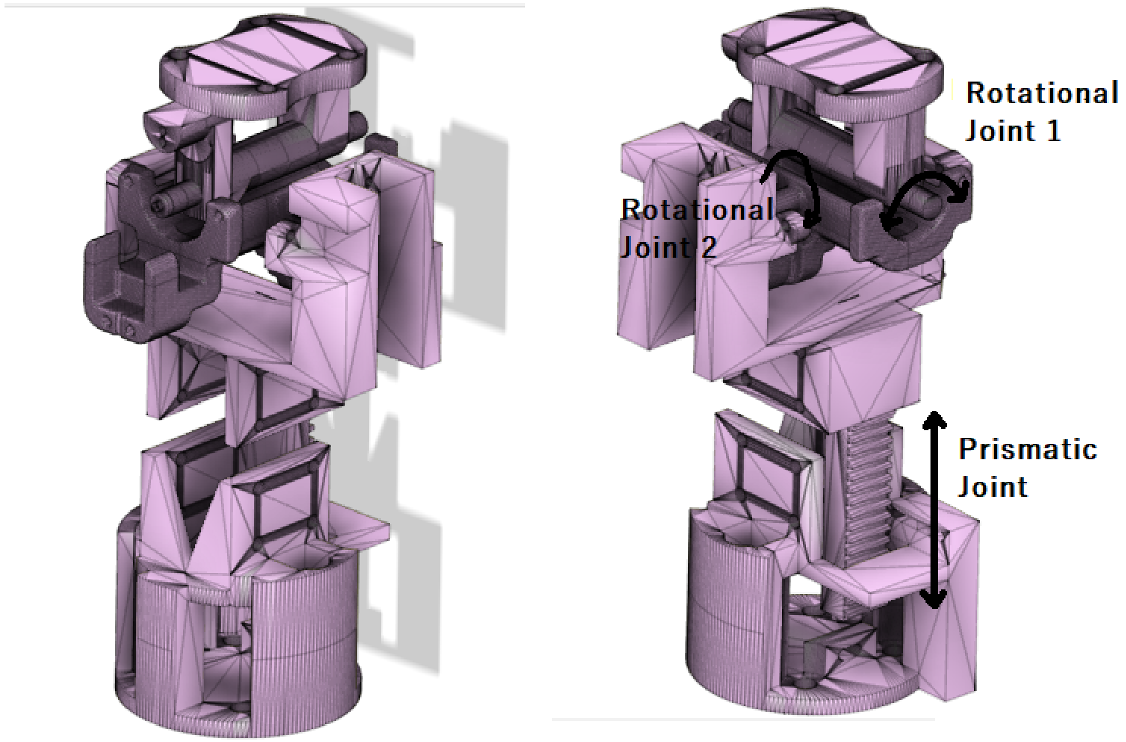

Figure 3.15: 3D model of independently controlled wrist

coordinates and position at a high frequency, so while it is not possible to strictly control its position, it is possible to continuously monitor it. The concept behind the independently controlled wrist is by keeping track of the robot's position and controlling the position of the wrist, the position of the end effector can be controlled in the global coordinate frame. The independently controlled wrist can be added with end-effector in the form of the Robotiq gripper with finger attachments or it can be added with even more biologically realistic hand like Accurate Prosthetic Hand [34] developed recently in Popovic Labs.

3.5.1 Mechanical Design

Unlike a spherical wrist, which contains three rotational degrees of freedom, the wrist contains two rotational joints as well as a prismatic joint which can extend and retract the wrist. This is because, unlike a spherical wrist, whose purpose is to provide precise orientation control, the primary purpose of this wrist is to provide positional control. The total angular range of each rotational joint in the wrist is 45° and the range of the prismatic joint is 45mm. Fully retracted the wrist is 192mm long.

The wrist consists of four custom components which are 3D printed in 40% density Acrylonitrile Butadiene Styrene (ABS). One end of the wrist can be affixed with machine screws to the end of the SDA10F's arm and the other end can be affixed to the Robotiq 2F-85. The linear motion is provided by a Greartisan 25 RPM 12-volt worm gear driven DC motor [35] on the base of the wrist drives a gear rack supported by two linear ball bearing slides. The motor is driven using a 12-volt power supply and DROK L298 Dual H Bridge [36]. The gear driven by the motor has 16 spurs and the spur density of the gear rack is 2.86 spurs/cm. A 25 RPM maximum speed yields a maximum linear speed of 2.33 cm/s. Position control for the linear motion is achieved with feedback from a linear potentiometer attached to the gear rack and base of the wrist. The two rotational joints are driven by Miuezuth 20KG Digital Servos [37] which are mounted on rotational bearings. The servo motors are position controlled via pulse width modulation (PWM). The PWM signal sent to the motor corresponds with the desired angle and the servo moves to that angle using an internal control system.

