

ROBOT LEARNING FOR OBJECT HANDLING IN AN UNSTRUCTURED
ENVIRONMENT

by

Cheng Cao

A thesis submitted to the Graduate College of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Engineering
December 2020

Committee Members:

Chen, Heping, Chair

Stapleton, William A, Co-Chair

Aslan, Semih

Lu, Yijuan

COPYRIGHT

by

Cheng Cao

2020

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

This work is protected by the Copyright Laws of the United States (Public Law 94-553, section 107). Consistent with fair use as defined in the Copyright Laws, brief quotations from this material are allowed with proper acknowledgement. Use of this material for financial gain without the author's express written permission is not allowed.

Duplication Permission

As the copyright holder of this work I, Cheng Cao, authorize duplication of this work, in whole or in part, for educational or scholarly purposes only.

DEDICATION

This thesis is dedicated to my thesis advisors, family and friends, who have been inspiring and supporting me the whole time.

ACKNOWLEDGEMENTS

I would like to thank Texas State University for providing me the opportunity to finish my studies. Texas State University has provided me excellent professors, wonderful library resources and opportunities to learn from peers.

Specifically, first, I would like to express my sincere gratitude to Dr. Heping Chen, my thesis advisor, for giving me the thesis topic, the access to the robotics lab and continuous guidance for my thesis experiment and writing. Dr. Willian Stapleton, Co-Chair of my thesis committee, provided valuable thesis guidance by helping me clarify my ideas and assist my writing. Besides, my sincere thanks to Dr. Semih Aslan for helping me with sentence, grammar and formatting in thesis writing. Dr. Yijuan Lu not only offered wonderful machine learning class, but also took the time to go through the machine learning concepts in the thesis with me. I really appreciate these suggestions that my supportive, knowledgeable and patient committee members provided. My special thank goes to Dr. Vishu Viswanathan, program advisor, as he kindly guided me every step of process through my graduate education; Dr. Hongliang Yu, former visitor scholar of Dr. Chen, helped with my experimental data collection and model validation, which was a long and repetitive task. My thank also extends to the other professors and staff in Texas State University, who gave me suggestions during my thesis time.

The last, I would like to acknowledge the support, understanding and encourage of my family and friends. Without their spiritual help, I would not have been able to survive my thesis period.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	xi
ABSTRACT	xii
 CHAPTER	
I. INTRODUCTION	1
Background	3
Problem Statement	4
Literature Review	7
Machine Vision	7
Machine Learning Algorithms	9
Neural Networks	10
Generalized Regression Neural Network	12
II. METHODOLOGICAL DESIGN	14
Pipe Picking	16
Keyway Alignment	21
Modeling	26
III. EXPERIMENTATION	34
System Configuration	34
Pipe Picking Procedure	34
Summary	42
Keyway Alignment Procedure	44
Data Collection	47
Modeling	49
Experiment Validation Using Predicted Angles	53
Summary	56

IV.CONCLUSION AND FUTURE WORK	57
Conclusion	57
Future Work	58
APPENDIX SECTION	60
REFERENCES	62

LIST OF TABLES

Table	Page
III.1 Data collected (Unit: Degrees).	48
III.2 Different σ of the GRNN model results (Part1)	51
III.3 Different σ of the GRNN model results (Part2)	51
III.4 Result 1 (Unit: Degrees).	55
III.5 Result 2 (Unit: Degrees).	55

LIST OF FIGURES

Figure	Page
I.1 Hunt & Hunt Perforating Gun Machining.	1
I.2 Perforating Gun Outer Pipe and Inner Pipe.	2
I.3 Objects of Experiment.	3
I.4 Current Keyway Alignment Method in Hunt & Hunt.	5
I.5 Proposed Method.	6
I.6 Machine Vision in Manufacturing.	9
I.7 Artificial Neuron Structure.	11
II.1 System Overview.	14
II.2 ABB IRB 4400 Industrial Robot.	15
II.3 Laser Stripe on Pipes.	18
II.4 Screenshot of showing red line on pipes in Cognex Designer.	18
II.5 Pipe Picking Flowchart.	19
II.6 Cognex Laser Displacement Sensor with Robot Gripper.	21
II.7 Keyway Alignment Experiment Setup.	23
II.8 Flowchart of Keyway Recognition.	24
II.9 Keyway and Circle in Keyway Recognition.	25
II.10 Angle Collection Range.	25
II.11 The Detailed GRNN Model.	28
II.12 K-fold Cross-validation.	32
II.13 System Process.	33
III.1 System Configuration.	34

III.2	Pipe Picking Execution Sequence Built in Cognex Designer.	35
III.3	<i>CogAcqFifoTool1</i> Settings.	37
III.4	Image Result from <i>CogAcqFifoTool1</i>	37
III.5	Output Image from <i>CogBlobTool1</i>	38
III.6	Getting the Pipe Point.	39
III.7	Sub-block <i>Cog3DRangeImageHeightCalculatorTool</i> Part Settings. . .	39
III.8	Final Result Outputs.	40
III.9	<i>Image</i> Block with Sub-blocks.	40
III.10	User Interface.	41
III.11	Final Pipe Picking.	43
III.12	The Integrated System Overview during the Experiment.	45
III.13	Keyway Pattern Captured from "FindPatMaxPatterns" Function. . . .	46
III.14	Execution Using Cognex In-sight Explorer.	47
III.15	Measured angle α_i	49
III.16	Measured angle β_i	49
III.17	Distribution of training errors.	52
III.18	Distribution of testing errors.	53
III.19	Predicted Angle from -20 degrees to 20 degrees.	54

LIST OF ABBREVIATIONS

Abbreviation	Description
TCP	tool center point
AI	artificial intelligence
PWC	PricewaterhouseCoopers
MV	machine vision
GigE	Gigabit Ethernet
MVA	Machine Vision Association
SME	Society of Manufacturing Engineers
VGR	vision guided robotics
ANN	Artificial Neural Network
GA	Genetic Algorithm
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
PNN	Probabilistic Neural Network
GRNN	General Regression Neural Network
PDF	Probability Density Function
AOI	area of interest
ERBP	Error Back Propagation
BP	Back Propagation
NN	Neural Network
MSE	Mean Square Error

ABSTRACT

In the oil exploration industry, perforated pipe assembly can be a prolonged process in the manufacturing environment. A pipe keyway must be aligned to successfully assemble the perforated outer and inner pipes. However, the current method uses vision devices to rotate a pipe multiple times, eventually rotating to the angle that meets the requirement, which is time-consuming, leading to a lack of productivity. Therefore, the purpose of conducting this research is to establish an automatic rotating angle correction method such that the keyway can be aligned by only rotating a pipe once.

The system executes a series of processes: recognizing a pipe, picking it up, detecting the keyway, and rotating it to the desired orientation using only a single rotation. A General Regression Neural Network (GRNN) model predicts the actual robot rotation angle needed for correct orientation. The robot will rotate the pipe using the predicted rotation angle. After rotation, the deviation from the desired keyway angle must be less than the given threshold.

