Zhejiang University - University of Illinois at Urbana-Champaign Institute

# Senior Design Final Report

# SMART LAUNDRY FOLDBOT

By

Jiadong Hong
jiadong5@illinois.edu
Channing Liu
qianqi2@illinois.edu
Jialin Shang
jialins5@illinois.edu
Weijie Liang
weijiel4@illinois.edu

Final Report for Senior Design, Spring 2024

Supervisor: Prof. Yu Lin, Prof. Liangjing Yang

TA: Jiahao Chen

May 29, 2024

Project No. 10

## Acknowledgement

We would like to sincerely thank Professor Yu Lin and Professor Liangjing Yang, our project sponsors, for their enthusiastic and meticulous guidance, which has been crucial to the smooth progress of our graduation project. Thank Jiahao Chen, our team’ s teaching assistant, for his timely communication with us and the professor review team throughout the project. Thank Professor Mark Butala and Professor Timothy Lee, the course directors, for their encouragement during weekly updates and valuable design feedback. We also thank the laboratory faculty and assistants of the ZJUI teaching laboratory team for their support in processing and presenting our project. Special thanks to Mr. Zhizhan Li, the Meta RoboMaster Team leader in ZJUI, for offering valuable mechanical parts and advice on mechanical assembly and electrical control.

## Abstract

We proposed a fully automated household folding solution for this problem, The Smart Laundry FoldBot project. In response to the growing demand for household automation, this project focuses on implementing an intelligent T-shirt folding machine capable of efficiently handling scattered clothes. While existing solutions can only fold flattened garments, our proposed solution involves the development of an automatic T-shirt folding machine equipped with advanced computer vision and robotic manipulation techniques. By leveraging vision models, our solution can recognize the T-shirt's shape, flatten the T-shirt through automated assembly line design, and fold the T-shirt. The automated assembly line design incorporates a flexible grabbing mechanism and a folding board with special designs that can flatten and fold T-shirts in different statuses.


**Keywords:** Clothes Folding, Clothes Flattening, Automated Assembly Line, Robotics, Computer Vision

# Contents

# 1 Introduction

## 1.1 Problem Statement with Background Information

Automating clothes folding in industrial settings presents significant challenges and opportunities for innovation. Presently, large-scale users such as dry cleaning facilities, clothing recycling centers, major retail outlets, and hotels face stringent demands for rapidly and efficiently folding a diverse array of garments. This task is predominantly performed manually, which is costly and lacks efficiency.

Although a few products are available in the market, such as FilpFold Board[1], T-Fold[2], and FoldiMate[3], they are limited to simple folding tasks and require garments to be perfectly flat and precisely positioned on a specific workbench area. This still necessitates considerable human intervention and fails to meet the needs of industrial-scale operations. Consequently, there is a notable absence of solutions that fully automate clothes flattening and folding.

## 1.2 Project Motivation and Propose

Our project proposes to solve the ”first step problem” of the current folding clothes machine, which requires manual labor to lay clothes flat and place clothes onto the folding machine. The motivation behind the solution is that the step of spreading the clothes flat and accurately on the folding mechanism is no longer necessary through this project.

We have developed a new industrial-grade automatic clothes-folding robot to tackle the challenges outlined. This robot seamlessly integrates into existing production lines, allowing users to directly input various types of disordered clothing.

## 1.3 Functionality

Equipped with a mechanical gripper and a vision module, our solution can accurately sort and position garments onto a folding board compatible with most clothing types. Depending on user needs, the garments are quickly folded and dispatched to a conveyor belt or storage unit. This solution boosts production efficiency, reduces labor costs, and accelerates garment cleaning and recycling processing speeds.

Figure 1.1 shows a brief figure depicting our system for visual aid.

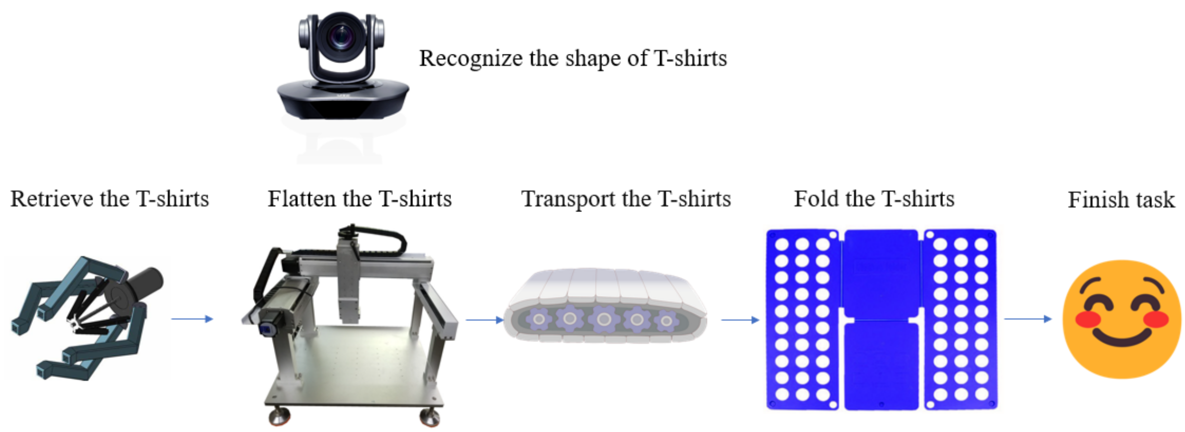



**Figure 1.1** Clothes folding robot workflow chart

## 1.4 System Overview with Block Diagram

Smart Laundry FoldBot project has four main sub-systems, the Central Control System, the Folding System, the Grabbing System, and the Vision System.

Figure 1.2 is the high-level diagram showing each subsystem's position.

Figure 1.3 is the high-level diagram that shows the execution flowchart for the whole system.

Figure 1.4 is the high-level diagram that shows how sub-systems integrate.

Figure 1.5 shows the final vision of FoldBot we built in the ZJUI laboratory.

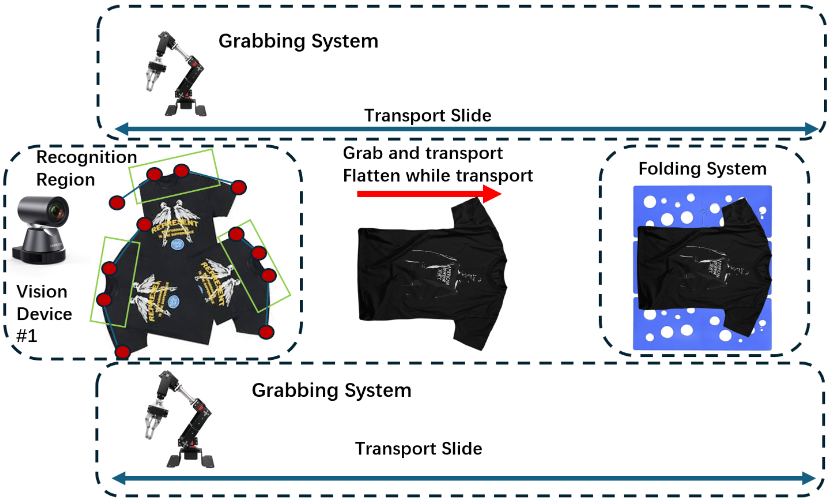



**Figure 1.2** System top-level diagram (Positional)

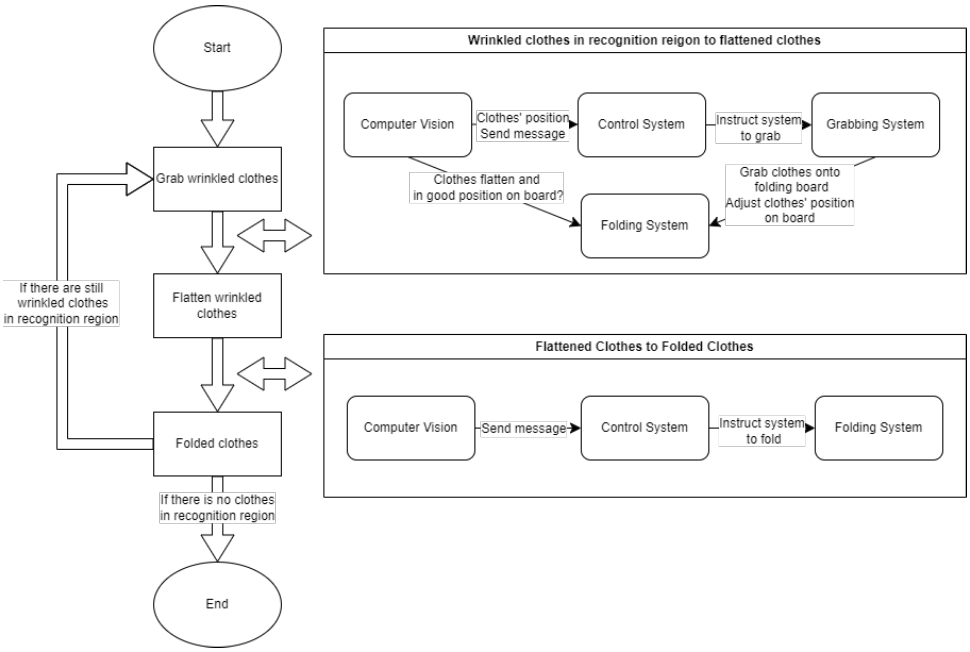



**Figure 1.3** System top-level diagram (Functional)

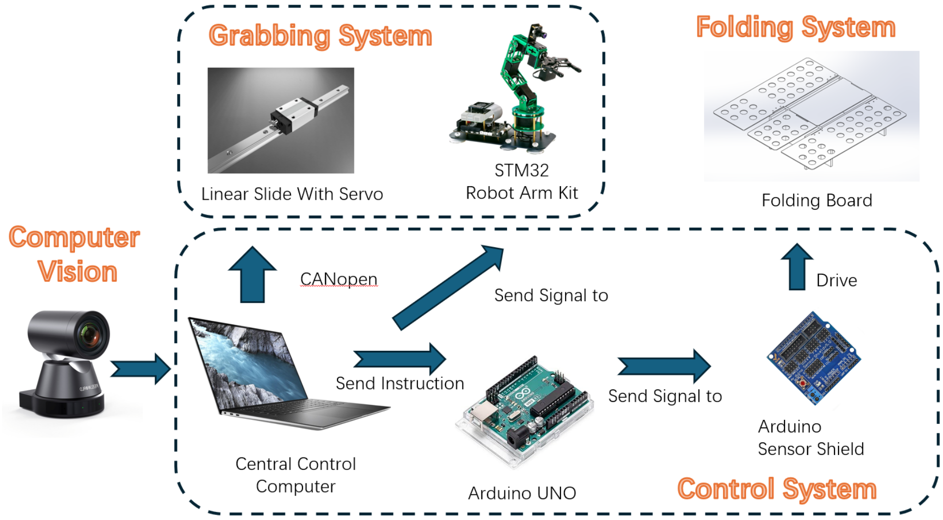



**Figure 1.4** System top-level diagram (Control Signal)

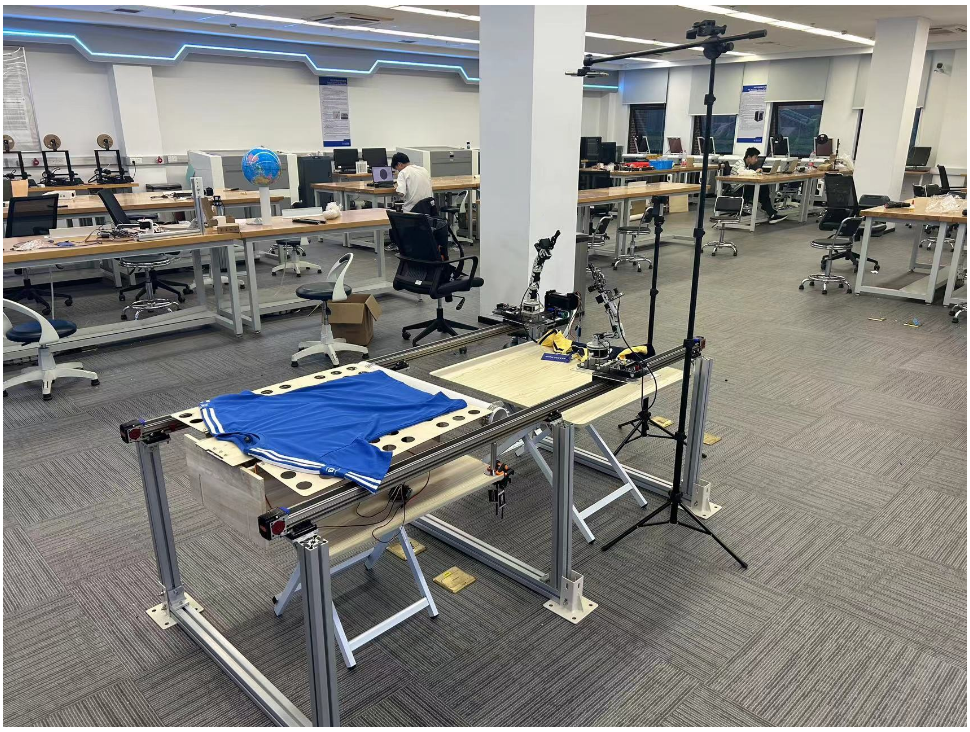


**Figure 1.5** Industrial clothes folding robot

## 1.5 Subsystem Introduction

### 1.5.1 Subsystem 1: Grabbing System

The grabbing system comprises two six-axis robot arms and two transport slides. In the recognition region, vision device No.1 can automatically identify the shoulder of each piece of clothing. Subsequently, the robot arms would be able to accurately grab the clothes' shoulders and transport them to the folding system. Figure 1.7 shows the robot arm connected to the slide. Figure 1.6 shows the connected grabbing system in the laboratory.