Figure 3.16: Left: Servo motor which drives the rotational joints of the Wrist, Right: Worm gear motor which drives the prismatic joint of the wrist

3.5.2 Kinematics

The positions of the prismatic and two rotational joints of the robot respectively represented as a_1 θ_1 and θ_2 .L_1, L_2, and L_3 are the lengths of each joint. L_1 is 148 mm, L_2 is 9mm, and L_3 is 35mm. The Cartesian coordinates of the tip of joint 3 are x, y, and z, where the origin is at the connection point between the wrist at the robot at the base of L1. The joint positions are considered at zero when a_1 is fully retracted and when θ_1 and θ_2 are such that L_1, L_2, and L_3 are co-linear.

The forward and inverse kinematics equations of the wrist are as follows:

$$\begin{aligned}
x &= sin(\theta_1)L_2 + sin(\theta_1)cos(\theta_2)L_3 \\
y &= sin(\theta_2)L_3 \\
z &= L_1 + a_1 + cos(\theta_2)L_2 + cos(\theta_1)cos(\theta_2)L_3
\end{aligned} \tag{3.3}$$

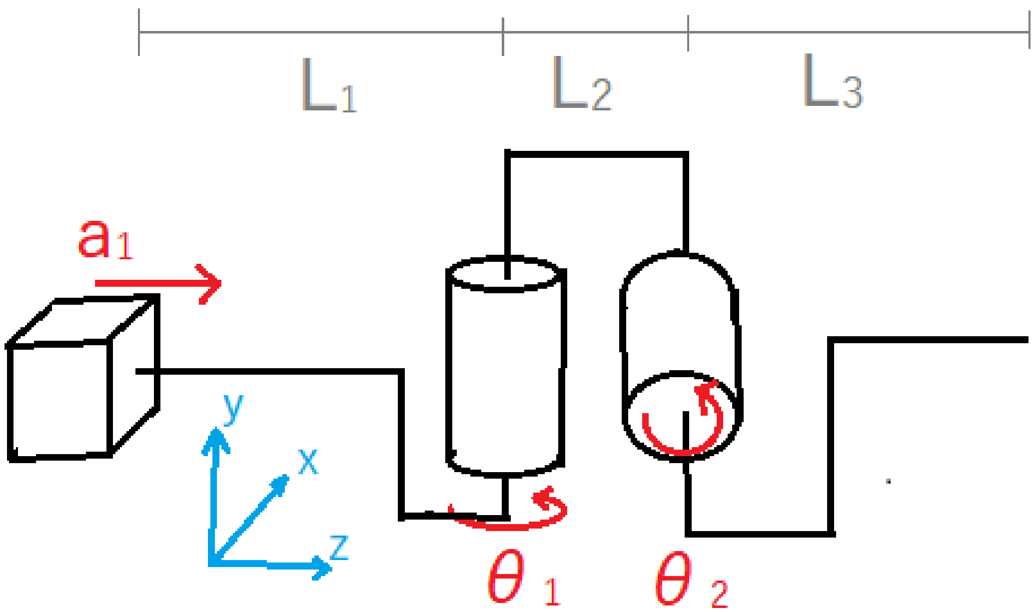

Figure 3.17: Kinematic model of the wrist, the kinematics equations are based on the variables shown in this diagram

$$\begin{aligned}
&\theta_2 = arcsin(\frac{L_3}{y}) \\
&\theta_1 = arcsin(L_2 + \frac{cos(\theta_2)L_3}{x}) \\
&a_1 = L_1 + cos(\theta_2)L_2 + cos(\theta_1)cos(\theta_2)L_3 - z
\end{aligned} \tag{3.4}$$

3.5.3 Software control

When a new position command is sent to the wrist, the linear component of the motion begins first, with the 12V motor moving prismatic joint toward its target length using proportional gain control. The linear potentiometer which provides positional data for this motion is also used to synchronize the motion of the servos. The servo motors are sent positional commands so that they progress towards their

Figure 3.18: Independently controlled wrist affixed to Robotiq Gripper as well as pen

goal position at the same relative rate as the linear motor is progressing toward its position. This causes the wrist to move in a smooth arc, regardless of the speed of the linear motor, which is usually slower than the rotational servos. The wrist is controlled by an Arduino Uno microprocessor which is able to communicate with a python script over USB. The python script uses the Klampt inverse kinematics library [38] to take Cartesian data from the user and transmit it to the wrist.

Chapter 4

Experiments and Results

In this section, the experimental tasks are described and the data from the experiments are presented.

4.1 Fabric Alignment System

The experimental alignment task is to pick up a piece of leather with a paper processing target affixed to it and align that processing target with the clamp under the stitching machine. The alignment task was performed nine consecutive times. Three tests were performed for each of three sizes of pig leather, each with an identical paper parallelogram acting the processing goal. The three leather samples are 100mm by 150mm, 125mm by 200mm, and 105mm by 250mm. There was no change to the software of the system between these trials. The fabric was placed manually and without any guide on the input station with a variation of approximately two centimeters to simulate the possible input variation in a manufacturing system.