This research is of importance in the application of machine vision (MV) in industrial production. In this thesis research, the pipe keyway alignment problem is addressed using a 2D machine vision method as well as the GRNN algorithm. The proposed method is tested using a pipe handling process. A steel pipe with a keyway is to be placed in a random orientation. As the keyway must be aligned in the following manufacturing process, a robot is used to rotate the pipe to the correct orientation by applying the machine learning algorithm. The experiment was set up

and used to test the proposed machine learning method. Also, for easier automatic picking up the pipe, we implemented a 3D machine vision recognition procedure. Compared with the current method, the proposed method allows the robot to only needs to rotate a pipe once to align the keyway. Hence the proposed method can greatly increase the manufacturing efficiency and reduce manufacturing cost.

The thesis introduces the experimental system, explains the theories and the methodologies, describes the procedure of the experiment, and arrives at a result. The system uses an industrial robot ABB IRB 4400; Cognex DS1300 3D Displacement Sensor; Cognex In-sight 7000 2D Smart Camera, and a computer with the GRNN algorithm.

I. INTRODUCTION

Harsh environments, over-sized components, and heavy workloads make some tasks beyond the grasp of unaided human limitations. Robots are designed to accomplish tasks that are beyond these human limitations. Robotics has been widely applied to the manufacturing environment. From PricewaterhouseCoopers (PWC) report, more than half of manufacturers are using robotics technology in some ways [1]. Industrial robots are speeding up operations as well as making core processes cheaper and more intelligent.

A Texas oil and gas perforating gun production and volume machining manufacturer, Hunt & Hunt, is working on improving the automation technology to accelerate the productivity of perforating guns to meet the oil and gas market requirements. An ABB robot is machining perforating guns, as shown in Figure I.1.



Figure I.1: Hunt & Hunt Perforating Gun Machining.

Human workers still manually align the keyway of the perforating gun at perforating gun manufacturers like Hunt & Hunt. The thesis introduces an object handling method to achieve automatic handling and rotating the perforating gun pipes accurately and aligning them using a keyway.



Figure I.2: Perforating Gun Outer Pipe and Inner Pipe.

Figure I.2 shows an outer pipe and an inner pipe of the perforating gun. The charge hole on the inner pipe should be aligned as the same phase as the outer pipe while being assembled so the explosive charges can explode through the thinner hole area on the outer pipe wall. Failure to align the outer and inner pipes would cause perforation failure and low oil production. The methodology described in this thesis is intended to improve this alignment process. The proper positioning of the keyway of the outer pipe is defined as the main focus in this research. To facilitate the complete the process, it required the addition of automatic recognition and "pick-and-place" of pipes. The objects of the experiment involved in the process are shown in Figure I.3.



(a) A pile of steel pipes.



(b) A single pipe with the keyway.

Figure I.3: Objects of Experiment.

Background

Texas is the largest domestic oil-producing area in the continental United States and has had a petroleum industry for more than a century. In recent years, an oil boom has recurred in Texas [2], [3]. With the increasing demand for oil production, improving the processes of the oil industry, which includes exploitation, production, refining, marketing etc. [4], comes back to researchers and businesspeople. In this situation, robotics technology has been applied in the oil industry more than before. According to the data available on Albert B. Alkek library search engine, from the year 2015 to 2019, the Academic Journals and Conference Materials have respectively 155 and 72 robotics and oil industry-related topics. From the year 2009 to 2014, the numbers are 72 and 42, respectively.

Texas State University's robotics laboratory, under the direction of Dr. Heping Chen, attempts to improve the accuracy of robotics control in various manufacturing environments. Some perforating gun manufacturers such as Hunt and Hunt seek a solution for the high productivity as well as good precision of the assembly of perforating gun system.

This presented research employed machine vision, robotics, and a machine-learning algorithm to address the problem mentioned in the first section. In the field of industrial manufacturing, the use of automated robotic assembly lines instead of human workers can significantly cut costs for manufacturing. The complexity of configuring robotics is growing due to the rapid development of automation for hundreds of industries [5]. As an answer to this, by deploying vision sensors in key applications, we can not only increase the industrial robot application's flexibility but also accelerate the rate of application development cycles, reducing the life-cycle costs of a company [6].

Due to this reason, vision-based robotics started playing a more significant role in modern manufacturing. From the 21st century, alliances within the manufacturing field have been searching for methods of the development of automation, promotion of intelligent manufacturing. Intelligent perception, industrial internet, and digital information become the leading trend.

Problem Statement

One of the most common repetitive processes used in manufacturing is object handling. Industrial robots are gradually replacing human workers due to increased efficiency and productivity. However, positional accuracy is a practical consideration in almost every production line. For a specific example, pipe assembly processes require accuracy and the alignment of slotted and threaded pipes. The efficiency

and accuracy of robotic manipulator control techniques are of great importance in such applications.

Hunt & Hunt company is researching using vision sensors to address the alignment problem. The current method is to rotate a pipe several times using a robot, or manually align the keyway, to finally meet the accuracy requirement, which is illustrated in Figure I.4. However, this method is time-consuming and not efficient. This thesis addresses this problem by predicting the keyway rotation angle using a machine learning algorithm. It improves the keyway alignment efficiency.

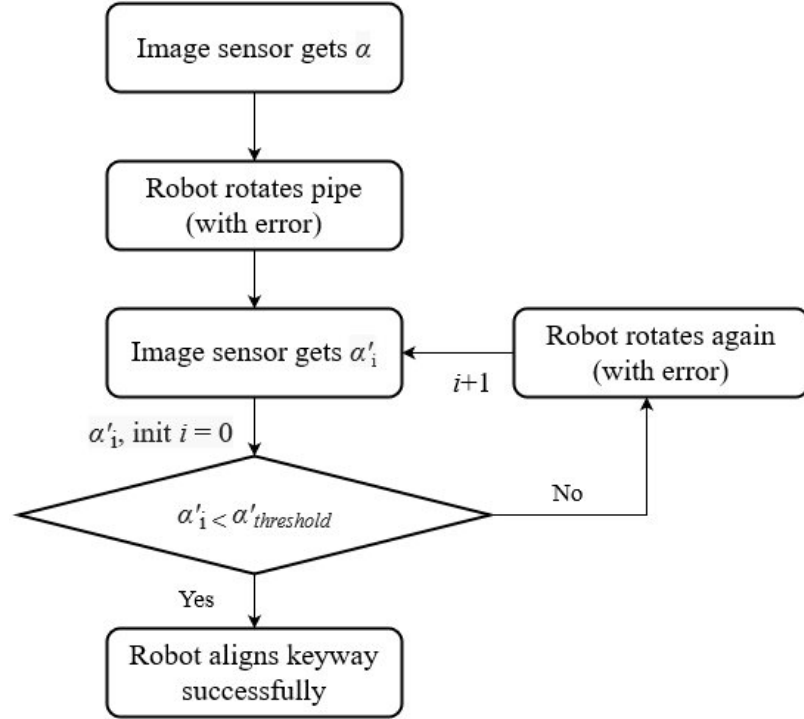


Figure I.4: Current Keyway Alignment Method in Hunt & Hunt.