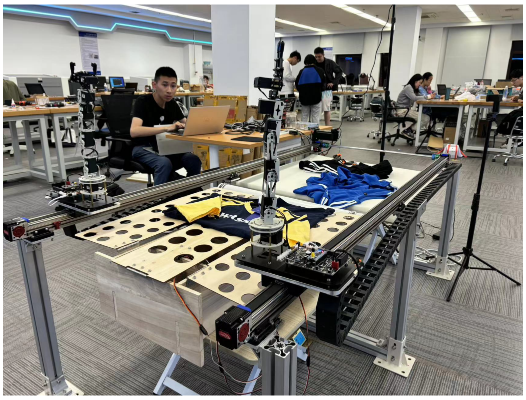


**Figure 1.6** Grabbing System: Final Version

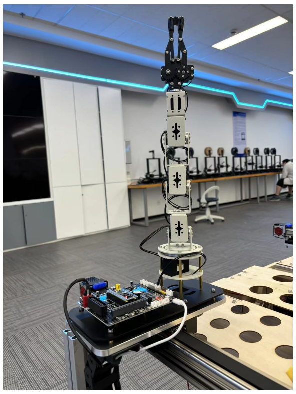


**Figure 1.7** Robot Arm: Final Version

### 1.5.2 Subsystem 2: Folding System

The folding system consists of four folding panels. Each folding board is driven by a servo. Figure 1.8 shows the workflow of the folding system.

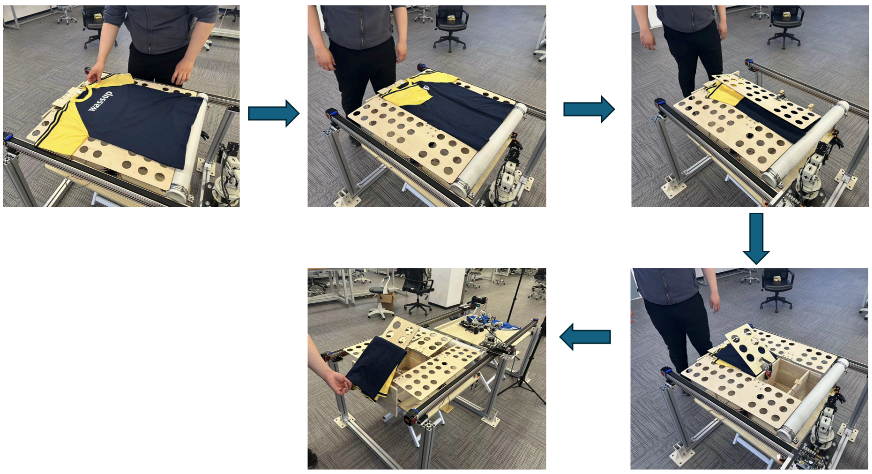


**Figure 1.8** Folding system workflow chart

### 1.5.3 Subsystem 3: Computer Vision System

We use a camera on an aluminum structure to take photos and transmit the images to a PC. We use a laptop PC with NVIDIA GPU as central control and machine learning calculation. The PC will be able to run Fast-RCNN to recognize the grabbing coordinate on the shoulder of the T-shirt.

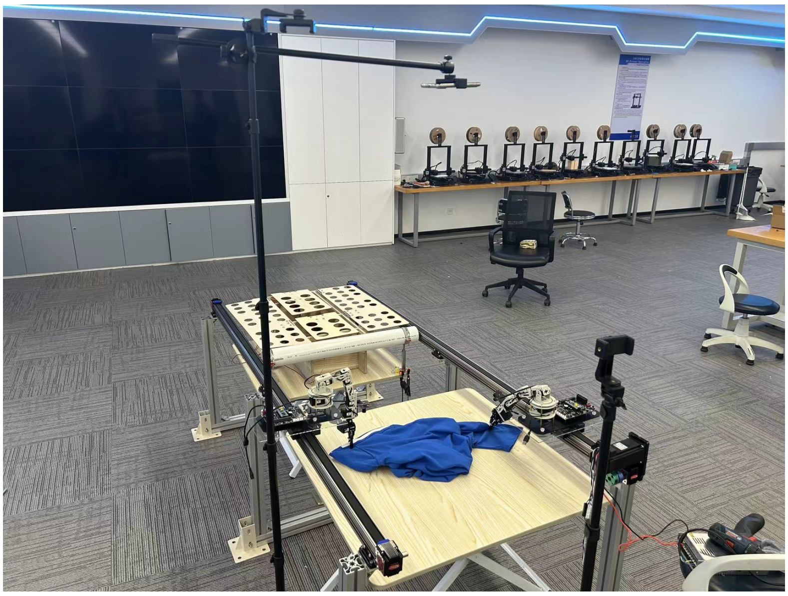


**Figure 1.9** Vision System: Hanging Camera

### 1.5.4 Subsystem 4: Control System

Our purpose for the control system is that the control system can handle the video streams and calculate the kinematic variables the grabbing and folding systems need.

We use the laptop mentioned above to transmit signals and control the components. Figure 1.10 shows how the PC connects and controls all subsystems using various extension devices.

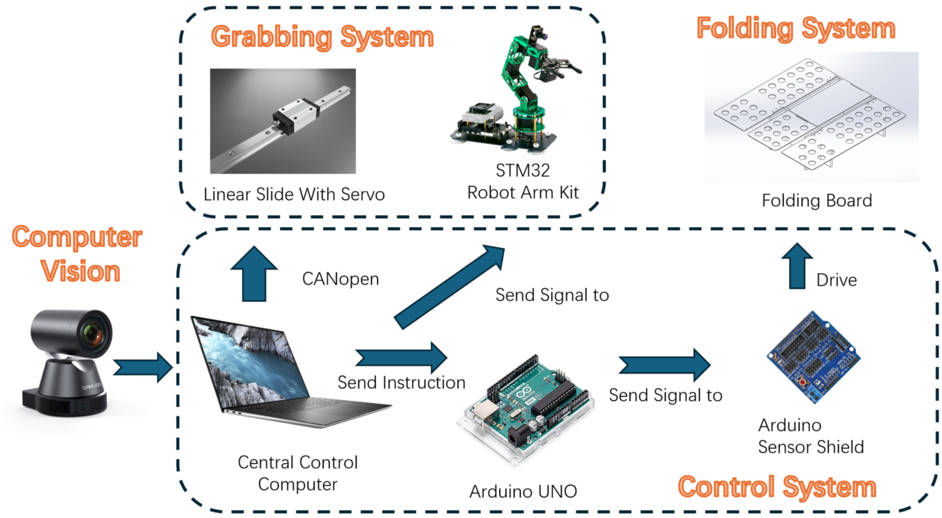



**Figure 1.10** Control System Interaction

# 2 Literature Review

This project focuses on three essential functionalities: efficient folding, accurate recognition grabbing, and effective clothes flattening. This section will briefly review the existing projects or patents related to these three topics.

## 2.1 Efficient Folding: Board Design

Efficient folding has long been a popular topic, with research efforts dating back several decades. The patent ”Shirt-folding board”[4], awarded to Helen M. Ziegler in 1961, designed to fold shirts, is one of the earliest American patents focusing on automating clothing folding garments.

Some existing designs in patents and products can accurately fold clothes, but no product can automate the flat laying of clothes without human intervention. The patent ”Garment folding apparatus”[5] awarded to Deborah Barker proposed a fundamental type of multi-use folding board. Figure 2.1 and Figure 2.2 are illustrations of the patent. This type of folding board has achieved great success in the market and has influenced many other products. For instance, FilpFold[1], an existing product that has sold well since the 2010s, is the same as this patent; T-Fold[2], another existing product in the market, is also based on the patent but driven by electrical motors to fold garments automatically. Figure 2.3 is the product FlipFold of different sizes. Figure 2.4 is the product T-Fold. This type of folding board design has proved efficient in the market. However, this design still needs human labor to place the flat garments on the board.

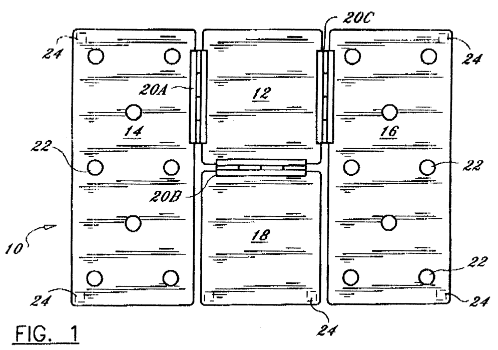



**Figure 2.1** Garment folding apparatus[5]

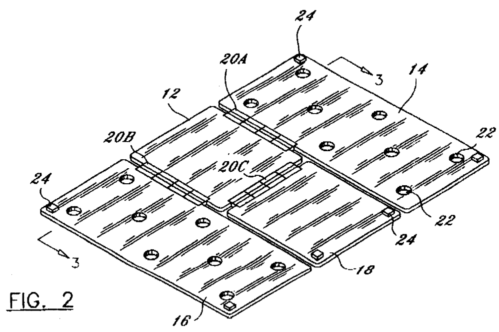



**Figure 2.2** Garment folding apparatus[5]


**Figure 2.3** FilpFold[1]


**Figure 2.4** T-Fold[2]

In recent years, with the continuous development of the home appliance industry, solutions for folding household clothes with smaller folding mechanisms have also emerged. FoldiMate[3], a household clothes folding solution, can fold clothes with clothes hanging flat on the hook of the product. Figure 2.8 illustrates the product FoldiMate. It's worth mentioning that this product uses a brand-new way to fold clothes. The patent ”Fabric article folding machine and method”[6] awarded to Ted Selker and Gal Rozov from FoldiMate proposed a new mechanism using tape and rod to complete folding clothes, which largely reduced space for the folding task. Figure 2.5 and Figure 2.6 show how to fold clothes via rods and tape. The patent ”Device, method and system for folding a moving article of clothing”[7], also awarded to the FoldiMate team, describes the whole folding process of FoldiMate in detail. Figure 2.7 shows how the patent describes the working procedure for folding a T-shirt.

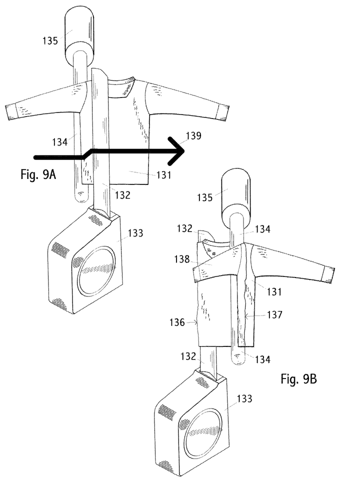



**Figure 2.5** Folding via Rods and Tape[6]

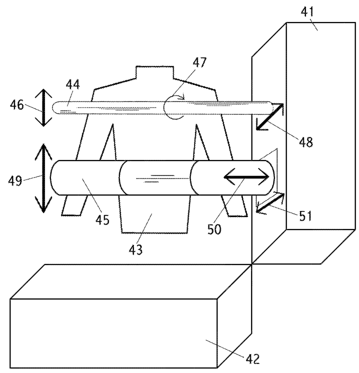


**Figure 2.6** Folding via Rods and Tape[6]

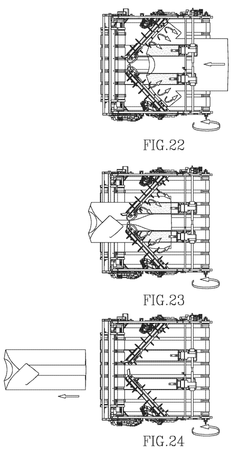



**Figure 2.7** FoldiMate Device Patent[7]

**Figure 2.8** FoldiMate[3]

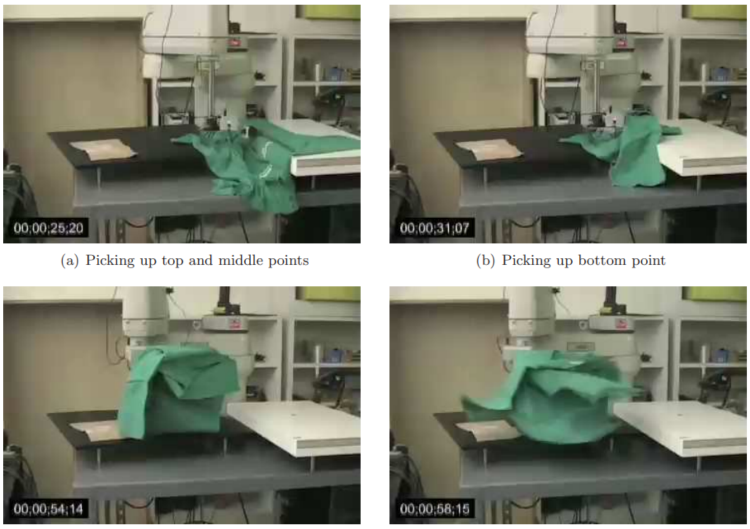
## 2.2 Recognition and Grabbing: Computer Vision and Reinforcement Learning

The booming development of computer vision has made it feasible to identify clothing and machine vision-assisted mechanical control accurately. In recent years, with the continuous development of machine learning and deep convolutional neural networks, computer vision has made great progress in various tasks, making it possible to identify complex clothing states. With the development of Reinforcement

Learning, Projects using robotic mechanisms for challenging tasks through reinforcement learning and computer vision are also emerging. ResNet[8] and its variants Mask RCNN[9], Fast RCNN[10] largely improved the performance of Neural Networks in Computer Vision tasks. Cychnerski[11] has shown the neural network's capability for clothes detection. Sun, Li[12] has shown deep reinforcement learning's promising capability for flattening Wrinkled clothes. Y Tsurumine[13] also shows deep reinforcement learning's capability for clothes smoothing. Fu, Zhao[14] developed a system that can autonomously complete complex mobile manipulation tasks via Reinforcement Learning.