The success of the task is measured by comparing the position of the processing target on the leather to the intended position stipulated by the guide on the clamp. The Cartesian and rotational errors are measured over the nine trials. The results of

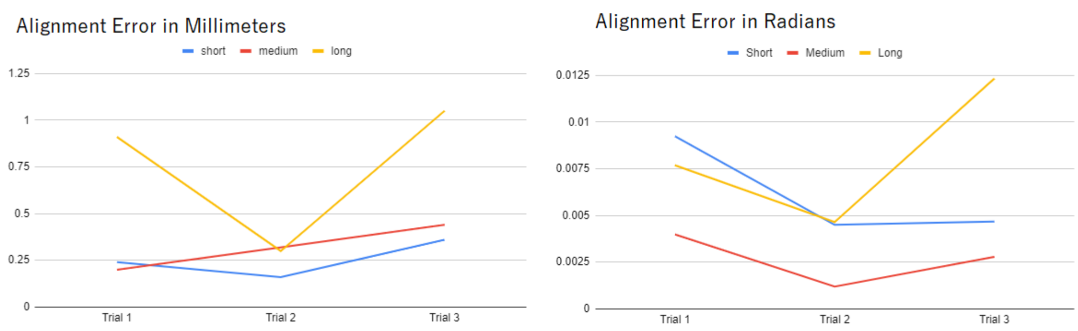

Figure 4.1: Cartesian and Rotational Alignment Error Data

the test are recorded in Figure 4.1 which presents the error position of the processing pattern in terms of Cartesian distance and rotation.

4.1.1 First Stage Alignment

As well as analyzing the final error, the results of the first stage alignment system are also examined. Figure 4.2 presents the Cartesian Error after in the first alignment stage for the nine trials separated by sample size. The differences in characteristic errors between the different materials show the degree of systematic error in the first alignment system.

Figure 4.3 presents how the error in the first alignment step effects the final alignment. The data shows that there is not a significant correlation between the magnitude of error present after the first alignment and the final process error.

4.2 Independently Controlled Wrist

The first test of the robotic wrist was an accuracy test. Several Cartesian coordinates were input into the control software and the position of the end of a pen attached to the wrist using calipers. The results of the five tests are recorded in Table 4.1.

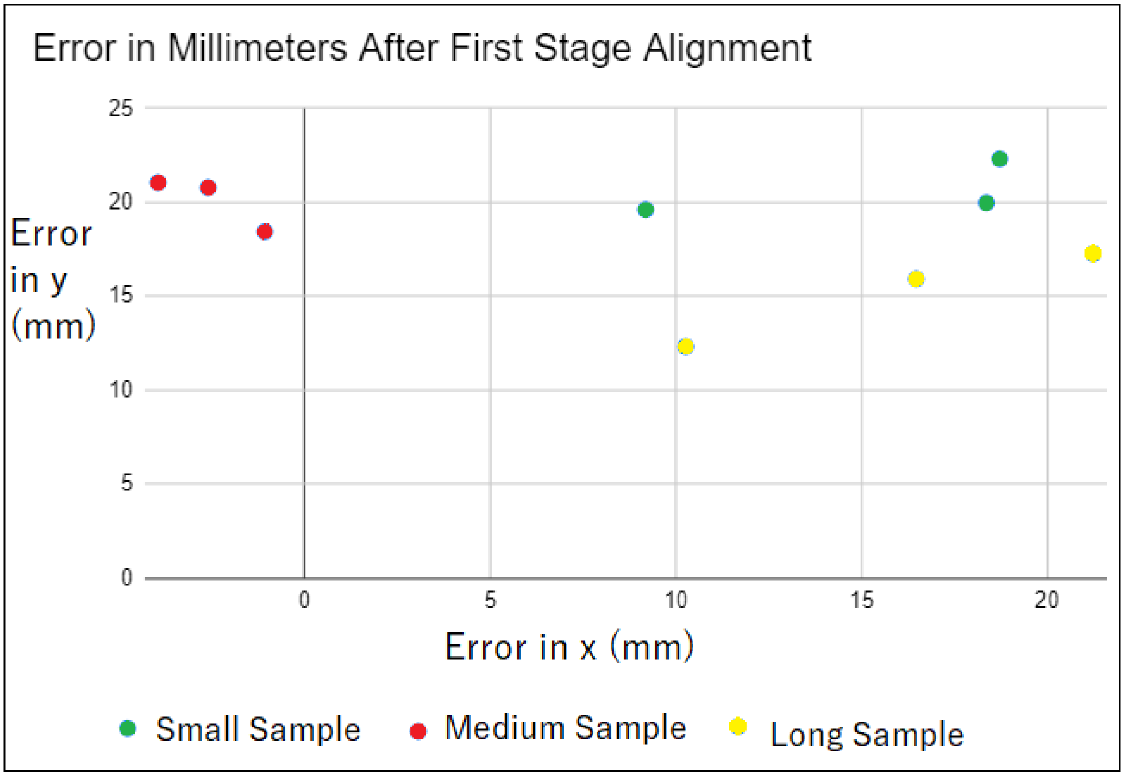

Figure 4.2: How the Error in the first alignment system effect the final task error