In the research presented, we proposed a machine learning technique of pipe's pick-rotate-place manipulation using a vision-based robotic system, as shown in Figure I.5.

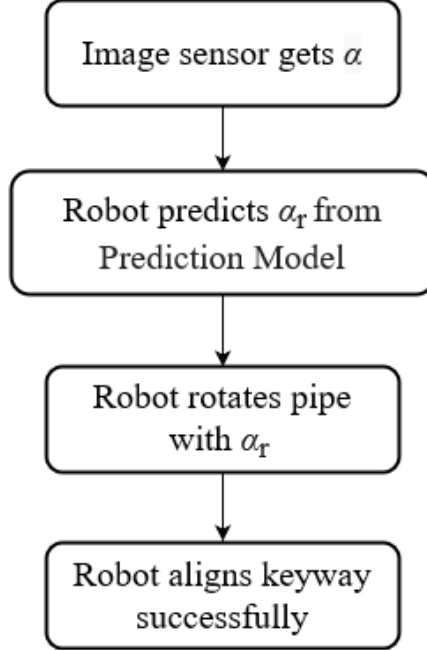


Figure I.5: Proposed Method.

The proposed method will be demonstrated using an industrial robotic arm, a 3D laser displacement sensor, a 2D smart camera, and a computer with a General Regression Neural Network algorithm. The robotic pipe handling process is carried out in two different steps, namely a picking process and an execution process.

The first process automatically locates, picks up, and rotates steel pipes. Without perfect tool center point (TCP) calibrated for this task, the robot end-effector could not precisely rotate the pipe axis based on the detected keyway position and orientation. The goal of this experiment is to solve the problem of a series of the auto-assembly process: from picking up heavy metal pipes to rotating the pipes based on the detected keyway angle, and finally, rotating the pipe using the algorithm-corrected angle to meet the keyway alignment requirement with less than 0.5 degrees of keyway angle error.

Literature Review

There are very few theories about how to align perforating gun pipe keyways with engineering methods. However, the literature covers a wide variety of topics using machine vision. This review will focus on two major themes that repeatedly emerge throughout the literature review, and two sub-themes under the machine learning algorithms. These two themes are: machine vision and machine learning algorithms, and the two sub-themes are: neural networks and generalized regression neural network.

Machine Vision

Machine vision is related to but still distinct from computer vision [7]. Machine vision involves fields including image processing, pattern recognition, signal processing, Opto-electromechanical integration, and so on [8]. Machine vision is concerned with integrated mechanical-optical-electronic-software systems for examining natural objects and materials, human artifacts, and manufacturing processes to detect product defects and improve product quality, operating efficiency, and safety of both products and processes [7]. It is also used to control machines in manufacturing [9]. In simple terms, a machine vision system uses machine instead of human eyes to make various measurements and judgments in manufacturing automation.

The Machine Vision Association (MVA) of the Society of Manufacturing Engineers (SME) describes machine vision as this: "The use of devices for optical non-contact sensing to automatically receive and interpret an image of a real scene in order to obtain information and/or control machines or processes" [10]. Machine vision is the use of devices that are for capturing an image from an industrial camera for inspection or process control of manufactured products.

The usage of machine vision devices can reduce labor costs [11]. Recently, machine vision application is fast-growing with the development of machine learning algorithms, an artificial intelligence branch. It has a huge market with wide prospects and big potential in manufacturing fields. Product measurement, part inspection, recognition, and orientation are typical usages in the area [12]. Machine vision is superior to manual inspection because it can handle high speed and volume production, offers more precise readings and provides better process control [13].

Vision systems include a wide variety of vision sensors. There are several types, including 2D camera, 3D camera. It is also called vision guided robotics (VGR) when it is deployed with the robotics system [14].

The goal of applying machine vision systems is to improve quality and productivity in the manufacturing process. Machine vision can be categorized by functions, such as code reading, print verification, robotic guidance and flaw detection.

Machine vision is one of the critical technologies of robot applications. It is also fast developing with artificial intelligence (AI) technologies. Recently, the rapid development of image processing and pattern recognition technology has greatly promoted the development of machine vision. With AI ability, machine vision highly increases manufacturing productivity. Because many factories have a demand for machine vision systems, it has a huge market with significant potential and broad prospects in the manufacturing filed. The typical deployment in manufacturing is presented in Figure I.6.

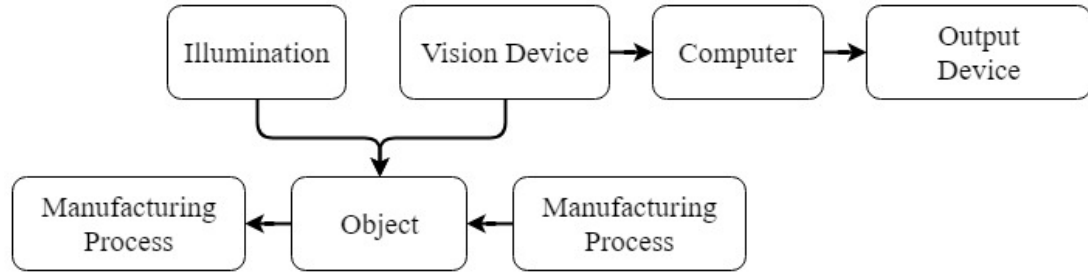


Figure I.6: Machine Vision in Manufacturing.

Machine Learning Algorithms

Machine learning is a multi-disciplinary, interdisciplinary subject and machine learning algorithms do not refer to only a single method, but a combination of many algorithms. It is regarded as the core of artificial intelligence and a fundamental way to make machines intelligent.

It is applied in all fields that required artificial intelligence. Regression and classification are two kinds of problems in machine learning. The main differences are: the predicted output of the regression problem is continuous, while the output of classification is finite discrete values representing different classes.

Specifically, machine learning algorithms attempt to extract implicit rules from a large number of historical data and use them for prediction or classification. More specifically, machine learning can be seen as looking for a function: the input is sample data, and the output is the desired result, but the function is too complex to be easily formalized. It should be noted that the purpose of machine learning is to make the functions learned to apply well to "new samples" rather than just perform well on training samples. Usually, a validation set is used in order to avoid over-fitting [15]. The ability to apply the functions learned to new samples is generalization [16].

Machine learning has been developed for more than twenty years and has penetrated into many fields, such as robotics, genome data analysis, and financial markets.

Neural Networks

Neural Network (NN) or Artificial Neural Network (ANN), is developed from modeling biological neural networks forming with biological brain neurons, cells and other components.