## 2.3 Effective Flattening: Arm System

Some articles offer solutions using arm systems to flatten clothing, but the folding methods provided in these articles are often less than satisfactory. For instance, Bell[15] uses a single-arm system flattening clothes by shaking the clothes as shown in Figure 2.9, Berg et al.[16] develop a two-arm system to flatten clothes via gravity and fold them via the arms. Similarly, the contributions from Xue et al.[17], and Doumanoglou et al.[18] also use two-arm systems, effectively utilizing knowledge from computer vision and robotics to achieve impressive results in flattening and folding operations. This project demonstrates significant advancements in handling garments through automated processes. However, it is noted that the folded clothes are not perfectly neat and orderly after the folding process, indicating areas for further improvement. Figure 2.10 is a brief workflow illustration of UniFolding. There are low-cost alternatives like Low-Cost Robot Arm Project[19], but the folding accuracy is far from satisfactory for those projects.




(a) Picking up top and middle points

(b) Picking up bottom point

(c) Before shaking

(d) Shaking to ensure that the shirt hangs freely

**Figure 2.9** Flattening via Shaking[15]

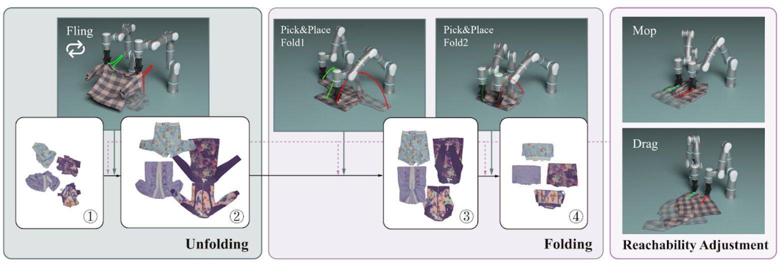



**Figure 2.10** UniFolding Workflow Illustration[17]

# 3 Methodology: Design

## 3.1 Mechanical Design

### 3.1.1 Folding System

**Design**

Based on the improvement of FlipFold[1], we designed a clothing folding plate driven by multiple servos. The specific function is to change several folding plates that initially required manual folding to use servo drives. Although our design does not refer to the product T-Fold[2], the final function and appearance are surprisingly similar. The only difference between our folding board and T-Fold is that our design makes clothes slide down to the next position while T-Fold flips the clothes. Figure 3.1 shows the workflow of the folding system.

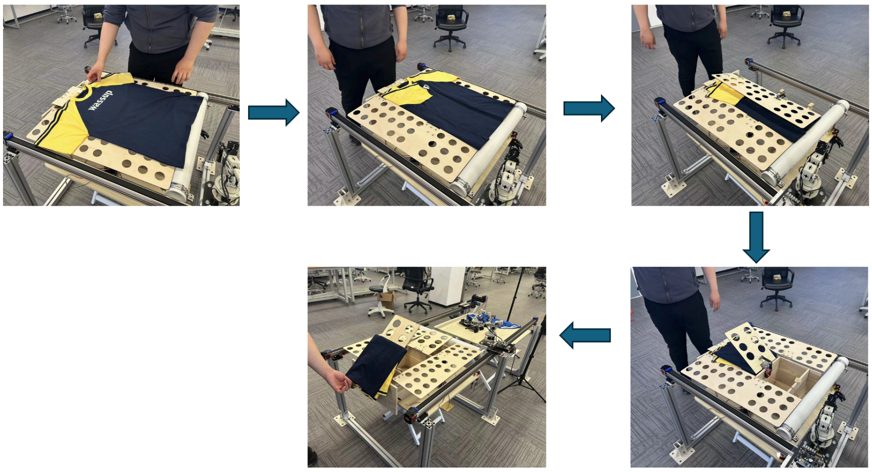


**Figure 3.1** Folding system workflow chart

- **Left Board:** Positioned on the left side, folds 180 degrees to the right, folding the left portion of the clothing.
- **Right Board:** Located on the right side, it folds 180 degrees to the left, mirroring the action of the left core board.
- **Center Lower Board:** Situated below the central part of the clothing, it folds upwards 180 degrees, folding the lower part of the garment.
- **Center Upper Board:** Located above the central part of the clothing, folds about 45 degrees. Rails or clothing-receiving devices will be installed behind this board (depending on the needs of the industrial user). The folded clothes will pass through a slide and slide to the conveyor belt or clothes-receiving device.

To install the four servos that drive the folding board and leave enough space to install the control circuit, Channing, Jialin, and Jiadong designed and manufactured a structure that was completely fixed with wooden boards and screws. The reason for using wooden structures is that wooden structures are easy to process, convenient to install various structures using self-tapping screws, and are very cheap. As Figure 3.2 shows, the price of T-Fold is up to $5995. However, as Table5.1 shows, our wooden board structure with circuits costs less than $500 without human labor. Figure 3.3 is the wooden board that Channing, Jialin, and myself assembled.

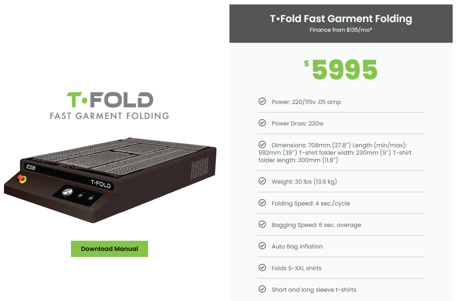



**Figure 3.2** T-Fold Price[2]

**Figure 3.3** Wooden Folding Board

#### Justification

In the literature review, we have mentioned that the folding board structure is widely used in the task of folding clothes, and based on this structure, there are already very mature folding board products on the market, so we will not make a detailed argumentation on the design of the folding system here.

### 3.1.2 Grabbing System

#### Design

#### Components

The grabbing system is designed to grab wrinkled clothes from a clothes recognition region and transport clothes to the folding system. While transporting, the grabbing system should flatten the T-shirt using a mechanical design.

The grabbing system comprises two six-axis robot arms and two transport slides. In the recognition region, vision device No.1 can automatically identify the shoulder of each piece of clothing. Subsequently, the robot arms would be able to accurately grab the clothes' shoulders and transport them to the folding system. Figure 3.5 shows the robot arm connected on the slide. Figure 3.4 shows the connected grabbing system in the laboratory.

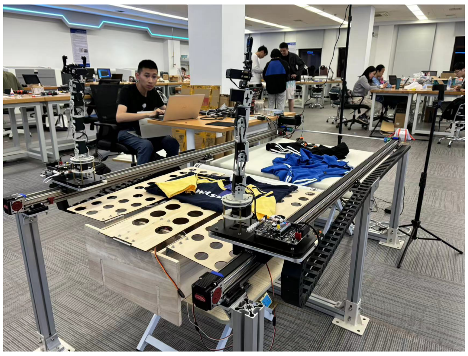


**Figure 3.4** Connected Grabbing System

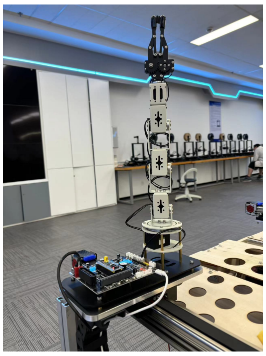


**Figure 3.5** Connected Robot Arm

Inspired by Bell[15] and the UniFolding[17] projects, Channing and Jiadong deployed a long roller device to flatten the T-shirt. The transport track would move back and forth while the robot arms hold

**Figure 3.6** Connected Roller

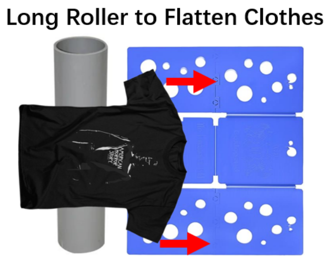



**Figure 3.7** Roller Flatten Illustration

the clothes. The clothes would be flattened with the help of gravity, which is the same idea as the shaking process in Bell's Paper[15]. Experiments have proved that the clothes will remain flat using this mechanism. Figure 3.7 shows how the long roller works during transport.

Jiadong designed and manufactured the connection base between the manipulator and the sliding table to fix the manipulator on the sliding table. Figure 3.8 is the detailed CAD schematic for the connector. Figure 3.9 is the 3-D model illustration of the connector.

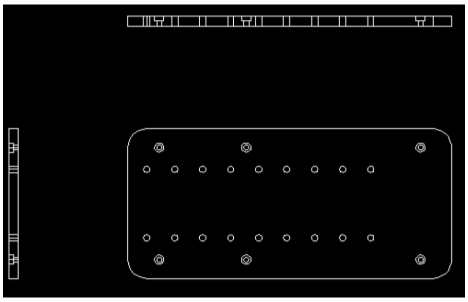


**Figure 3.8** Connector Schematic

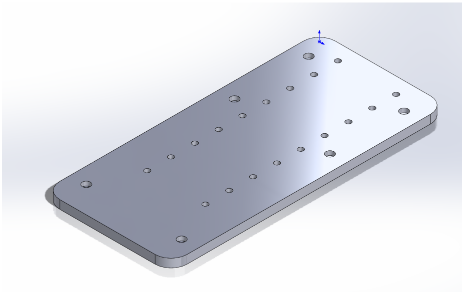


**Figure 3.9** Connector 3-D Model

### Justification

In the literature review, we mentioned that a multiple-arm structure is feasible for flattening clothes and transportation. A fancy example is the UniFolding[17] project from Shanghai Jiao Tong University. They used two robot arm structures for grabbing, flattening, and folding garments. Meanwhile, since the scenario we set is to grab from a recognition region to a folding system, the transport track structure is also an appropriate approach to transport the garments.

## 3.2 Control System Implementation

### 3.2.1 Control Implementation for Robot Arm

#### Introduction with Justification

We want to implement the robot arms' feasibility in locating a specific coordinate in a plane to grab garments through the arms. Consequently, we derive forward and inverse kinematic methods for the robot arms and compare their actual performance with simulations on the software platform. 3.10 shows the actual view of the robot arm, and 3.11 is the physical model of the robot arm.

**Figure 3.10** Six-Axis Manipulator

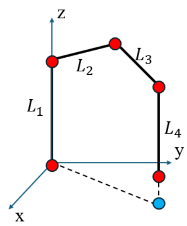



**Figure 3.11** Physical model

Thus, for justifications, We derive some equations to justify our robot arms. Furthermore, we performed some simulations on the software platform and did some tests in real cases to justify that the grabbing system could meet our expectations.

#### Implement Description

We use the inverse kinematic methods to calculate the rotation angle for each servo and compare the actual grabbing point with the correct point. The bias of the length of links, bias of length measurement, and error of computation will finally lead to the offset of the position of the end effector, so we will stimulate grabbing cases and evaluate the error. Four of the six steering engines can determine the position of the end effector. The first engine is used to determine the orientation and other three engines are used to determine the destination from the original point. We notated length of links as $L_1, L_2, L_3, L_4$ and rotation angles for the four steering engine as $\theta_1, \theta_2, \theta_3, \theta_4$. So given coordinate $x, y$ and plane $z$, we calculate $\theta_1, \theta_2, \theta_3, \theta_4$ using equations below.

$$\left(\frac{x}{y}\right) = \tan\theta_1$$

$$z = L_1 + L_2\cos(\theta_2) + L_3\cos(\theta_2+\theta_3) + L_4\cos(\theta_2+\theta_3+\theta_4)$$

$$\sqrt{x^2+y^2} = L_2\sin(\theta_2) + L_3\sin(\theta_2+\theta_3) + L_4\sin(\theta_2+\theta_3+\theta_4)$$

We then write code to test whether the end effector can arrive at the target position and find that the error is within 1cm. Which is acceptable for our task. The code is listed in the appendix.

We simulate the robot arm's performance on the Rviz platform. Figure 3.12 shows the reset state of the robot arm, and the figure 3.13 shows the grabbing state of the robot arms.