The second test of the wrist was an evaluation of its ability to follow a trajectory smoothly and accurately. A ball point pen was attached to the wrist and it was given a 2000 point trajectory to follow over the surface of a piece of paper. Two trajectories were tested, a circle a sine wave that loops back to a negative sine wave. The results of these tests are shown in figure 4.5.

X target	Y target	Z target	X result	Y Result	Z Result	Total Error
30 mm	30 mm	12 mm	29.0 mm	28.4 mm	12.9 mm	2.0 mm
-40 mm	10 mm	22 mm	-38.5 mm	10.5 mm	23.4 mm	2.2 mm
-30 mm	50 mm	17mm	24.8 mm	48.5 mm	17.7 mm	5.5 mm
-20 mm	-20 mm	70 mm	-19.8 mm	22.3 mm	67.0 mm	4.2 mm
10 mm	30 mm	13 mm	11.2 mm	27.5 mm	14.7 mm	3.25 mm

Table 4.1: [Wrist Position Error Data]

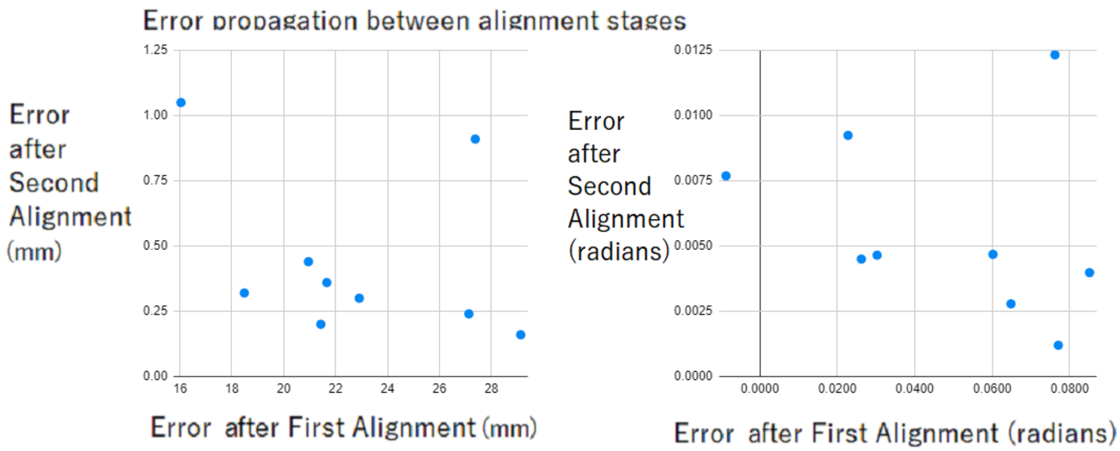

Figure 4.3: The Cartesian error of the first state alignment system

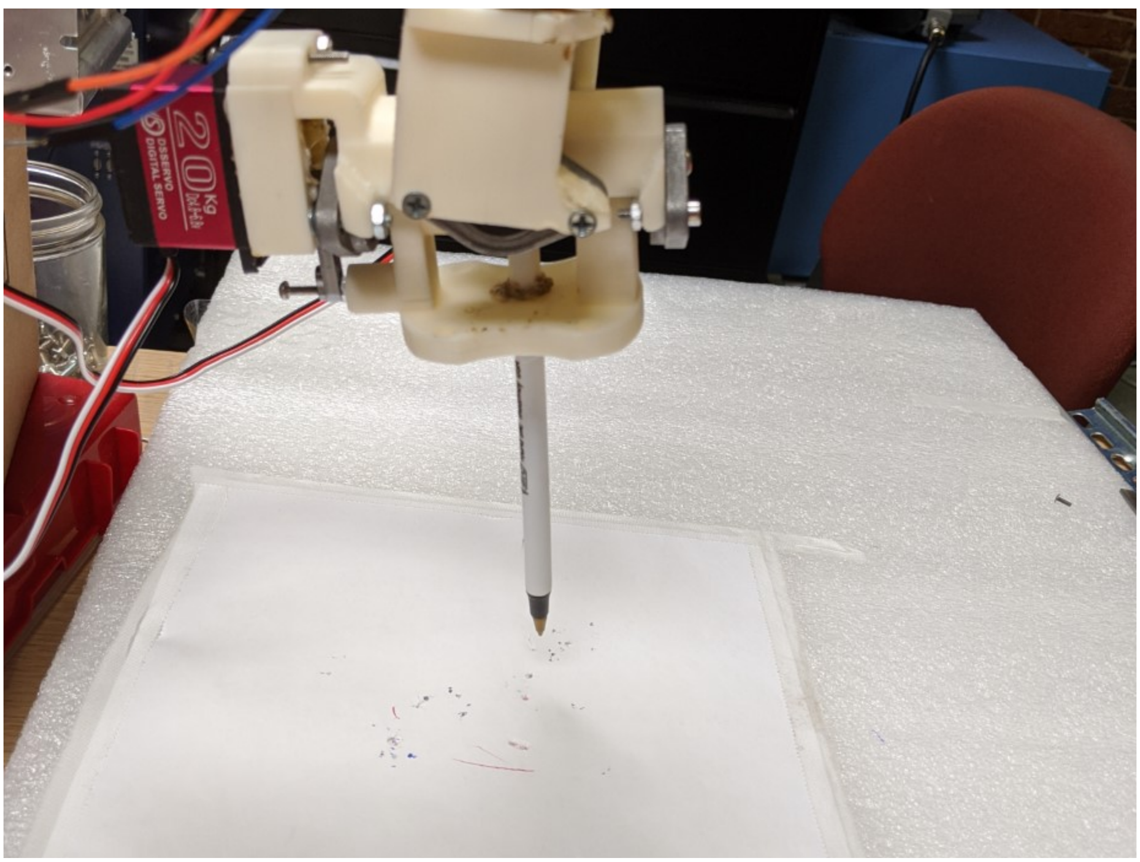

Figure 4.4: Wrist Accuracy Test Setup

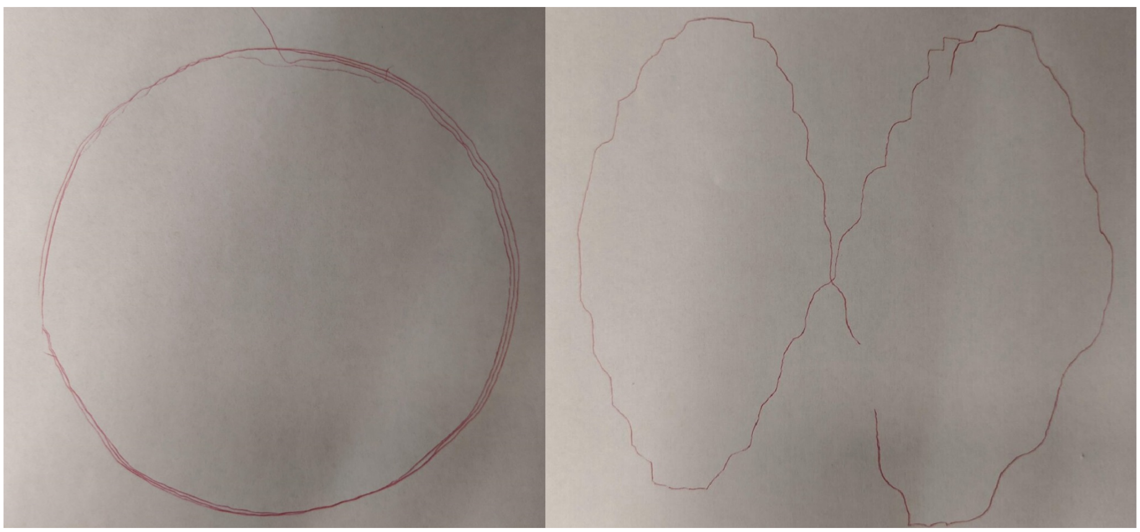

Figure 4.5: Circle and Sine wave drawn by the wrist

Chapter 5

Discussion

In this section, the results of the experiments are discussed to evaluate the level of success of the system and identify possible areas for future improvement.

5.1 Fabric Alignment System

The results of the fabric alignment process demonstrated that the system can perform its task one several different sizes of fabric without any modification to system hardware or software. The largest samples, which had the highest degree of material deformation, showed the largest errors, but an error of approximately 1mm is acceptable in many fabric assembly tasks. The increase in error in larger samples are to be expected, as the larger samples will exhibit more irregularity and unexpected behavior during processing. For example, the further from the rigidly controlled robotic gripper the processing target, the more the soft intervening material will reduce the precision of the control. There is also significant systematic error in the first alignment step, as the size of the material effects the error in that step. It is unclear exactly what causes this error, but there appears to exists a property of the fabric that the system is not accounting for during the first alignment stage.

However, these errors are able to be corrected in the second stage alignment, and the error in the first stage does not significantly effect the final error.

Both of the computer vision algorithms developed here are designed to be simple and easily modifiable for use in similar tasks. The first stage alignment system will function on many sizes and shapes of leather and likely on other fabric of similar material properties. If a new processing target cannot be identified with visual contour tools, changing the way that the program recognizes the center of the object will correct the issue and the rest of the software will work unmodified. By using a mono-color background, the contour of any piece of fabric should be easily identifiable. If the algorithm can identify the contour of the fabric and position processing target it will be able to move them into the correct processing position.