ANNs are designed to imitate the working process of human brains. The construction of a neural network is inspired by the operation of a biological neural network. ANN is a kind of operation model, which consists of a large number of connected nodes. Each node acts as a specific output function, named activation function. The connection between every two nodes represents a weighted value of the signal passing through the connection. Neural network simulates human memory in this way. The network output is determined by structure, connection mode, weight and activation function of the network. The network itself is usually the approximation of some algorithms or functions in nature, or the expression of a logic strategy. ANN combines the knowledge of biological neural network with mathematical model and realizes it with a mathematical tools. Figure I.7 shows the structure of a basic artificial neuron.

The Artificial Neural Network has been applied in the control field for a long time. Liang and Du [17] designed and applied a direct neural network thermal comfort controller for Variable-Air-Volume applications. In the paper, this Artificial Neural Network learned from the user's comfort zone and optimized the system operation to achieve energy savings. Chow and Lin [18] demonstrated a control approach as applying ANN and Genetic Algorithm (GA) integrated optimization

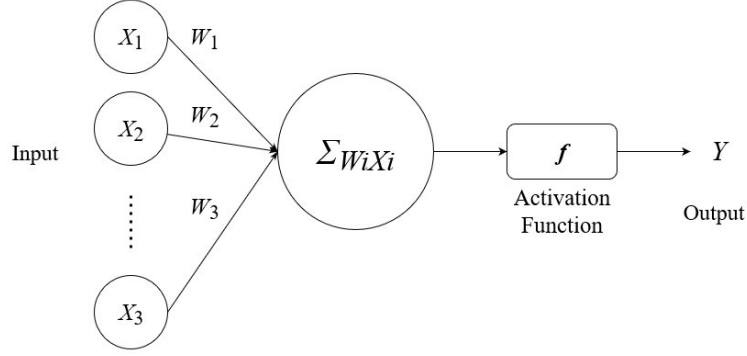


Figure I.7: Artificial Neuron Structure.

methods in the optimal control of an absorption chiller system [19], [20].

Error Back Propagation (ERBP) method was put forward by Rumelhart. in 1986 [21]. It is also known as Back Propagation (BP) algorithm. BP neural network is a feed-forward and multi-layer neural network; it is the typical representative of the artificial neural network and the most widely used neural network [22]. BP neural network has been applied in different fields, such as function approximation, pattern recognition, prediction, classification application automatic control, and data compression research with its unique properties. BP neural network is proven to offer an effective method in these above fields. Until today, the BP algorithm is still the most important and most applied efficient algorithm in automatic control. It is a well-known algorithm for multi-layer neural network training. It has the advantages of a solid theoretical basis, rigorous deduction process, clear physical concept, and strong versatility. However, people find that the BP algorithm has some shortcomings, such as a slow convergence speed and a tendency to fall into a local minimum.

In 1988, based on Powell's Radial Basis Function (RBF) method of multi-variable interpolation [23], Moody and Darken proposed a neural network structure very different from BP neural network, named Radial Basis Function Neural Network

[24].

BP neural network uses a gradient descent method to approximate the minimum error by constantly adjusting the weights of neurons. RBF network is a feed-forward neural network. It does not approximate the minimum error by constantly adjusting the weights. Its excitation function is generally a Gaussian function. Unlike the S-type function of BP, the Gaussian function calculates the weight by the distance between the input and the center of the function.

Generalized Regression Neural Network

In 1991, D. F. Specht first proposed a different neural network structure, named General Regression Neural Network. GRNN is a kind of neural network which uses Probability Density Function (PDF) to replace the pre-determined equation and obtains the regression value of the independent variables and the dependent variables from the samples [25]. GRNN is a variant of RBFNN mentioned previously. From D. F. Specht, GRNN features fast learning that does not require an iterative procedure and a highly parallel structure. It can be used for prediction, modeling, mapping, and interpolating or as a controller. GRNN is similar in form to the Probabilistic Neural Network (PNN), also proposed by D. F. Specht [26].

The advantages of GRNN relative to other nonlinear regression techniques include but are not limited to [27]:

1. GRNN has good local approximation and global optimality;
2. GRNN has faster computing speed;
3. GRNN is easy to use. There is only one parameter, smoothing parameter *Spread* that needs to be adjusted.

GRNN has been implemented in different areas.

In Al-Mahasneh's paper [28], an evolutionary GRNN is developed based on limited incremental evolution and distance-based pruning to online dynamic systems. Also, a variance-based method is suggested to adapt to the smoothing parameter in GRNN for online applications. Kaur [29] portrays a blind audio watermarking scheme in the transform domain using the combination of properties of audio signal extracted through singular value decomposition and general regression neural network leading to the exact extraction of the watermark. A GRNN that can work with measurements without quantization has been evaluated in the paper [30]. The result shows that the proposed method is reliable and effective for condition assessment of transformers through an automated index calculation. In the paper [31], a general regression neural network was applied to predict the pressure loss of Herschel-Buckley drilling fluids in the concentric and eccentric annulus. The predicted pressure losses in annulus using GRNN closely followed the experimental ones with an average relative absolute error less than 6.24%, and a correlation coefficient (R) of 0.99 for pressure loss estimation.

II. METHODOLOGICAL DESIGN

From Chapter 1, there are two different problems to be addressed: automatically picking up perforating gun pipe and automatically aligning pipe keyway. Chapter II illustrates the detailed methods and equipment.

According to the demand, the proposed methodology is described as the following:

- pipe picking process: the robot should pick up the pipes randomly placed or piled on the ground automatically;
- keyway alignment process: rotating the pipe to align the keyway.

The two portions should be linked up by the operation movement of the robot. The proposed robotic working process is shown in Figure II.1.

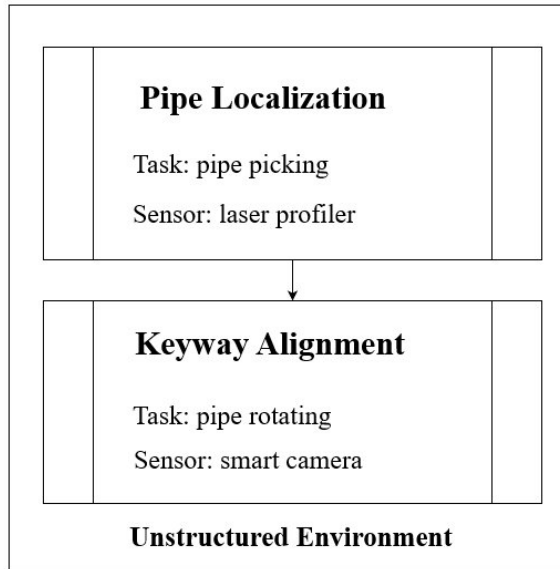


Figure II.1: System Overview.