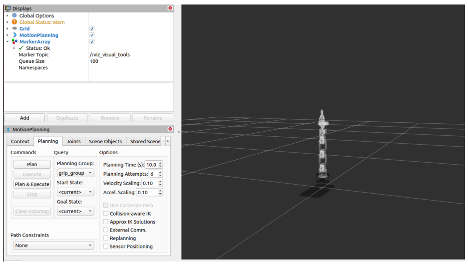


**Figure 3.12** Reset Pose

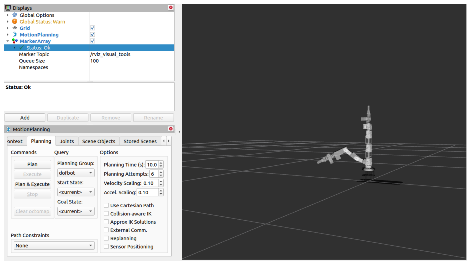


**Figure 3.13** Grabbing Pose

### 3.2.2 Control Implementation for Transport Track

#### Introduction with Justification

The precise control of the transport track mainly relies on the precise control of the servo that drives the track. We must precisely control the robotic arm's position according to our needs. Thus, we implemented the Profile Position Method based on the CANopen serial protocol to control the transport track.

#### Implement Description

**Brief Introduction for Profile Position and CANopen:**

Profile position[20] mode or standard position mode is used for precise point-to-point movements. The servo motor plans the motion path based on the input parameters: target position, profile velocity, profile acceleration, and profile deceleration. Figure 3.14 is the control loop diagram of the Profile Position mode for servo motors.

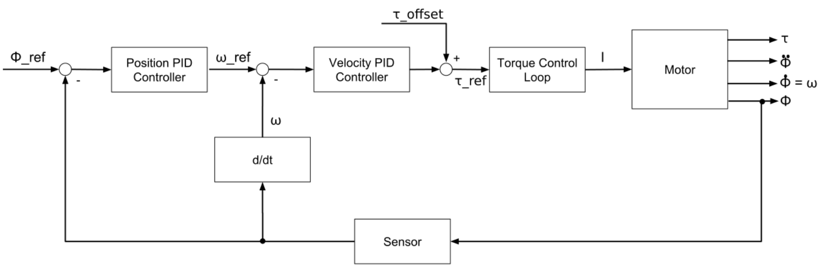



**Figure 3.14** Profile Position Control Loop Diagram[21]

As shown in Figure 3.14, the underlying control approach of the Profile Position method is Proportional-Integral-Derivative (PID) control, a feedback mechanism widely used in industrial control systems. PID control calculates an error value as the difference between a desired setpoint and a measured process variable. The Profile Position method controls the timing of signals sent to each servo actuator, as illustrated in Figure 3.15.

CANopen[22] is a communication protocol used in embedded systems, particularly for automation applications. CANopen facilitates data exchange between devices such as sensors, actuators, and controllers within a network, ensuring interoperability and efficient communication.

The CANopen architecture integrates key components such as the Object Dictionary, Communication Unit, and Network Management to efficiently process and execute control commands issued by a computer[23]. The detailed function of each component is shown as follows:

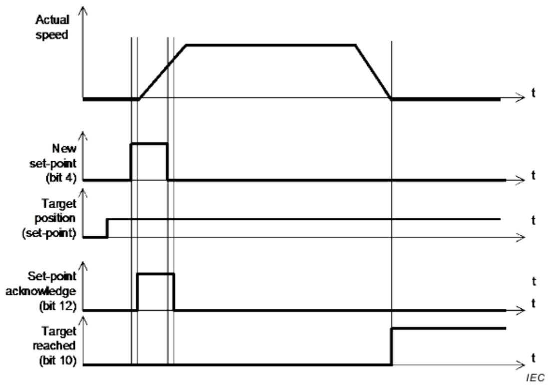



**Figure 3.15** Profile Position Time-domain Signal Diagram[21]

- **Object Dictionary (OD):** Stores configuration and runtime parameters, mapping the control commands to relevant data objects.
- **Communication Unit (CU):** Manages message transmission and reception, ensuring commands conform to the CANopen protocol.
- **Network Management (NMT):** Controls device states, ensuring proper execution of commands by managing transitions between operational states.

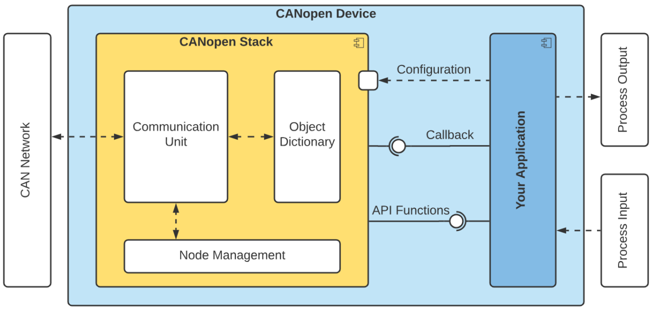



**Figure 3.16** CANopen Protocol Interaction[23]

For instance, in Figure 3.16, a control instruction is the process input into an application; the application will parse the instructions and call API functions in the CANopen stack. OD would store the API call's information and then map the control commands to relevant data objects. CU will transmit the data objects that OD modified to NMT. NMT will control devices to execute motions. Figure 3.17 is the NMT State Machine.

**Implement Description**

According to the introduction of the CANopen control communication protocol and Profile Position control mode, the key to realizing the transport track motion function lies in the following three points:

- Set up the communication between the CANopen hardware and the main control computer.
- Select the appropriate motor, speed, and acceleration to drive the guide rail.
- Implement the Application that can call the CANopen stack.

The following mainly focuses on hardware selection and software implementation. Hardware selection meets the second point of the above requirements and lays the foundation for software implemen-

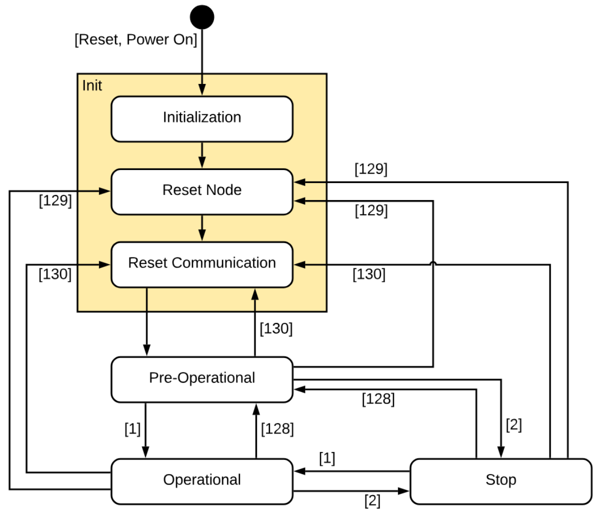
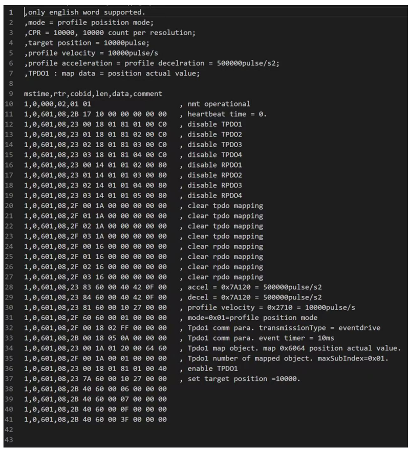



**Figure 3.17** Network Management State Machine[23]

tation. Software implementation solves the first and third requirements.

**For the hardware part**, I chose the FlowCAN CANopen portal extension[24] produced by Flow Servo to set up the communication between the CANopen servo motor and the central control computer. The FolwCAN portal can be used as an NMT management device to send signals by parsing the Datagram from the control computer. Figure 3.18 shows the shape of FlowCAN. Figure 3.19 is an example of a datagram.


**Figure 3.18** FlowCAN USB-CANopen Portal[24]



```
,only english word supported.
,mode = profile poisition mode;
,CPR = 10000, 10000 count per resolution;
,target position = 10000pulse;
,profile velocity = 10000pulse/s
,profile acceleration = profile decelration = 500000pulse/s2;
,TPDO1 : map data = position actual value;

mstime,rtr,cobid,len,data,comment
1,0,000,02,01 01                    , nmt operational
1,0,601,08,2B 17 10 00 00 00 00 00   , heartbeat time = 0.
1,0,601,08,23 00 18 01 81 01 00 C0   , disable TPDO1
1,0,601,08,23 01 18 01 81 02 00 C0   , disable TPDO2
1,0,601,08,23 02 18 01 81 03 00 C0   , disable TPDO3
1,0,601,08,23 03 18 01 81 04 00 C0   , disable TPDO4
1,0,601,08,23 00 14 01 01 02 00 80   , disable RPDO1
1,0,601,08,23 01 14 01 01 03 00 80   , disable RPDO2
1,0,601,08,23 02 14 01 01 04 00 80   , disable RPDO3
1,0,601,08,23 03 14 01 01 05 00 80   , disable RPDO4
1,0,601,08,2F 00 1A 00 00 00 00 00   , clear tpdo mapping
1,0,601,08,2F 01 1A 00 00 00 00 00   , clear tpdo mapping
1,0,601,08,2F 02 1A 00 00 00 00 00   , clear tpdo mapping
1,0,601,08,2F 03 1A 00 00 00 00 00   , clear tpdo mapping
1,0,601,08,2F 00 16 00 00 00 00 00   , clear rpdo mapping
1,0,601,08,2F 01 16 00 00 00 00 00   , clear rpdo mapping
1,0,601,08,2F 02 16 00 00 00 00 00   , clear rpdo mapping
1,0,601,08,2F 03 16 00 00 00 00 00   , clear rpdo mapping
1,0,601,08,23 83 60 00 40 42 0F 00   , accel = 0x7A120 = 500000pulse/s2
1,0,601,08,23 84 60 00 40 42 0F 00   , decel = 0x7A120 = 500000pulse/s2
1,0,601,08,23 81 60 00 10 27 00 00   , profile velocity = 0x2710 = 10000pulse/s
1,0,601,08,2F 60 60 00 01 00 00 00   , mode=0x01=profile position mode
1,0,601,08,2F 00 18 02 FF 00 00 00   , Tpdo1 comm para. transmissionType = eventdrive
1,0,601,08,2B 00 18 05 0A 00 00 00   , Tpdo1 comm para. event timer = 10ms
1,0,601,08,23 00 1A 01 20 00 64 60   , Tpdo1 map object. map 0x6064 position actual value.
1,0,601,08,2F 00 1A 00 01 00 00 00   , Tpdo1 number of mapped object. maxSubIndex=0x01.
1,0,601,08,23 00 18 01 81 01 00 40   , enable TPDO1
1,0,601,08,23 7A 60 00 10 27 00 00   , set target position =10000.
1,0,601,08,2B 40 60 00 06 00 00 00
1,0,601,08,2B 40 60 00 07 00 00 00
1,0,601,08,2B 40 60 00 0F 00 00 00
1,0,601,08,2B 40 60 00 3F 00 00 00
```

**Figure 3.19** FlowCAN Datagram Example[24]

I chose a servo motor controlled by the CANopen bus, produced by FlowServo Company, model FCMT24P60B400W[25]. Figure 3.20 shows the appearance of the servo motor. Figure 3.21 and Figure 3.22 are schematics of the servo motor. As Table 3.1 shows, this motor does support the CANopen control protocol and has a large operating temperature and overload headroom.

**For the software part**, there are two aspects. First is the interconnection and communication between the USB and FlowCAN portal, which is the communication unit component in CANopen. Second, the application layer should be implemented to parse the motion instructions the central control computer gave and translate them to datagrams for the FlowCAN portal.

Thus, the program will first set up the USB serial connection. Then, the program should send the

**Figure 3.20** Servo Motor[25]

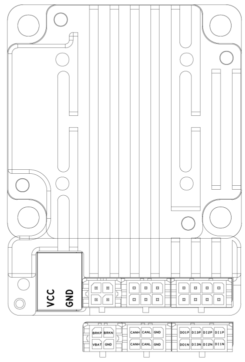



**Figure 3.21** Servo Motor Schematic[25]

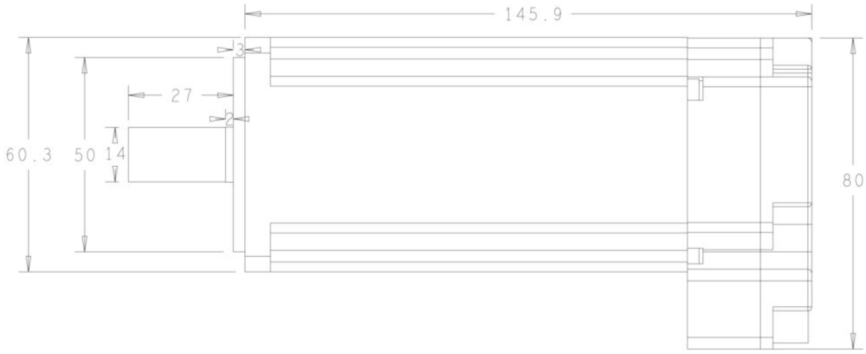



**Figure 3.22** Servo Motor Schematic[25]

| **Model** | FCMT24P60B400W |
|---|---|
| **Type** | Multi-loop control integrated servo motor, built-in driver |
| **Rated Power** | 400W |
| **Rated Voltage** | 24V |
| **Rated Current** | 20A |
| **Rated Torque** | 1.27Nm |
| **Rated Speed** | 3000RPM@24V |
| **Working Voltage** | 24 60V |
| **Working Temperature** | -20°C~50°C |
| **Overload Capacity** | 2.0 times |
| **Communication Interface** | CANOPEN, STEP+DIR |
| **Dimensions** | 60mm flange, total length 146mm |
| **Control Method** | Position loop - Speed loop - Current loop = 20KHz, CANOPEN = 2KHz |

**Table 3.1** Specifications of FCMT24P60B400W Servo Motor[25]

enable operation instruction to the hardware and assign the target position (either relative or absolute), the target velocity, the target acceleration (not necessary), and deceleration (not necessary) to the FlowCAN portal. The program needs to monitor serial port communication and solve the datagram simultaneously. I used PyQt5 for thread management and PyQt5 sub-packages for serial port communication. Figure 3.23 shows how PyQt5 threads work to manage CANopen Profile Position control.