The second stage alignment system, because it is based on the geometry of the target area, would need slight modifications to work for different shapes. A simple scaling factor would be sufficient to work on a differently sized target area. For simple geometric shapes involving straight lines, the same alignment formula could be applied, all that would be necessary would be to determine from what points on the shape to measure the distance to a guide.

5.2 Robotic Wrist

The positional accuracy of the wrist was verified to the limit of hardware, the servo motors used have a maximum angular precision of 1.4 degrees, which given the 0.15m arm used in the experiments can create an error of up to 5.1 mm. This is very close to the average error observed in tests of the wrist. It is therefore likely that in order to improve upon the accuracy of the wrist, the servos can be exchanged for high precision stepper motors. This may cause a reduction in speed and torque performance but for the intended purpose of this device, providing small adjustments for fabric assembly tasks, torque and speed are not likely to be priorities.

In the drawing test, the wrist was able to draw a circle much more accurately then it was able to draw a sine wave. This is likely due to the fact that drawing with a ballpoint pen is relatively error-tolerant in the axes in the plane of the page, but is not error tolerant in the in the axis perpendicular to the page. When drawing a circle, the wrist did not need to change its distance from the page, and thus the visible error is minimal. When drawing the sine wave, the height of the gripper was changing constantly, and so small errors in that axis cause the pen to warp, creating a much more visible error in the drawing.

When the 0.9 kg Robotiq Gripper is attached to the end of the wrist, the worm gear driven the prismatic joint can slow considerably. Additionally, the momentum of the hand can cause the wrist to bend and oscillate if the hand is moved at a high speed. There are two solutions to this problem. The first would be to use higher power motors, as well as a fully metal machined frame to reduce oscillation. The other solution is to use a smaller and lighter gripper at the end of the wrist. A lightweight gripper is likely to supply sufficient grasping power to satisfactorily manipulate fabric for the purposes of further automation research.

5.3 Conclusion

During this work, a fabric assembly automation platform was created, and its capabilities were demonstrated by reliably performing a proof of concept manufacturing assembly task. The system was designed considering the many challenges and barriers to automating fabric assembly tasks. A fabric assembly automation tool must be able to contend with variable material properties and frequent task changes.

It was demonstrated that the system could perform its task on input materials of different sizes and shapes, as well as with input placement deviation, as would be required of a viable manufacturing platform. A robotic wrist attachment was developed and tested to enhance the feedback capabilities of the system. The limitations of the robotic wrist are well understood and opportunities for improvement have been discussed. The addition of the wrist opens up new research possibilities in the area of fabric assembly. Research into more automating more complex actions such as direct sewing is now possible. The ROS integrated computer vision framework created during this process is designed to be easily adaptable for different shapes and sizes of fabric and alignment goals. The necessity for robust fabric assembly automation tools is clear. As the world becomes more industrialized, the demand for clothing and other textile products is rising, while production is becoming more expensive. The development of economical and robust fabric assembly tools will become increasingly valuable in the years to come.

Appendix A

A.1 Inverse Kinematics Python Script

```
import sys
import time
import math
import klampt
import OpenGL
import numpy as np
from klampt import vis
from klampt.model import ik
from klampt import IKObjective,IKSolver
import pyautogui
from pySerialTransfer import pySerialTransfer as txfer


if __name__ == '__main__':
try:
com = txfer.SerialTransfer('COM14')
com.open()
time.sleep(2) # allow some time for the Arduino to completely reset\

world = klampt.WorldModel()
world.loadElement("testrobot.rob")


robot = world.robot(0)
robot.setConfig([0,0,0])
link = robot.link(2)
```

```
print("Position")
print(link.getWorldPosition([1,0,0]))
print(robot.getConfig())
# Wrist.setName("Wrist")
goal = [0.3,0,0]

obj = ik.objective(link,local=[1,0,0],world=goal)
solver = ik.solver(obj)
solver.solve()
robot.getConfig()
print( solver.getResidual())

vis.add("world",world)
vis.add("local point",link.getWorldPosition([1,0,0]))
vis.setAttribute("local point","type","Vector3")
vis.add("target point",goal)
vis.setAttribute("target point","type","Vector3")
vis.setColor("target point",1,0,0)
vis.show()

i =  0.5
time.sleep(3)

while vis.shown():
vis.lock()

x, y = (pyautogui.position())
x = x/7000.0 - 0.08
y = y/6000.0 - 0.08

goal = [0.3,x,y]
i = i-0.002
obj = ik.objective(link,local=[0.165,0,0],world=goal)
solver = ik.solver(obj)
solver.solve()

vis.add("local point",link.getWorldPosition([0.165,0,0]))
vis.setAttribute("local point","type","Vector3")
vis.add("target point",goal)
vis.setAttribute("target point","type","Vector3")

vis.unlock()
time.sleep(0.1)
```

```
send_size = 0
zdata, xdata, ydata = (robot.getConfig())
xsend = int((xdata+0.3) / 0.002737)
ysend = int((ydata+0.3) / 0.002737)
zsend = int((zdata)*1000*231.0/46)

if (xsend == 0):
xsend = 1
if (ysend == 0):
ysend = 1
if (zsend == 0):
zsend = 1

print(xsend,ysend,zsend)
list_ = [xsend,ysend,zsend]
print (sys.getsizeof((list_)))
list_size = com.tx_obj(list_)
send_size += list_size
com.send(send_size)
while not com.available():
if com.status < 0:
if com.status == -1:
print('ERROR: CRC_ERROR')
elif com.status == -2:
print('ERROR: PAYLOAD_ERROR')
elif com.status == -3:
print('ERROR: STOP_BYTE_ERROR')
rec_list_  = com.rx_obj(obj_type=type(list_),
                         obj_byte_size=list_size,
                         list_format='i')
print('SENT: {}'.format(list_))
print('RCVD: {}'.format(rec_list_))
print(' ')
if done():
vis.show(False)
except KeyboardInterrupt:
com.close()
except:
import traceback
traceback.print_exc()
com.close()
```