In this experiment, a similar industrial robot to the one used by Hunt & Hunt was utilized for the research. The industrial robot ABB IRB 4400 system is composed of a manipulator, a controller, as in Figure II.2a and Figure II.2b. The corresponding software is RobotStudio. RobotStudio 6.04 software and RAPID, a robot programming language, RAPID, are used for configuring and editing the commands to automatically control the robot in the experiment. This is a list of the main parts of the robot system:

- An industrial robot ABB IRB 4400 system;
- A gripper as the end-effector of the robot;
- A host computer with RobotStudio;
- Corresponding power supplies, Ethernet cables and other related cables.



(a) Robot Manipulator.



(b) Robot Controller.

Figure II.2: ABB IRB 4400 Industrial Robot.

Pipe Picking

Before running the pipe picking process, there are some preparations for the experiment. On the robot control side, the gripper movement path can be planed in two common ways. In the program design, it shows the actual position on the ABB teach pendant. There are two common ways to program a robot path:

- Teaching method;
- Lead through and offline programming.

This experiment used the "teaching method" by the teach pendant to determine the points for moving. The robot was controlled by pressing buttons and moving a joystick on the teach pendant in the process, so it can move from one precise point to another. The teach pendant is used to save the position information into the robot controller. It is also used off-line programming for the other robot commands.

The selected sensor for locating the pipe is a 3D vision system, Cognex DS1300 Series Laser Displacement Sensor. It can measure the object position. The laser sensor system consists of the following parts [32]:

- A Cognex DS1300 Laser Displacement Sensor;
- A Cognex VC5 Vision Controller;
- A Cognex Designer software that is installed in the VC5.

This pipe picking process is defined as using the robot to pick the located pipe up and place it on a fixed shelf automatically; the pipe should be located by a sensor. The process requires two major tasks. The first is to use the sensing system to locate the pipe, and the second is to move the robot to the current pipe location to relocate

the pipe to the fixed inspection station. The fixed position is used for connecting the keyway alignment process with pipe picking in the experiment setup, which is optional or can be moved after setup. The shelf locates about 5 inches beside the keyway alignment experiment position, and directly faced by the 2D image sensor.

The light source for the pipe picking process is the normal fluorescent lamp light in the lab. In the Cognex system, all the parts are connected using Gigabit Ethernet (GigE), a high-speed transmission Ethernet cable.

For the preparation of the pipe recognition process, the pipe should be put within the working area of interest (AOI) of the laser sensor. It means the pipe must be in the working range or it could cause the issue of the image not generated clearly. The 3D Laser Displacement Sensor oscillates during emission, resulting in a laser scanning line. This line contains thousands of laser points. The Cognex Designer software has a 3D Image Acquisition Wizard for setting up before the image acquisition. In its adjustment tab, the Acquisition Wizard indicates a correct exposure of the reflected laser stripe with red color on the VC5 screen. There are three levels of reflected laser strength defined: red color lines/dots mean the strength of the laser is strong, an adequate exposure time, around 0.05 ms; magenta lines/dots means medium strength reflection; green lines/dots means the strength of the laser is very weak. If the reflected laser stripe is represented as green or magenta, it indicates the view is overexposed, should be readjusted before acquiring clear images. The density of the dots can also indicate the strength level; thus it demonstrates a sense of the depth in image form. Figure II.3 shows the laser scanning the pipes. It is taken by a normal camera under the lab lighting environment. Figure II.4 shows the screenshot of the responding reflection area where the laser strikes on the pipes, taken by Cognex Laser Displacement Sensor and shows in this image Acquisition Wizard in Designer. It shows the object is

within the AOI, so the image acquisition setup of the laser sensor is done.



Figure II.3: Laser Stripe on Pipes.

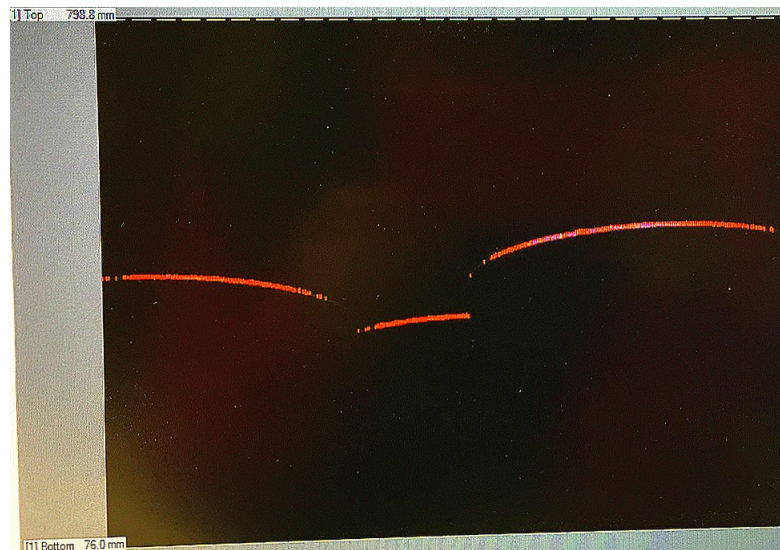


Figure II.4: Screenshot of showing red line on pipes in Cognex Designer.

The coordinates of a point location can be defined in three dimensions as $P_i(x_i, y_i, z_i)$. Images captured from DS1300 Laser Displacement Sensor, are made up of pixels. The relative location information $P_i(x_i, y_i, z_i)$ can be derived from the pixels of the image. The point of the pipe is defined as $P_{sensed}(x_{sensed}, y_{sensed}, z_{sensed})$. The value of $x_{sensed}, y_{sensed}, z_{sensed}$ are measured from DS1300 Laser Displacement Sensor. The pipe picking workflow is shown in Figure II.5.