Table 3.2 shows the configuration serial datagram for nodeid=1 in PPM mode for absolute position

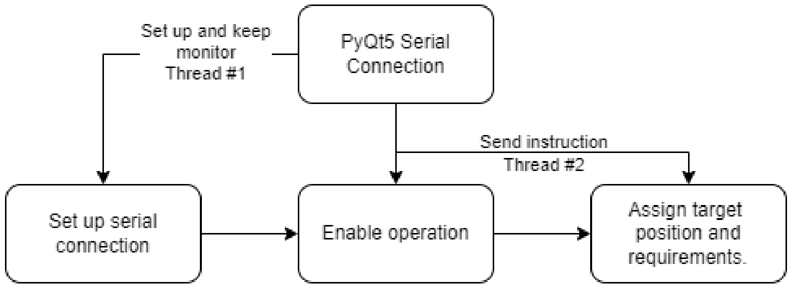



**Figure 3.23** CANopen Profile Position control managed PyQt5 Threads

control, immediately execute, and move to position 10000. Table 3.3 shows the configuration serial datagram for nodeid=1 in PPM mode for relative position control, immediately execute, and move forward 10000 pulses.

| COBID | Data | Description |
|---|---|---|
| 000 | 01 01 | NMT enter operational |
| 601 | 2F 60 60 00 01 00 00 00 | mode=0x01=profile position mode |
| 601 | 23 7A 60 00 10 27 00 00 | target position =0x2710=10000 pulse |
| 601 | 23 81 60 00 20 A1 07 00 | profile velocity =0x2710=500000 pulse/s=3000 RPM |
| 601 | 2B 40 60 00 06 00 00 00 | control word = 0x06 |
| 601 | 2B 40 60 00 0F 00 00 00 | control word = 0x0F |
| 601 | 2B 40 60 00 3F 00 00 00 | control word = 0x3F |

**Table 3.2** Configuration for nodeid=1 in PPM mode for absolute position control, immediately execute, move to position 10000[26]

| COBID | Data | Description |
|---|---|---|
| 000 | 01 01 | NMT enter operational |
| 601 | 2F 60 60 00 01 00 00 00 | mode=0x01=profile position mode |
| 601 | 23 7A 60 00 10 27 00 00 | target position =0x2710=10000 pulse |
| 601 | 23 81 60 00 20 A1 07 00 | profile velocity =0x2710=500000 pulse/s=3000 RPM |
| 601 | 2B 40 60 00 06 00 00 00 | control word = 0x06 |
| 601 | 2B 40 60 00 0F 00 00 00 | control word = 0x0F |
| 601 | 2B 40 60 00 7F 00 00 00 | control word = 0x7F |

**Table 3.3** Configuration for nodeid=1 in PPM mode for relative position control, immediately execute, move forward 10000 pulses[26]

### 3.2.3 Control Implementation for Folding System

**Introduction with Justification**

We designed and manufactured the folding system, and Jiadong designed a PCB based on the PCA9685 chip that could be driven by Arduino. Weijie and Jiadong successfully controlled the movement of the servo through the PCB I designed.

However, due to the serious out-of-synchronization of the self-designed PCB instructions, We switched to the more stable Arduino Sensor Shield to control the servo during later debugging.

**Implement Description**

**Electronic Design: Servo Driver PCB based on PCA9685PW For Folding System**

For folding system control, we use our self-designed PCB SoC based on PCA9685PM LED control chip to execute the motions of the folding board. Figure 3.24 is the wiring between components on the PCB, and Figure 3.25 is the schematic of the PCB.

The principle of this PCB is that the control computer first sends control instructions to Arduino, and then Arduino sends instructions to the PCB to achieve control. During the experiment, we found that the computer instructions and the Arduino were out of sync, and the Arduino and the PCB were out of sync, resulting in serious delays and instability in the entire system. Thus, we switched to using the Sensor Shield[27], the official extension for Arduino, to control the folding board. Figure 3.26 shows the Arduino Sensor Shield pin assignments.

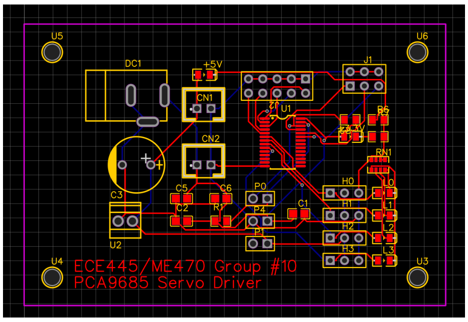



**Figure 3.24** PCB Illustration

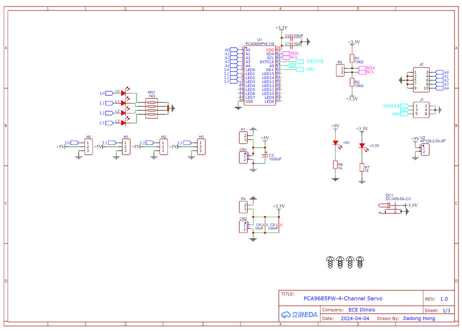


**Figure 3.25** PCB Schematics

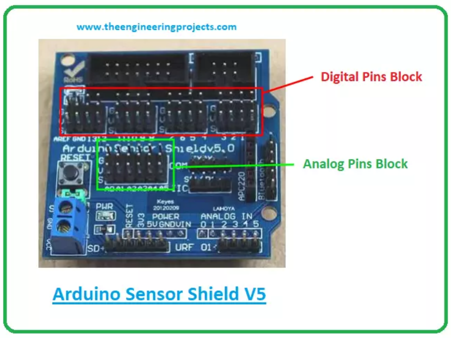



**Figure 3.26** Arduino Sensor Shield[27]

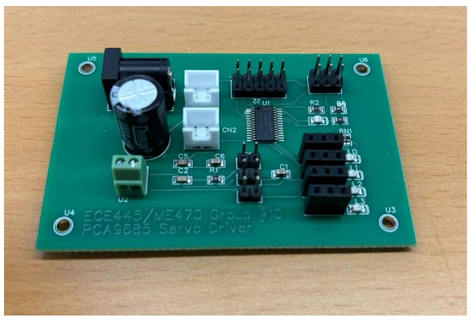


**Figure 3.27** PCB Product

### 3.2.4 Integration of Control System: Fixed-Point Mode

#### Introduction with Justification

We integrated the control of the mutually asynchronous folding and grabbing systems into a single-threaded Python program. By designing a finite state machine, We realized the process of grabbing, transporting, flattening, and folding the shoulders of clothing at specific positions. Figure 3.28 shows how the control computer controls other subsystems.

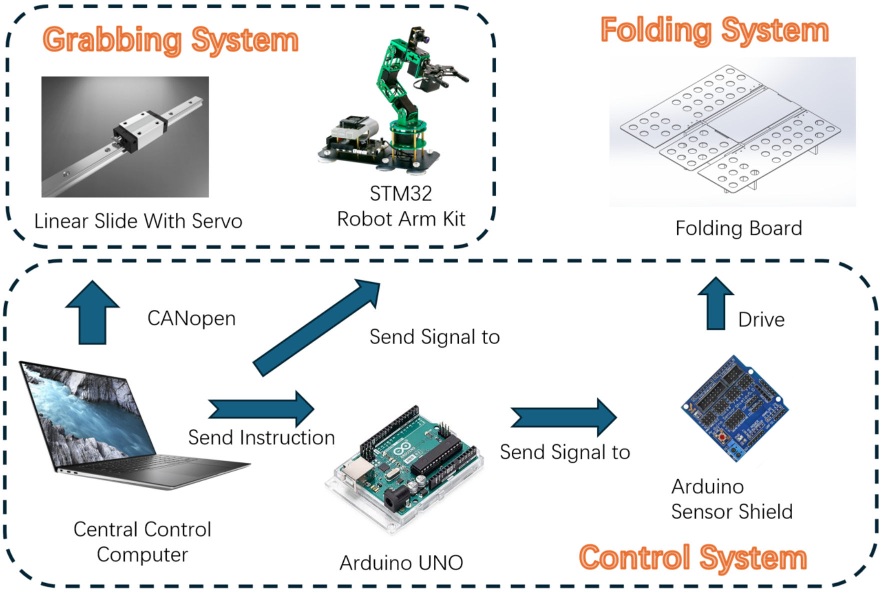



**Figure 3.28** Control System without Vision System

#### Implement Description

We developed a specific mode focused on industrial production for Smart Laundry FoldBot based on finite state machines: fixed point mode. That is to say, no matter how wrinkled or messy the clothes are, as long as the shoulders are at two fixed points, our robot can flatten and fold the clothes through a purely mechanical automated process without relying on vision. Figure 3.29 shows the Finite State Machine of the Fixed Point mode of Smart Laundry FoldBot.

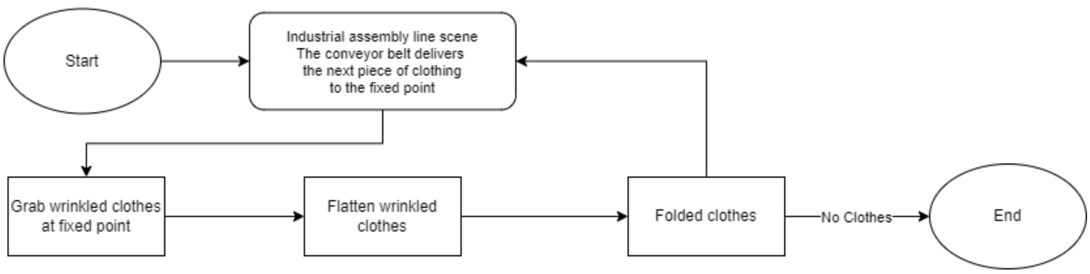



**Figure 3.29** Fixed-Point Mode Control Finite State Machine

## 3.3 Vision System Implementation

### 3.3.1 Implement Description

**Purpose:**

We trained a Mask R-CNN[9] model to precisely identify shoulder positions on shirts. The camera above the table will transmit the image to the vision model. The vision model will identify shoulder positions on shirts and generate an (x,y) coordinate on the plate, enabling the robot arm to grab clothes at that position. Figure3.30 illustrates Mask R-CNN from the original paper.

Mask R-CNN[9] is a deep-learning model used for object detection and instance segmentation. It extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI) in parallel with the existing branch for classification and bounding box regression. This enables Mask R-CNN to precisely detect objects and generate high-quality segmentation masks for each instance in an image.

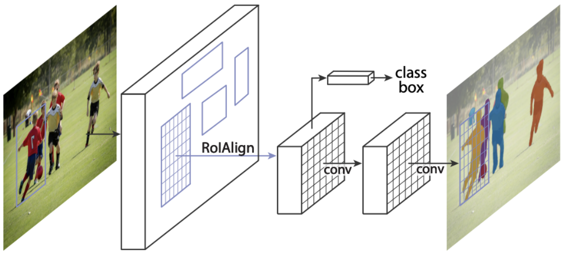



**Figure 3.30** Mask R-CNN Structure Illustration[9]

**Components:**

We use a camera on an aluminum structure to take photos and transmit the images to a PC. We use a laptop PC with NVIDIA GPU as central control and machine learning calculation. The PC will be able to run Fast-RCNN to recognize the grabbing coordinate on the shoulder of the T-shirt.

**Figure 3.31** Structures of hanging camera(Two black camera stands)

### 3.3.2 Justification

In this section, we want to justify that our computer has enough resources to run Fast RCNN In this section, we compare the computing power required by computer vision models with the computing power that the GPU in current control and signal processing systems can provide. We conclude that current devices are fully sufficient to support computer vision inference tasks.

The Fast RNN framework we use is modified based on the VGG-16 computer vision model. So, the computing power required by the computer vision model is determined by the computing power requirements of VGG-16. The industry already has mature literature on the computing power requirements of computer vision tasks. Vassilieva[28] from Hewlett Packard Enterprise made a clear benchmark for the mainstream computer vision models. Our vision system uses a laptop PC with NVIDIA GPU GTX1650ti. There is also theoretical computation power throughput among different technology websites. TechPowerUp[29] made detailed GPU specs on their website.

**Computer Vision Model Requirements**

Figure 3.32 is from the slides made by Vassilieva[28]. The figure shows the detailed computing power needed by mainstream models. The acceleration method used in the model is parallel computing of GPU, which is also the method we used in our vision model. As figure 3.32, the VGG-16 needs 15.5 GFLOPS for the forward pass inference task and needs 528 megabytes cache in GPU memory.