A.2 Wrist Control Code on Ardunio Uno

```
#include "SerialTransfer.h"
#include <Servo.h>
// bottom is 930, top is 100, 43mm length
int analogPin = A0; // potentiometer wiper (middle terminal) connected to analog pin
int val = 400;  // variable to store the value read
int pos = 0;
Servo myservo;  // create servo object to control a servo
Servo myservo2;  // create servo object to control a servo
Servo myservo3;  // create servo object to control a servo
SerialTransfer myTransfer;
uint16_t datarx[10] = {};
int xgoal = 0;
int ygoal = 0;
int old_xgoal =  0;
int old_ygoal =  0;
int old_zgoal =  0;
int zgoal = 100;
int factor = 1;

void setup()
{
  Serial.begin(115200);
  myservo.attach(5);
  myservo2.attach(6);
  myservo3.attach(11);  // attaches the servo on pin 9 to the servo object
  myTransfer.begin(Serial);
  pinMode(11, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);
  myservo.write(90);
  myservo2.write(60);
}
void loop()
{
  val = analogRead(analogPin);  // read the input pin
```

```
  pos = ((val-120)*255.0/950.0);

  if(myTransfer.available())
  {
    val = analogRead(analogPin);  // read the input pin
    pos = ((val-120)*255.0/950.0);
    myTransfer.rxObj(datarx, sizeof(datarx), 0);
    factor = max(0,min(1, float(pos - old_zgoal) / float(zgoal - old_zgoal)));
    old_xgoal =  xgoal;
    old_ygoal =  ygoal;
    old_zgoal =  zgoal;
    xgoal = myTransfer.rxBuff[0]/255.0*40.0;
    ygoal = myTransfer.rxBuff[4]/255.0*40.0;
    zgoal = abs(myTransfer.rxBuff[8]- pos)*10;

    analogWrite(3,myTransfer.rxBuff[8]);
    analogWrite(11,min(abs(pos - myTransfer.rxBuff[8])*3  ,255));
    if (pos >= myTransfer.rxBuff[8]) {
      digitalWrite(10, HIGH);
      digitalWrite(9, LOW);
      //myservo3.write(abs(pos-myTransfer.rxBuff[8]));
    }
    if (pos < myTransfer.rxBuff[8]) {
      digitalWrite(9, HIGH);
      digitalWrite(10, LOW);
      //myservo3.write(abs(pos-myTransfer.rxBuff[8])*5);
    }
    // send all received data back to Python
    for(uint16_t i=0; i < myTransfer.bytesRead; i++) {
      myTransfer.txBuff[i] = myTransfer.rxBuff[i];
    }
    myTransfer.sendData(myTransfer.rxBuff[0]);
  }
}
```

A.3 ROS Package

The ROS package which includes the control code for the SDA10F as well as the first and second stage vision alignment algorithms can be found in the following github repo. https://github.com/briggscalum/MotoWorkspace

References

[1] The World Bank,The Changing Nature of Work, Washington, D.C: International Bank for Reconstruction and Development 2019, pp. 17–28.

[2] World Trade Organization, World Trade Statistical Review: 2018. Geneva, Switzerland: World Trade Organization, 2019, pp. 143.

[3] World Trade Organization. World trade report 2018: the future of world trade: how digital technologies are transforming global commerce. Geneva: World Trade Organization, 2018, pp. 50.

[4] N. S. Choubey and M. Agrawal, "Automation in Textile Industry," International Journal on Textile Engineering and Processes, Jan. 2016.