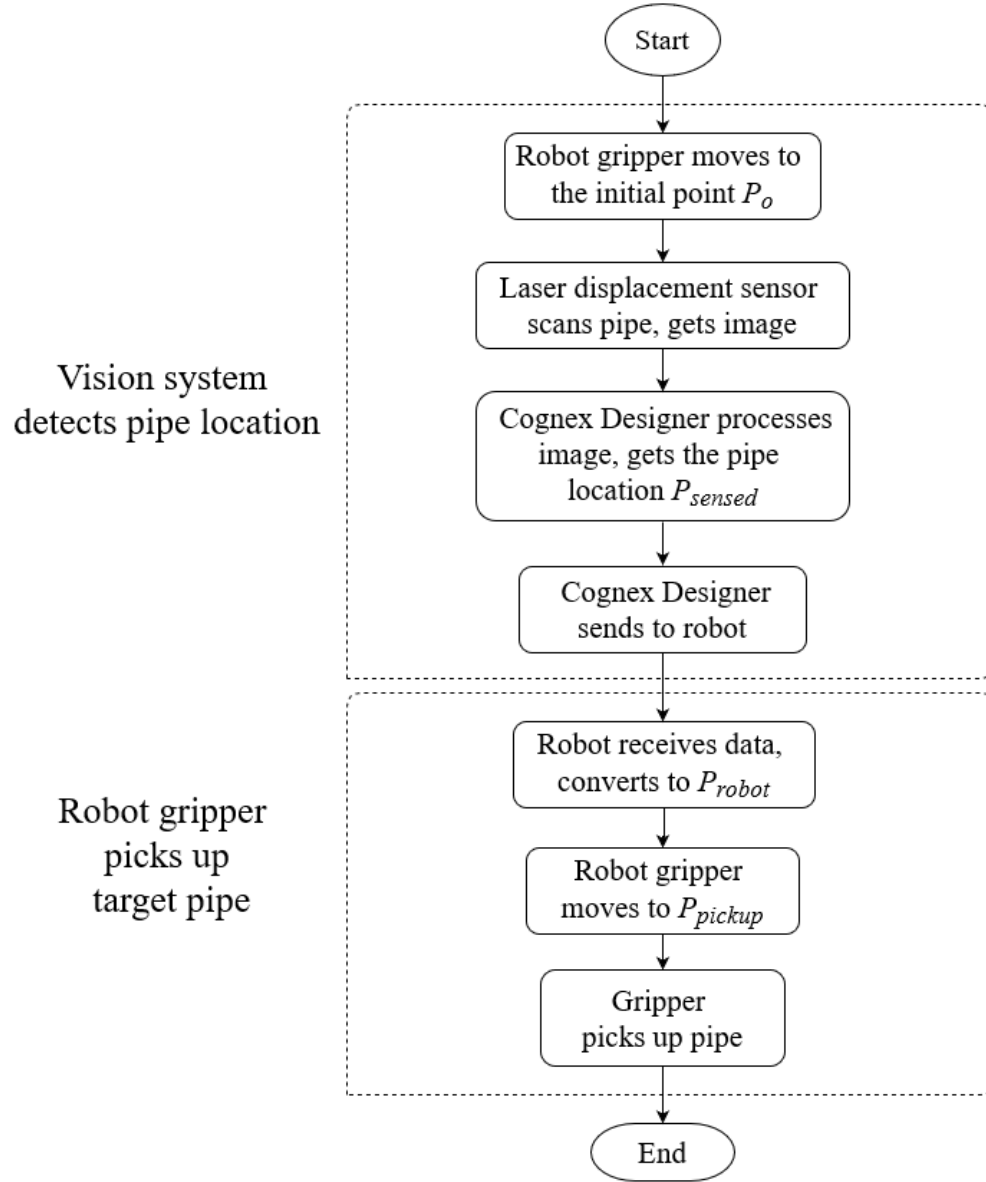


Figure II.5: Pipe Picking Flowchart.

Firstly, the robot gripper with the 3D Laser Displacement Sensor moves to a point $P_o(x_o, y_o, z_o)$, defined as the initial position, then scans the working area. Robot moving is programmed in advance. We build a task sequence in Cognex Designer to accomplish the pipe recognizing, so the robot can move to the pipe location to grab it. The task sequence is makeup by built-in function blocks. For example, the block, *Cog3DRangeImageHeightCalculatorTool1*, measures the height from the laser sensor to the point. The sequence making-up process is elaborated in Chapter III. The pipe position result from the 3D Laser Displacement Sensor is defined as $P_{sensed}(x_{sensed}, y_{sensed}, z_{sensed})$. Planar coordinates of the point (x_{sensed}, y_{sensed}) denote the location in the plane of the floor of the central point of the highest area on top of the pipe. The height from the laser sensor to the pipe defined point is z_{sensed} .

The location of the point sent to the robot needs to be converted to the robot frame. With these coordinates of the point location, the robot gripper moves to the location and picks the pipe up under the robot's commands. The robot gripper picks the pipe up from its location P_{pickup} , and places it on the shelf, then moves it to the keyway alignment location.



Figure II.6: Cognex Laser Displacement Sensor with Robot Gripper.

Keyway Alignment

A 2D image sensor is used for pipe keyway recognition. The 2D image sensor that is qualified should have a quick response time to commands of the robot; also, it should have great accuracy and be well-calibrated such that it can measure the actual angle value. In the research, the 2D image sensor used is a Cognex In-sight

7000 series Industrial Camera. It has a communication module and can connect with Cognex Explorer software installed on a computer. The Cognex Explorer software on the computer transmits the angle data from the camera to the robot.

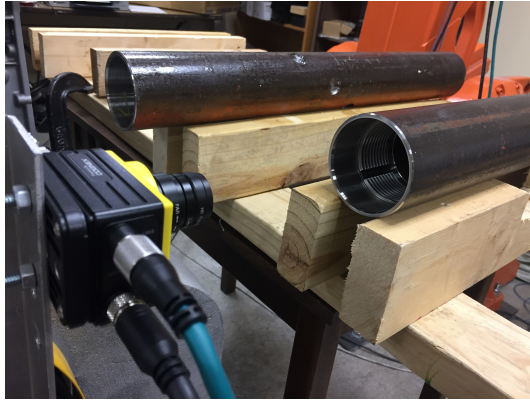
The programmable module of the In-sight 7000 camera allows researchers to develop applications in a spreadsheet. The spreadsheet tool categories include blob, pattern recognition, calibration and image filters, histogram and edges, and so on [33].

The Cognex camera is fitted with a Navitar F1.4/8mm lens to fit the required depth of field, and a StockerYaleM10 High-Frequency Ring Light is chosen for consistent illumination.

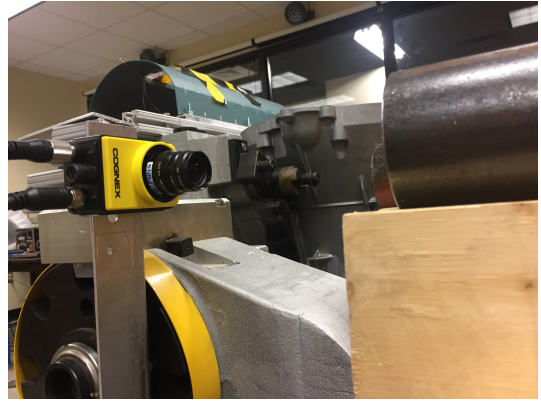
Thus, the 2D image sensor system for measuring the keyway angle contains these main parts:

- A Cognex In-sight 7000 Series Camera and adaptable lens, Navitar F1.4/8mm;
- A Cognex In-sight Explorer software installed on a computer [34];
- A StockerYale M10 High Frequency Ring Light;
- Corresponding power supplies, Ethernet cables and other related cables;

Firstly, as shown in Figure II.7, the lens of the camera is pointing at the pipe cross-section so that the lens, pipe cross-section and the axis of the robot gripper are in a straight line. The focus tuning function of Cognex Explorer is used to adjust the focal length and other parameters of the camera before taking images so that it can be well exposed. Afterwards, we used Cognex In-sight Spreadsheet to implement the image processing and angle recognition. The process workflow is



(a) Pipe and Shelf.