**Popular models**

| Name | Type | Model size (# params) | Model size (MB) | GFLOPs (forward pass) |
|---|---|---|---|---|
| AlexNet | CNN | 60,965,224 | 233 MB | 0.7 |
| GoogleNet | CNN | 6,998,552 | 27 MB | 1.6 |
| VGG-16 | CNN | 138,357,544 | 528 MB | 15.5 |
| VGG-19 | CNN | 143,667,240 | 548 MB | 19.6 |
| ResNet50 | CNN | 25,610,269 | 98 MB | 3.9 |
| ResNet101 | CNN | 44,654,608 | 170 MB | 7.6 |
| ResNet152 | CNN | 60,344,387 | 230 MB | 11.3 |
| Eng Acoustic Model | RNN | 34,678,784 | 132 MB | 0.035 |
| TextCNN | CNN | 151,690 | 0.6 MB | 0.009 |

Hewlett Packard Enterprise

10

**Figure 3.32** Computer Vision Model Computing Power Benchmark

**GPU Specifications**

Figure 3.33 and figure 3.34 are from the website made by TechPowerUp[28]. The figures show the key specifications of GPU GTX1650ti we are using for computer vision tasks. As figure 3.34 shows, the memory of GPU is 4 gigabytes, which is way larger than 512 megabytes. Figure 3.33 shows that even if we use the most precise 64-bit floating point calculation, GPU can still process 95.04 GFLOPS, which is also much larger than 15.5 GFLOPS.

| Theoretical Performance | |
|---|---|
| Pixel Rate: | 47.52 GPixel/s |
| Texture Rate: | 95.04 GTexel/s |
| FP16 (half): | 6.083 TFLOPS (2:1) |
| FP32 (float): | 3.041 TFLOPS |
| FP64 (double): | 95.04 GFLOPS (1:32) |

**Figure 3.33** GTX1650ti Computing Specs

| Memory | |
|---|---|
| Memory Size: | 4 GB |
| Memory Type: | GDDR6 |
| Memory Bus: | 128 bit |
| Bandwidth: | 192.0 GB/s |

**Figure 3.34** GTX1650ti Memory Specs

In addition, it is worth mentioning that we rent a more powerful server for training. So, in this section, we do not consider whether the system can successfully train the model.

## 3.4 Vision-Control System Integration

### 3.4.1 Introduction with Justification

In the visual system, placing the camera precisely on the recognition plane is difficult. Moreover, after the machine learning model infers the position of the clothing and the grasping point, it can only return the target's position on the image. It cannot directly reflect the actual position of the clothing. Therefore, we need to correct the recognition plane and establish a mapping from image coordinates to actual coordinates based on the calibration results so that the mechanical mechanism can work under the guidance of vision.

### 3.4.2 Implement Description

**Convert Computer Vision results to Control Motions**

We used perspective transformation to solve the plane correction problem. We successfully found the transformation function between photo and kinematic coordinates through testing and experiments, achieving accurate vision-assisted mechanical control.

**Difficulties**

There are two main difficulties in converting visual recognition results into mechanical motion coordinates:

The first difficulty is that the photo is not necessarily parallel to the rail and perpendicular to the identification plane. We cannot guarantee the camera position is precise and flat so photos can be taken in good condition. Therefore, the recognition result requires perspective transformation to correct the photo. Figure 3.35 is one of the images in our test dataset during the Computer Vision training process. The camera imaging center plane is not a rectangle, and the camera imaging center axis is not perpendicular to the plane. Figure 3.36 is the visual result of Figure 3.35 after inference.

Our solution is to perform perspective correction first and then automatically convert the control coordinates based on the perspective correction results and the recognition area's inherent size, thereby obtaining the actual position coordinates.

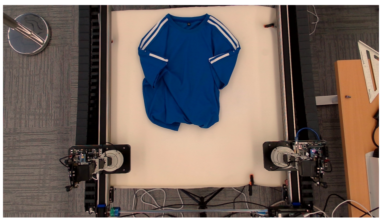


**Figure 3.35** Photo before CV Inference

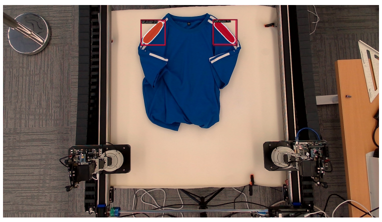


**Figure 3.36** Photo after CV Inference

The second difficulty is that there is more than one grasping strategy to grab the clothes because of our project's redundant degree of freedom. Choosing a grabbing strategy is a key issue.

**Correction using Perspective Transformation**

The key to correcting the plane problem is to restore the deformed rectangular recognition area. Mathematically, the simplest method is to mark the four points of the rectangle and perform a four-point perspective transformation. We need to use two mathematical concepts: the first is perspective transformation[30], and the second is homography.

- *Perspective Transformation*:
  Perspective transformation is a method to convert an image from one viewpoint to another using a $3 \times 3$ matrix. This transformation can correct distortions caused by the camera angle.
- *Homography*:
  Homography is a specific form of perspective transformation describing the mapping between two planes. In image processing, a homography matrix can be determined using four points, which define a plane in the image. The homography transformation can then map this plane to a target plane.

**The detailed Mathematical Formula is shown as follows:** Given four points in the original image $(x_i, y_i)$ and four points in the target image $(x'_i, y'_i)$, the homography matrix $H$ satisfies the following equation:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

The matrix $H$ is a $3 \times 3$ matrix with 8 degrees of freedom (the ninth element is typically set to 1 for normalization).

**Using these concepts, the procedure we should follow is:**

- **Select Four Points**: Select four points in the image, defining the rectangular area to be corrected.
- **Compute Homography Matrix**: Use these four points to compute the homography matrix. This typically involves solving a linear system of equations.
- **Apply Homography Transformation**: Use the computed homography matrix to transform the original planar region in the image to the target plane.

Figure 3.37 shows how the perspective transformation corrects the plane.

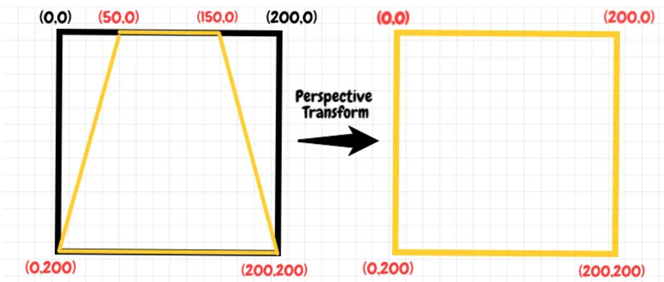



**Figure 3.37** Perspective Transformation Illustration[30]

We selected four feature points on the recognition area plane as correction point input. Figure 3.38 is a sample image using the transformation matrix to instruct the robot arm to reach the specific position. Figure 3.39 shows the grabbing position in the original image.

We used the Python OpenCV library function to implement the perspective transformation because our project does not require a visual correction plane. We only need to know the transformation matrix and convert the distance and the coordinate relationship of the pixels so that our grabbing system can accurately reach the location.

Our project's perspective transformation function is only to correct the plane transformation kinematic coordinates and image coordinates. It will not limit the robot arm grabbing coordinates to the transformed rectangular area. It should be noted that although the corrected image area is relatively small, the executable area for grabbing is larger than Figure 3.38. Coordinates outside the correction rectangle will become negative, but the corresponding motion command coordinates can still be calculated.

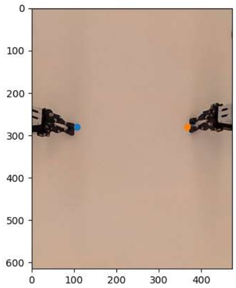



**Figure 3.38** Grabbing Position: Corrected image

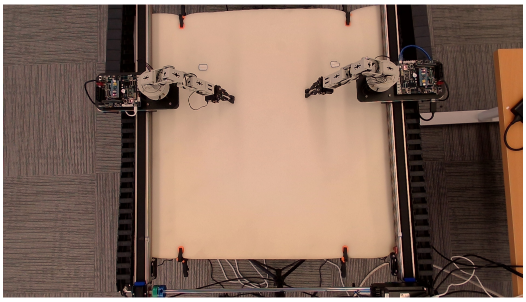


**Figure 3.39** Grabbing Position: Original image

**Integration of Vision System: Computer Vision mode**

**Strategy of Grabbing System**

The grab point strategy for visual model recognition is finding the shoulder position's center point. Given the situation and the coordinate axis shown in Figure 3.40, we assume that the coordinates of the two points are $(x_1, y_1)$, $(x_2, y_2)$ respectively. Then, the guide rail will bring the robotic arm to the position of $(y_1 + y_2)/2$ and then instruct the robotic arm to grab based on the relative coordinates. This aims to solve the problem of clothes not parallel to the x-axis direction and fully use the redundancy of the grabbing system's degrees of freedom.

Due to the limitation of the length of the robotic arm, it is difficult for our machine to grab clothes placed at a large angle with the transport track. Figure 3.41 shows one of the example situations that is hard to handle.

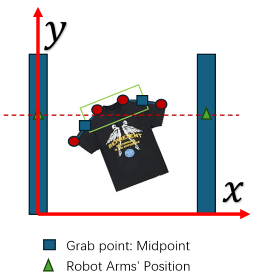



**Figure 3.40** Grabbing Position Illustration

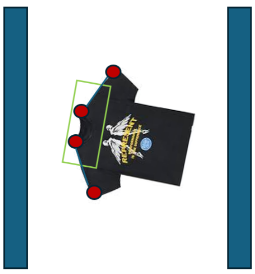


**Figure 3.41** Extreme clothes placing sample

Figure 3.42 is an example of our clothing-grabbing strategy.

**Finite State Machine for Computer Vision Mode**

With the Vision System integrated, the Smart Laundry FoldBot can recognize the T-shirts' status, grab the T-shirts' shoulders, flatten the T-shirts during transport, and finally fold the T-shirts. The mode with the complete function is named computer vision Mode. Figure 3.43 shows how subsystems interact in computer vision mode. Figure 3.44 is the flowchart illustrating how computer vision mode works.

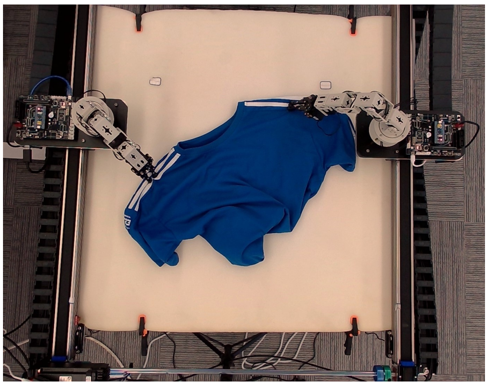


**Figure 3.42** Grabbing Strategy Example

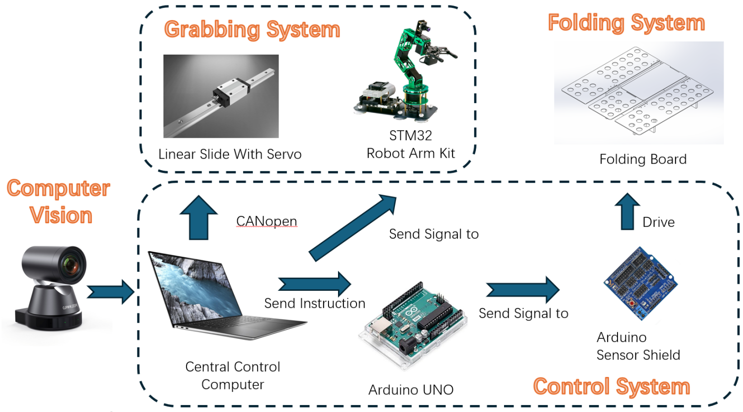



**Figure 3.43** Subsystem Interaction Illustration With Vision System

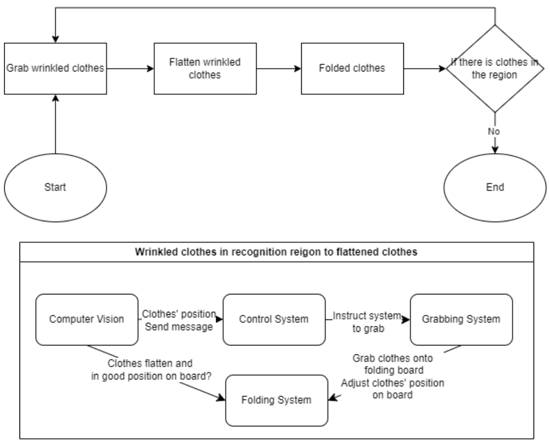



**Figure 3.44** Computer Vision Mode Finite State Machine

# 4 Verification and Results

## 4.1 High-Level Requirements

According to the literature review section, our product combines the advantages of the two projects T-Fold[2] and UniFolding[17]. The success of T-Fold lies in its effective and regular folding method. UniFolding's success lies in its visual identification and flattening strategy.

In our project, we used UniFolding's robotic arm and T-Fold's folding plate and added a transport track to cooperate with the robotic arm for flattening and transportation. Therefore, we put forward high-level requirements for the following three aspects:

- **Grabbing System**: Carefully pick up and flatten the T-shirt and transport it to the folding area.
- **Folding System**: Efficiently fold the T-shirt and transport it to its next destination.
- **Vision and Control System**: Identifies each item of the T-shirts and determines the precise point for grasping.