[5] Sanchez, Jose, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey." The International Journal of Robotics Research 37, no. 7 (2018): 688-716.

[6] P. M. Taylor, Sensory robotics for the handling of limp materials. Berlin: Springer-Verlag, 1990.

[7] F. Vercraene and P. Esquirol, "Analysis of a Ply-Separation Gripper," Sensory Robotics for the Handling of Limp Materials, pp. 127–136, 1990.

[8] R. B. Kelley, "2D Vision Techniques for the Handling of Limp Materials," Sensory Robotics for the Handling of Limp Materials, pp. 141–157, 1990.

[9] F. W. Paul and E. Torgerson, "Tactile Sensors: Application Assessment for Robotic Handling of Limp Materials," Sensory Robotics for the Handling of Limp Materials, pp. 227–237, 1990.

[10] The Apparel Research Committee of the American Apparel Manufacturers Association, The U.S. textile and apparel industry: a revolution in progress. Washington, D.C.: Congress of the United States, Office of Technology Assessment, 1987.

[11] H. A. Seesselberg, "A Challenge to Develop Fully Automated Garment Manufacturing," Sensory Robotics for the Handling of Limp Materials, pp. 53–67, 1990.

[12] R. Nayak and R. Padhye, Automation in garment manufacturing. Duxford, England: Woodhead Publishing, an imprint of Elsevier, 2018.

[13] Softwear Automation Inc,
`http://softwearautomation.com`

[14] Sewbo Inc
`https://www.sewbo.com/`

[15] R. Fanson and A. Patriciu, "Model based deformable object manipulation using linear robust output regulation," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.

[16] J. Das and N. Sarkar, "Planning and control of an internal point of a deformable object," 2010 IEEE International Conference on Robotics and Automation, 2010.

[17] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.

[18] S. Kinio and A. Patriciu, "A comparative study of H and PID control for indirect deformable object manipulation," 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012.

[19] P. T. Zacharia, "An Adaptive Neuro-fuzzy Inference System for Robot Handling Fabrics with Curved Edges towards Sewing," Journal of Intelligent and Robotic Systems, vol. 58, no. 3-4, pp. 193–209, 2009.

[20] D. Seita, N. Jamali, and M. Laskey, "Deep Transfer Learning Using Depth Sensing of Deformable Fabric," Oct. 2018.

[21] Yaskawa, SDA10F, 2017
`https://www.motoman.com/en-us/products/robots/industrial/assembly-handling/sda-series/sda10f`

[22] Robotiq, 2F-85 Two-Finger Gripper, 2020.
`https://robotiq.com/products/2f85-140-adaptive-robot-gripper`

[23] ORISOL, B1510-OCS Computerized Stitching Machine, 2010
`https://static1.squarespace.com/static/59a3d5a88fd4d26e0447bedd/t/5ba33bd988251bd3cdb46df2/1537424346936/OCS-B1510-180920.pdf`

[24] Yaskawa, FS100 Controller, 2017.
`https://www.motoman.com/en-us/products/controllers/fs100`

[25] Arduino, Arduino Uno, 2020
https://store.arduino.cc/usa/arduino-uno-rev3

[26] ELP, 720p Camera, 2018.
http://www.elpcctv.com/hd-720p-usb-cameras-c-85_87.html

[27] e-con Systems, AR1335 USB camera, 2018.
https://www.e-consystems.com/4k-camera-module.asp

[28] C Briggs, T Brown, A Lewis, S O'Grady, W Sullivan, Tactile Fabric Manipulation, Worcester Polytechnic Institute, 2018.

[29] Willow Garage, ROS, 2020.
https://www.ros.org/

[30] Dave Coleman, MoveIt, 2020.
https://moveit.ros.org/

[31] Willow Garage, ROS-Industrial, 2020.
https://rosindustrial.org/

[32] Yaskawa, Moto Ros, 2020.
http://wiki.ros.org/motoman_driver

[33] Intel, OpenCV, 2020.
https://opencv.org/

[34] E. Saint-Elme, M. Larrier, C. Kracinovich, D. Renshaw, K. Troy and M. Popovic, "Design of a Biologically Accurate Prosthetic Hand," 2017 International Symposium on Wearable Robotics and Rehabilitation (WeRob), Houston, TX, 2017, pp. 1-2, doi: 10.1109/WEROB.2017.8383866.

[35] Greartisan, 12V 25RPM Worm Gear Motor
https://www.amazon.com/gp/product/B0721W14ZN/

[36] DROK, L298 Dual H Bridge, 2020.
https://www.amazon.com/gp/product/B06XGD5SCB/

[37] Miuezuth, 20KG Digital Servo, 2020.
https://www.amazon.com/gp/product/B07PG8M4G9

[38] Intelligent Motion Laboratory, Duke University, Klampt, 2020.
https://github.com/krishauser/Klampt