(b) Cognex In-sight Camera.

Figure II.7: Keyway Alignment Experiment Setup.

shown in Figure II.8. Figure II.9 demonstrates the keyway lays inside of a pipe, marked with a red line and a circle to show the detected pattern for measuring the keyway angle. The angle here is 0 degree. This 2D keyway recognition system has been built and tested before the angle data collection experiment. The detailed steps are elaborated in Chapter III.

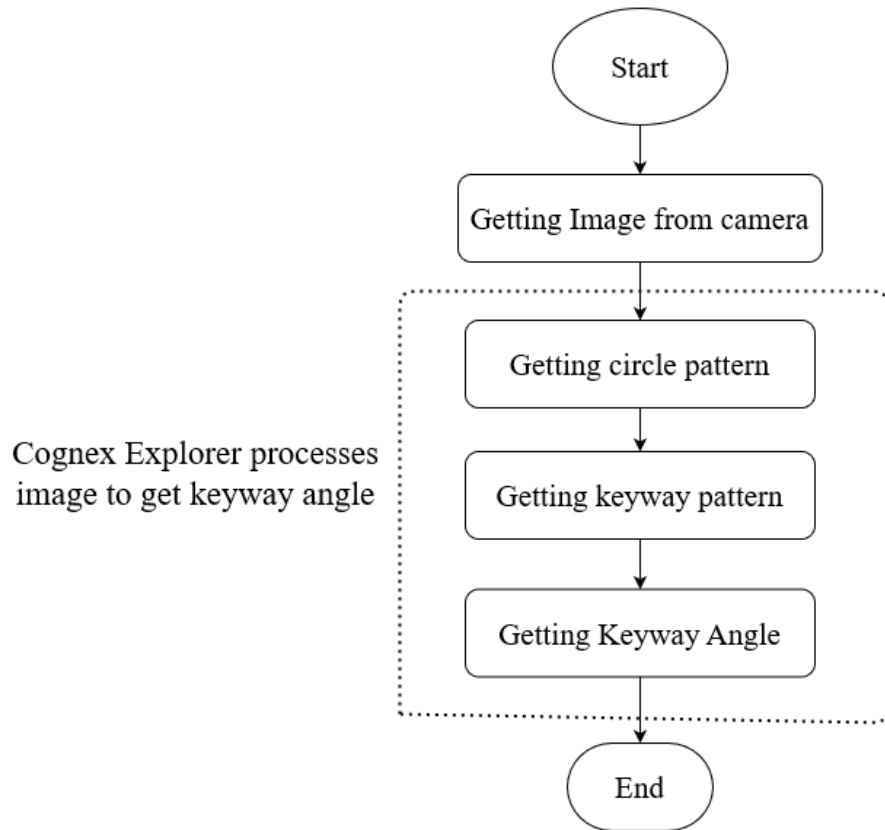


Figure II.8: Flowchart of Keyway Recognition.

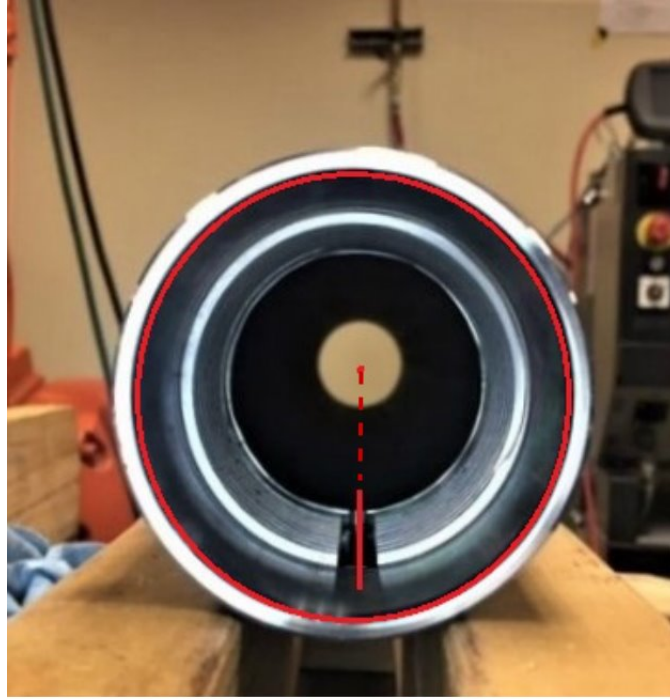


Figure II.9: Keyway and Circle in Keyway Recognition.

The angles from -20° to $+20^\circ$ are considered in the experiment. The range is shown in Figure II.10.

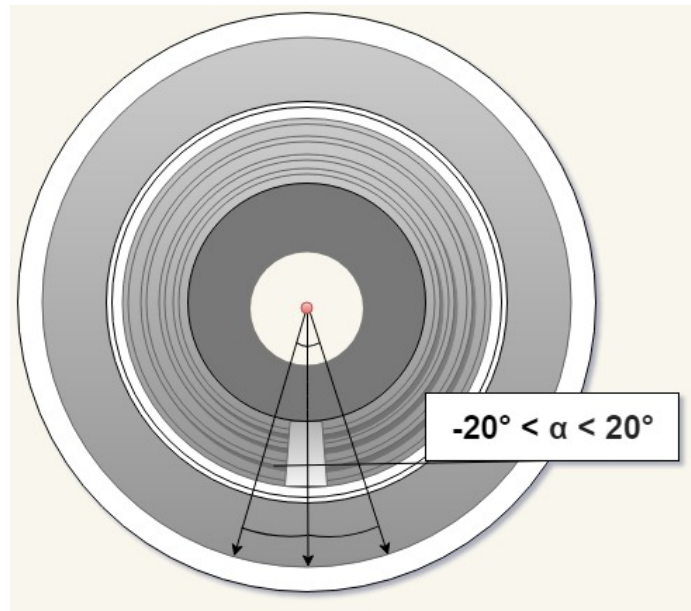


Figure II.10: Angle Collection Range.

Modeling

To further find out the performance of this system, we attempted to collect a series of data from operating the robot. When we tried to get several angle data and rotate the pipe to the baseline, we find the robot rotation has an error. System error is common in robot control area [35], [36]. Our research is to minimize the error.