## 4.2 Verification Tests

In response to the first and second requirements, we developed the Fixed Point mode of the Smart Laundry FoldBot. For the third requirement, we implemented the Computer Vision mode. We will now describe the functional implementation of these two modes, followed by an introduction to the testing methods and test results for the corresponding subsystems.

### 4.2.1 Test 1: Fixed Point Mode Test

We set up two fixed identification points in the identification area. No matter how wrinkled the clothes are, we need to ensure that the shoulders of the clothes can complete the grabbing-flattening-folding operation at the two identification points.

**Test Criteria:**

- 1. The grabbing system stably grabs the clothes from the recognition region as Figure 4.1.
- 2. The grabbing system successfully flattens the clothes and transports them to the folding plate as Figure 4.2 and Figure 4.3.
- 3. The folding plate folds the clothes neatly as Figure 4.4.

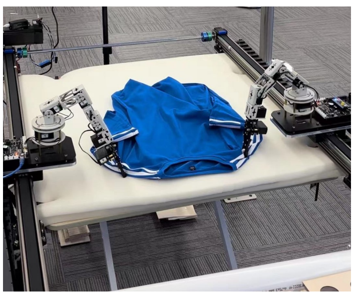


**Figure 4.1** Fixed Point Test: Grab

**Figure 4.2** Fixed Point Test: Flatten

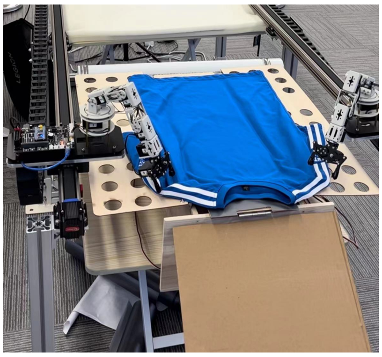


**Figure 4.3** Fixed Point Test: Transport

**Figure 4.4** Fixed Point Test: Fold

**Tolerance and Quantitative Requirements:**

- **Grabbing Stability**: The grabbing system must successfully pick up the clothes 95% of the time. Deviations up to 5% are acceptable due to external factors.
- **Flattening Success**: The flattening mechanism must work correctly in at least 90% of trials. Allowable deviations include minor misalignments not exceeding 2 cm.
- **Folding Accuracy**: The folding mechanism must achieve neat folds with a tolerance of ±2 cm deviation from the ideal fold line in 95% of trials.

### 4.2.2 Test 2: Computer Vision Mode Test

We randomly threw a wrinkled T-shirt on the recognition region, spread the shoulders of the clothes, and then conducted the test. It must be noted that our randomly throwing this test behavior may not necessarily be considered completely random.

It must be pointed out that placing wrinkled clothing in the recognition region is a condition that cannot be completely randomized. The computer vision mode test data can only be used as a loose reference.

**Test Criteria:**

- 1. The vision model can correctly identify the shoulders of the T-shirt as figure4.5.
- 2. The grabbing system can grab the T-shirt based on the coordinates the vision model gives as Figure 4.6.
- 3. The grabbing system successfully flattens the clothes and transports them to the folding plate as Figure 4.2 and Figure 4.3.
- 4. The folding plate folds the clothes neatly as Figure 4.4.

It is worth mentioning that since the computer vision mode only replaces the fixed point in the first part of the fixed point mode with the coordinates input to the control system after calculation by the vision system. Thus, the second and third criteria of fixed point mode are identical to the third and fourth evaluation criteria.

**Tolerance and Quantitative Requirements:**

- **Recognition Accuracy**: The vision system must correctly identify the shoulders of the T-shirt

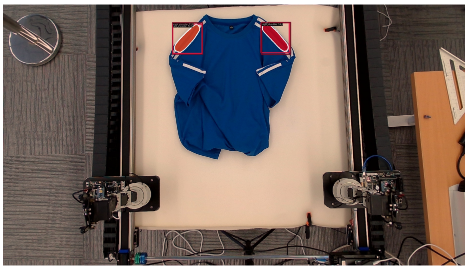


**Figure 4.5** Computer Vision Test: Correct Identify

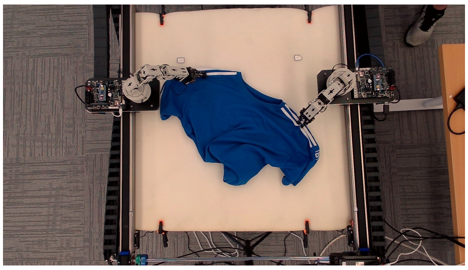


**Figure 4.6** Computer Vision Test: Correct Grab

100% of the time, with acceptable false positives or negatives not exceeding 5%.

- **Grabbing Precision**: The robotic arm must successfully grab the T-shirt based on vision coordinates with 85% accuracy. The tolerance for coordinate deviation is ±1 cm.
- **Transportation Efficiency**: The system must successfully transport and flatten the T-shirt 85% of the time, allowing for minor slips or misalignments up to 2 cm.

## 4.3 Testing Result 1: Fixed Point Mode Test

The fixed point mode showed good performance in the test. We tested the fixed point mode 50 times, and the number of successes was 47 times, with a success rate of 94%.

**Detailed Result with Description**

In the 50 tests, the grabbing system grasped stably 50 times, successfully flattened the clothes, and transported them to the folding board 47 times; based on this, the folding board successfully folded the clothes 47 times.

We can conclude that the crawling success rate is 100%, the flattening success rate is 94%, and the folding success rate is 100%.

The three failed situations were: the clothes fell twice when flattened, and once when the grabbing system transported the clothes to the folding board.

## 4.4 Testing Result 2: Computer Vision Mode Test

Our computer vision mode shows good recognition and grasping performance within the range of motion of the robotic arm. However, due to the limitation of the length of the robotic arm and the imperfect implementation of robotic arm control, our system cannot perfectly capture T-shirts in some extreme placement situations.

In our testing, with clothing randomly placed and shoulders exposed, the computer vision Mode succeeded 17 times out of 25 tests.

**Detailed Result with Description**

In the 25 tests, the vision system accurately and successfully identified T-shirt sleeves 25 times. Based on the accuracy of the recognition results, the number of times the robot arm successfully grabbed and lifted it was 20 times. Out of the twenty times the garment was picked up, it was successfully flattened and transported to the folding board 17 times. The number of successful folds is 17.

We can conclude that the recognition success rate is 100%, the vision-aided grabbing success rate is 80%, and the transport success rate is 85%.

The five grasping tests failed because the visual guidance coordinates exceeded the robotic arm's movement range. Three failures during transportation and flattening occurred because the garment slipped from the mechanical claws.

## 4.5 Test Conclusion

After testing, our system can accurately identify and grab clothes, flatten them, and fold them effectively within the defined tolerances after transporting them to the folding board.

Table **??** is the summary table of our test results.

| **Test Mode** | **Metric** | **Result** |
|---|---|---|
| Fixed Point Mode | Grabbing Stability | 100% (50/50) |
| | Flattening Success | 94% (47/50) |
| | Folding Accuracy | 100% (47/47) |
| Computer Vision Mode | Recognition Accuracy | 100% (25/25) |
| | Grabbing Precision | 80% (20/25) |
| | Transportation Efficiency | 85% (17/20) |

**Table 4.1** Summary of Testing Results

# 5 Costs and Schedule

## 5.1 Cost

We have purchased most of the parts that we need, but since the building and assembly process are not completed, there might be new items that we need to purchase. Table 5.1 shows the estimated cost table for this project.

| Category | Item | Price(CNY) | Quantity | Total |
|---|---|---|---|---|
| Control | XPS15 9500 Laptop(Used) | 2390 | 1 | 2390 |
| Control | Arduino UNO Developer Set | 199 | 3 | 597 |
| Control | 5v 12A DC Power | 35 | 3 | 105 |
| Control | DC 5.5*2.5 Power Plug | 10.5 | 1 | 10.5 |
| Control | RV Power Wire | 19.8 | 2 | 39.6 |
| Control | USB Extension for Laptop | 33.15 | 1 | 33.15 |
| Folding | Foldable Table | 72.9 | 2 | 145.8 |
| Folding | M6 Screw-Nut Set | 7.35 | 1 | 7.35 |
| Folding | Custom Wood Board | 82 | 1 | 82 |
| Folding | L-bar | 5.7 | 1 | 5.7 |
| Folding | M4 Self-tapping screws | 14.04 | 1 | 14.04 |
| Folding | Metal Hinge | 2.89 | 8 | 23.12 |
| Folding | DS3218 Servo | 115 | 4 | 460 |
| Folding | DS3218 Servo Support | 8 | 4 | 32 |
| Folding | Servo Extension Wire | 2.2 | 30 | 66 |
| Folding | Custom Arm for Servo | 90 | 4 | 360 |
| Grabbing | Smart 6-axis Manipulator | 1350 | 2 | 2700 |
| Grabbing | Transport Slides Module | 3000 | 1 | 3000 |
| Grabbing | Support for Slides | 500 | 1 | 500 |
| Grabbing | 400W 24V DC Power | 92 | 1 | 92 |
| Grabbing | Long Roller Tube | 30 | 1 | 30 |
| Grabbing | Roller Stand | 15 | 2 | 30 |
| Grabbing | Clamps (to fix roller) | 6.8 | 4 | 27.2 |
| Grabbing | Guide Rail Chain | 24 | 4 | 96 |
| Grabbing | DC Extension wire (for smart robot arm) | 9.8 | 2 | 19.6 |
| Manpower | Man Power for 4 ZJUI students* | 8000 | 4 | 32000 |
| Vision | Camera | 400 | 2 | 800 |
| Vision | Server Rent | 500 | 1 | 500 |
| Total | | | | 44166.06 |

**Table 5.1** Total Estimated Cost for FoldBot

* Note that we count the manpower as a full-time Teaching Assistance salary for one semester in ZJUI, which is 8000 yuan per semester.

## 5.2 Schedule

| | Week 3/18 | Week 3/25 | Week 4/1 |
|---|---|---|---|
| Grabbing system （hardware) | Mechanical gripper completed design Complete the design of the guide rail | Starting procurement of guide rails | Complete the assembly of the guide rail; Robot arms starting to purchase |
| Folding system (hardware) | Complete the design of the folding board; The design of the folding board drive device has been completed; Folding board base completed design | The folding board base has been manufactured; Folding board completed manufacturing | Purchase of folding board drive device started |
| Control System | Continue to design the folding board control system; Design of grasping system control begins | The same as previous week | The same as previous week |
| Vision Device & ML | / | / | / |
| | | | |
| Qianqi Liu(ME) | Mechanical gripper completed design Complete the design of the guide rail； Complete the design of the folding board; The design of the folding board drive device has been completed; Folding board base completed design | The folding board base has been manufactured; Folding board completed manufacturing | Complete the assembly of the guide rail; Purchase of folding board drive device started |
| Jiadong Hong(EE) | Start component procurement | Starting procurement of guide rails | Robot arms starting to purchase |
| Jialin Shang(ECE) | Continue to design the folding board control system; Design of grasping system control begins | The same as previous week | The same as previous week |
| Weijie Liang (ECE) | Continue to design the folding board control system; Design of grasping system control begins | The same as previous week | The same as previous week |
| | | | |

**Figure 5.1** Schedule

| | Week 4/8 | Week 4/15 | Week 4/22 |
|---|---|---|---|
| Grabbing system （hardware) | Mechanical arm installation completed | The hardware of the grabbing system has been fully assembled | Building aluminum profile brackets |
| Folding system (hardware) | The folding device has been fully assembled (except for the motor) | The folding system has been fully assembled | Building aluminum profile brackets |
| Control System | The control system has basically completed its design and all control systems have been purchased in place | Folding system control works successfully | The guide rail runs successfully; The robotic arm has initially been successfully operated |
| Vision Device & ML | / | / | Start designing visual system |
| | | | |
| Qianqi Liu(ME) | The folding device has been fully assembled (except for the motor) | The hardware of the grabbing system has been fully assembled | Building aluminum profile brackets |
| Jiadong Hong(EE) | Mechanical arm installation completed | The folding system has been fully assembled | Building aluminum profile brackets |
| Jialin Shang(ECE) | The control system has basically completed its design and all control systems have been purchased in place | Folding system control works successfully | Start designing visual system |
| Weijie Liang (ECE) | The control system has basically completed its design and all control systems have been purchased in place | Folding system control works successfully | The guide rail runs successfully; The robotic arm has initially been successfully operated |
| | | | |

**Figure 5.2** Schedule

| | Week 4/29 | Week 5/6 | Week 5/13（Final Pre） |
|---|---|---|---|
| Grabbing system （hardware) | All working properly | Successfully working together with other subsystems | Prepare for Final pre |
| Folding system (hardware) | All working properly | Successfully working together with other subsystems | Prepare for Final pre |
| Control System | The robotic arm can successfully grasp and transport clothes; The folding board can be successfully folded | Successfully working together with other subsystems | Prepare for Final pre |
| Vision Device & ML | Visual system completion training | Successfully working together with other subsystems | Prepare for Final pre |
| | | | Prepare for Final pre |
| Qianqi Liu(ME) | The robotic arm can successfully grasp and transport clothes; The folding board can be successfully folded | Perform final maintenance and testing on mechanical components | Prepare for Final pre |
| Jiadong Hong(EE) | The robotic arm can successfully grasp and transport clothes; The folding board can be successfully folded | Perform final calibration on the guide rail and folding board | Prepare for Final pre |
| Jialin Shang(ECE) | Visual system completion training | Perform final testing and calibration on the mechanical gripper | Prepare for Final pre |
| Weijie Liang (ECE) | Visual system completion training | Perform final testing on the visual system | Prepare for Final pre |

**Figure 5.3** Schedule

# 6 Discussions

In this chapter, we will introduce the limitations of this project and the design reasons for the limitations. We will propose some feasible direction for improvement for this project in the future. We will also introduce what we think are potential solutions to the clothing folding problem. Finally, we will briefly talk about the design solutions we abandoned and the reasons for abandoning them.

## 6.1 Project Limitations

Our project limitations mainly include the following three aspects:

- Our project cannot capture identified clothing exceeding the robotic arms' motion range.
- Our project can only fold T-shirts but cannot fold other kinds of clothes.
- Our project can only solve T-shirts with flat necklines and shoulders.

In the next section, we will describe in detail the design directions of promises for further development in these three aspects. First, for existing projects that can only fold T-shirts, we can design a potential new folding mechanism to fold more types of clothes.

## 6.2 Further Developments

We believe three future work directions can help solve the limitations of our existing projects, including using longer robotic arms with stronger crawls, designing or applying new folding mechanisms to handle more kinds of clothes, and applying more powerful AI agents to handle truly wrinkled clothes.

### 6.2.1 Stronger Robotic Arm

The most direct limitation of our project is the limitation of the robotic arms and the crawls on the arms. The most direct way to improve the project is to use longer robot arms with stronger crawls like those in the UniFolding[17].

### 6.2.2 New Folding Mechanism

Since our current project can only fold T-shirts, one possible direction is redesigning the folding plate mechanism to handle folding other clothes, such as pants and sweatshirts.

The structure of FlipFold[1] supports the folding of various clothes but needs to pre-fold long clothes such as sweatshirts and pants. As Figure 6.1 shows, when folding long-sleeved clothing and pants, we need to fold the parts beyond the folding board into the inside of the board in advance. One potential design solution is to add expanded boards to fold outliers so that sweatshirts and pants can be folded.

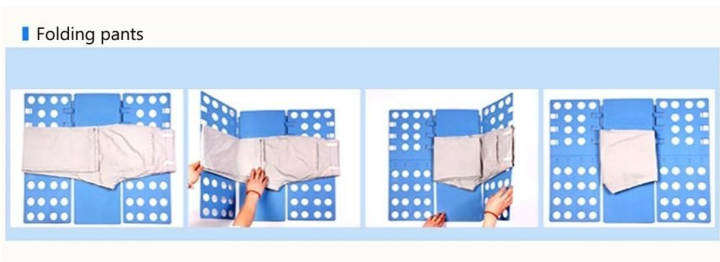



**Figure 6.1** Folding Various Clothes on Folding Board [31]

### 6.2.3 More Powerful AI Agents

Since our project cycle is very short and the schedule is very tight, most of the time is spent on mechanical construction and design modifications. As a result, there is not enough design time for the AI part, and only the functions of computer vision recognition and assisted grabbing of clothes are added.

Most importantly, the data set we trained the model on is limited. With a more complete and larger data set, we can accurately identify clothing in any state and plan suitable grabbing points. Our project's most promising development direction is to use more advanced computer vision perception and recognition skills to flatten wrinkled clothing of any shape.

We previously discussed in the literature review that many reinforcement learning approaches effectively use multi-arm structures to flatten an entire piece of clothing, exemplified by the UniFolding project[17]. AI-enhanced control is another promising avenue. The current grabbing strategy of the robot arms may not always be optimal. We could incorporate more advanced AI agents, such as those from the ALOHA project [14], to develop more robust grabbing strategies and planning.

## 6.3 Alternatives

We mentioned in the literature review that another very effective and already marketed solution is the folding solution that uses complex lever operations, represented by FoldiMate[3]. This type of solution greatly saves the space required by the folding machine. However, this type of solution still requires human participation, and people need to send the clothes smoothly into the FoldiMate conveyor. Our current project is fully automated and does not require human participation. We think using rods for folding is a good direction, but FoldiMate has applied for a series of strict patents for using rods to fold clothes, and the protection has not been lifted yet. The patent "Device, Method, and System for Folding a Moving Article of Clothing"[7] and patent "Fabric article folding machine and method"[6] are two significant patents protecting the rod folding methods and the FoldiMate products.

# 7 Conclusion

## 7.1 General Conclusion

Our project, Smart Laundry FoldBot, successfully combines T-Fold, which folds clothes neatly, with UniFolding, which effectively flattens clothes through visual recognition, automating the entire process from wrinkled clothes to neat folding.

The Smart Laundry FoldBot project also successfully developed an automated solution for industrializing clothes. This system integrates advanced robotics and machine vision technologies to improve efficiency and reduce labor costs in textile management. By effectively sorting, folding, and stacking garments of various sizes and materials, the FoldBot has demonstrated significant advancements over existing technologies, showcasing robust functionality in real-world industrial settings.

## 7.2 Uncertainties

Despite its successes, the project faced challenges, particularly in the precision and adaptability of the robotic arms when handling diverse fabric types and garment conditions. Occasionally, the system's response to unexpected garment presentations underperformed, highlighting the need for further refinement in its sensory and processing capabilities.

The Fixed Point Mode, while achieving a high success rate of 94%, experienced failures in 6% of the trials due to garments falling during the flattening or transportation phases. Specifically, in the 50 tests conducted, the grabbing system grasped 100% of the time stably, but the flattening mechanism failed in 6% of the trials, leading to the garments falling twice when flattened and once during transportation to the folding board. This indicates a need for improved grip mechanisms and stability during movement.

In the Computer Vision Mode, the system demonstrated robust recognition accuracy with a 100% success rate in identifying T-shirt shoulders. However, the grabbing precision was lower, with an 80% success rate (20 out of 25 tests), and transportation efficiency stood at 85% (17 out of 20 successful transports). The reduced performance in these areas was often due to the visual guidance coordinates exceeding the robotic arm' s range of motion and the occasional slippage of garments during transportation. Specifically, 5 grasping tests failed because the visual guidance coordinates were out of the robotic arm' s movement range, and 3 failures during transportation and flattening occurred because the garment slipped from the mechanical claws. These issues underscore the necessity for enhancing the robotic arm's range and control precision and optimizing the grip strength to handle varied garment textures and weights more effectively.

Moreover, the project revealed that extreme placement situations of garments posed significant challenges for the system. These limitations highlight the need for advancements in the robotic arm' s adaptability and control algorithms better to manage a wider array of garment presentations and conditions. Future work should focus on refining the sensory feedback mechanisms and integrating more sophisticated processing algorithms to improve the system' s overall robustness and reliability. Enhancing the robotic arm's control to increase the grabbing precision beyond the current 80% and transportation efficiency above 85% will be crucial for the system's development.

## 7.3 Future Work and Alternatives

**Future Work**

One possible direction of future work is refining the computer vision system to improve garment recognition accuracy under variable operational conditions. Alternative strategies may include the integration of a more sophisticated AI algorithm capable of learning from a broader dataset, thus enhancing the system's adaptability and reliability.

The AI-enhanced control might be another promising direction. The robot arms' grabbing strategy may not be optimal in some cases. In the future, we can introduce more powerful AI Agents, such as agents in the project ALOHA[14], into this project to achieve more robust grabbing strategies and planning.

Additionally, Using a longer robotic arm and grabbing crawls would improve the system. Designing extension boards to handle long-sleeve clothes and pants is also a good way to extend the functionality of this project.

**Alternatives**

There are alternative design directions, such as using rods and transport conveyors to fold the clothes, but this method is under strict patent protection for protecting the product FoldiMate[3]. However, it is still a promising direction for clothes-folding machine design.

***Please refer to the previous chapter, Discussion, for more details.**

## 7.4 Ethical Considerations

In line with the IEEE Code of Ethics, the project was developed strongly committed to social responsibility, ensuring that automation technology augments human labor without displacing jobs. Ethical considerations were prioritized, with rigorous testing to ensure safety and reliability. The project team maintained transparency with all stakeholders about the system's capabilities and ongoing development needs, ensuring that ethical standards in engineering were upheld.

# References


[1] FlipFold. "Filpfold- the ultimate laundry clothes folding tool board." (2011), [Online]. Available: https://flipfold.com/.

[2] ColDesi. "T-fold | electric shirt folding machine - coldesi." (2022), [Online]. Available: https://coldesi.com/t-fold-electric-shirt-folding-machine/.

[3] Foldimate. "Foldimate, the machine that folds your laundry." (2019), [Online]. Available: https://foldimate.website/products/foldimate-the-machine-that-folds-your-laundry.

[4] "Shirt-folding board," US2988251A, Jun. 13, 1961.

[5] "Garment folding apparatus," US6360927B1, Mar. 26, 2002.

[6] "Fabric article folding machine and method," WO2014078106A1, May 22, 2014.

[7] "Device, method, and system for folding a moving article of clothing," US20210079587A1, Mar. 2021.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn*, 2018. arXiv: 1703.06870 [cs.CV].

[10] R. Girshick, *Fast r-cnn*, 2015. arXiv: 1504.08083 [cs.CV].

[11] J. Cychnerski, A. Brzeski, A. Boguszewski, M. Marmolowski, and M. Trojanowicz, "Clothes detection and classification using convolutional neural networks," in *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)*, IEEE, 2017, pp. 1–8.

[12] L. Sun, G. Aragon-Camarasa, P. Cockshott, S. Rogers, and J. P. Siebert, "A heuristic-based approach for flattening wrinkled clothes," in *Towards Autonomous Robotic Systems: 14th Annual Conference, TAROS 2013, Oxford, UK, August 28–30, 2013, Revised Selected Papers 14*, Springer, 2014, pp. 148–160.

[13] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," *Robotics and Autonomous Systems*, vol. 112, pp. 72–83, 2019.

[14] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *arXiv*, 2024.

[15] M. P. Bell, "Flexible object manipulation," Ph.D Dissertation, Dartmouth College, 2010. [Online]. Available: https://digitalcommons.dartmouth.edu/dissertations/28.

[16] J. Van Den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," in *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2011, pp. 409–424.

[17] H. Xue, Y. Li, W. Xu, H. Li, D. Zheng, and C. Lu, "Unifolding: Towards sample-efficient, scalable, and generalizable robotic garment folding," in *Conference on Robot Learning (CoRL)*, Shanghai Jiao Tong University, 2023. [Online]. Available: https://unifolding.robotflow.ai/.

[18] A. Doumanoglou, J. Stria, G. Peleka, *et al.*, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, Oct. 2016. DOI: 10.1109/TRO.2016.2602376.

[19] A. Koch, K. Wu, M. Sidat, I. E. A. Ashimine, A. Arbues, and T.-C. Huang, *Low-cost robot arm*, https://github.com/AlexanderKoch-Koch/low_cost_robot, Accessed: 2024-04-18, 2024.

[20] A. M. Controls, *Position mode control for servo drives*, Accessed: 2024-05-22, 2024. [Online]. Available: https://www.a-m-c.com/experience/technologies/operating-mode/position-mode/.

[21] Synapticon. "Profile position mode - motion control documentation." Accessed: 2024-05-23. (2024), [Online]. Available: https://doc.synapticon.com.cn/software/42/motion_control/operation_modes/profiled_modes/profile_position/index.html.

[22] M. Falch. "Canopen explained - a simple intro." Accessed: 2024-05-21. (2022), [Online]. Available: https://www.csselectronics.com/pages/canopen-tutorial-simple-intro.

[23] CANopen Stack. "Canopen standardized architecture." Accessed: 2024-05-23. (2024), [Online]. Available: https://canopen-stack.org/v4.4/usage/architecture/.

[24] Flow Servo. "Flow servo." Accessed: 2024-05-23. (2024), [Online]. Available: https://www.flowservo.com/.

[25] F. Servo, *Fcmt24p60b400w product page*, Accessed: 2024-05-22, 2024. [Online]. Available: https://www.flowservo.com/servo/fcmt24p60b400w/.

[26] F. Servo, *心流伺服 canopen 手册 a.23*, Accessed: 2024-05-22, 2024. [Online]. Available: https://www.flowservo.com/wp-content/uploads/2024/05/%E5%BF%83%E6%B5%81%E4%BC%BA%E6%9C%8D_CANOPEN%E6%89%8B%E5%86%8C_A.23.pdf.

[27] D. Watson. "Introduction to arduino sensor shield." Accessed: 2024-05-21. (2020), [Online]. Available: https://theengineeringprojects.com/2020/10/introduction-to-arduino-sensor-shield.html.

[28] P. S. R. M. Natalia Vassilieva, "Characterization and benchmarking of deep learning," Hewlett Packard Lab, Tech. Rep., n.d., slides.

[29] TechPowerUp, *Nvidia geforce gtx 1650 ti mobile specs*, https://www.techpowerup.com/gpu-specs/geforce-gtx-1650-ti-mobile.c3517, n.d.

[30] A. Vidhya. "Opencv perspective transformation." Accessed: 2024-05-24. (2020), [Online]. Available: https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143.

[31] Amazon, *Geniusidea folding board*, https://www.amazon.com/Geniusidea-Folding-Clothes-flipfold-Organizer/dp/B07RW31PC6?th=1, Accessed: 2024-05-25, 2023.