GRNN PRINCIPLE

The generalized regression neural network is a feed-forward neural network model based upon the theory of nonlinear regression analysis. The regression analysis of dependent variable Y to independent variable X is to calculate the maximum probability value of Y . Assuming that the known joint probability density function of random variables X and Y is $f(x, y)$, the conditional mean of Y given x is defined as:

$$E[Y|x] = \frac{\int_{-\infty}^{\infty} yf(x, y)dy}{\int_{-\infty}^{\infty} f(x, y)dy} \quad (\text{II.1})$$

Where x is the sample observation of measured variable X , \hat{Y} is the Expectation of Y under condition X . When $f(x, y)$ is unknown, $\hat{f}(x, y)$ can be estimated by Parzen nonparametric estimation [37], [38] of the sample observations of $\{x_i, y_i\}$. The probability estimator $\hat{f}(x, y)$ can be expressed as

$$\hat{f}(x, y) = \frac{1}{(2\pi)^{(m+1)/2}\sigma^{(m+1)}} \cdot \frac{1}{n} \sum_{i=1}^n \exp\left[-\frac{(x - x_i)^T(x - x_i)}{2\sigma^2}\right] \cdot \exp\left[-\frac{(y - y_i)^2}{2\sigma^2}\right] \quad (\text{II.2})$$

Where n is the number of sample observations, m is the dimension of X (in this experiment, m equals 1) and σ is the Gaussian kernel width. The σ is also named as

the standard deviation of the PDF, but in GRNN, the σ is called smooth parameter. The estimated probability equation for $\hat{f}(x, y)$ is derived at point (x, y) by a measurement comparing that point with each of the neural network training cases (x_i, y_i) leading to Equation II.2.

A new expression for $\hat{Y}(x)$ can be written as Equation II.3 when using $\hat{f}(x, y)$ as an estimate of the probability distribution, which is represented as:

$$\hat{Y}(x) = \frac{\sum_{i=1}^n \exp\left[-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} y \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy}{\sum_{i=1}^n \exp\left[-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy} \quad (\text{II.3})$$

The final network output is obtained using Equation (II.4). $\hat{Y}(x)$ is the weighted sum of the dependent variable y_i of all the training samples. A detailed derivation can be found in [25].

$$\hat{Y}(x) = \frac{\sum_{i=1}^n y_i \exp\left[-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right]}{\sum_{i=1}^n \exp\left[-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right]} \quad (\text{II.4})$$

GRNN STRUCTURE

The General Regression Neural Network implemented here has four layers in the network structure. The four layers are: input layer, pattern layer, summation layer and output layer. The structure of a GRNN depends on the number and type of training samples. In addition, the smoothing parameter *Spread*, also noted as σ , is the primary factor that affects the generated output of the network. The GRNN structure is shown in Figure II.11.

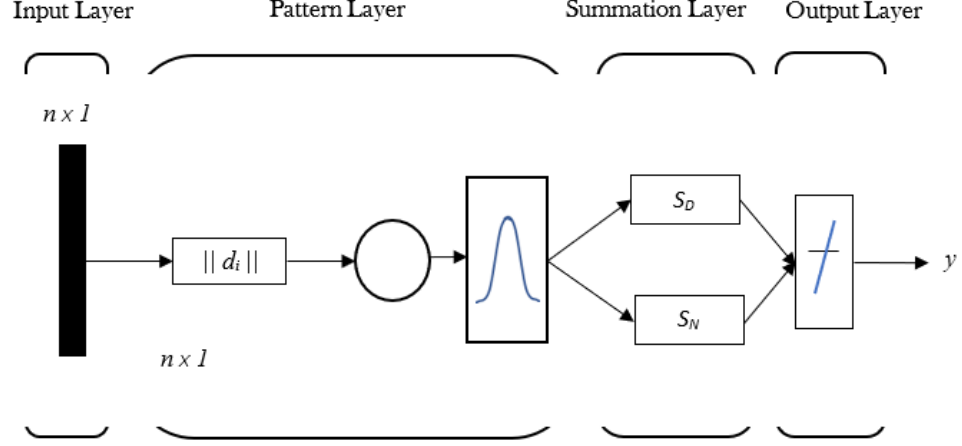


Figure II.11: The Detailed GRNN Model.

Input Layer

The number of neurons in input layer equals the dimension of the input. In this case, the input vector consists of only the measured angle value for the pipe keyway. Each neuron is a simple distribution unit, which directly transfers the input variables to the pattern layer.

Pattern Layer

The number of neurons in pattern layer equals the number of learning samples n . The training process for the neural network assigns weighting factors to each neuron corresponding to the different training data samples. The activation function of neurons in pattern layer is the Gaussian kernel function, p_i ,

$$p_i = \exp^{-\frac{d_i^2}{2\sigma^2}}, i = 1, 2, \dots, n \quad (\text{II.5})$$

d_i is the Euclidean distance between the network input variable α and the i^{th} training sample.

$$d_i = \sqrt{(\alpha - \alpha_i)^2}, i = 1, 2, \dots, n \quad (\text{II.6})$$

The output of the pattern layer is the exponential square of the Euclidean distance d_i .

Summation Layer

From D. F. Specht, two kinds of neurons are in summation layer. One is Numerator S_N , which is represented as

$$S_N = \sum_{i=1}^n \beta_i p_i \quad (\text{II.7})$$

Where β_i is the i^{th} training output. Numerator contains the summation of the multiplication of activation function and the training output. Another one is Denominator, S_D

$$S_D = \sum_{i=1}^n p_i \quad (\text{II.8})$$

Denominator contains the summation of all activation function. Summation layer feeds both the Numerator and Denominator to the output layer.

Output Layer

The output of the output layer is the Numerator node divided by the Denominator node.

$$y = \frac{S_N}{S_D} \quad (\text{II.9})$$

GRNN IMPLEMENTATION

MATLAB is used as the main tool for data analysis and modeling. MATLAB is a common calculation software widely used in engineering fields such as automatic control, mechanical design, and mathematical statistics[39]. The neural network toolbox covers most basic commonly used NN models, including GRNN functions, *newgrnn*[40][41]. The detailed steps will be explained in Chapter III.

Spread Selection

The controlling parameter to be determined in GRNN is the *Spread* σ . Measures of Mean Square Error (MSE) at various σ are used to select a value of σ , which produces results within the desired error tolerance. We only need to set the range of *Spread* and the standard of selecting *Spread*. The definition of MSE is:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \beta_i)^2 \quad (\text{II.10})$$

We also used Cross-validation method to find the training set leading to the best MSE value.

Cross-validation

Cross-validation is a validation method in statistical analysis for both regression and classification problems. Cross-validation splits the whole dataset into several subsets. The Cross-validation uses some subset(s) for training and the remaining subset(s) for testing the resulting model.

The selected cross-validation model for this effort is called k -fold Cross-validation, also called rotation estimation[42]. For k -fold Cross-validation, the total training set is randomly selected into k subsets. Then, $k - 1$ subsets are used for training; the remaining subset is used for validation to generate an MSE value. The process is repeated rotating which subset is reserved for testing until all combinations have been tried. This is illustrated in Figure II.12. The model being developed begins with 90 training sets of data and 4-fold Cross-validation with subsets of 23 training sets is used to create the final model. The equation of Overall Cross-validation MSE can be written as

$$MSE_{cv} = \frac{1}{k} \sum_{i=1}^k MSE_i \quad (\text{II.11})$$



Figure II.12: K-fold Cross-validation.

The principal advantages of the GRNN are fast learning and convergence to the optimal regression surface as the number of samples becomes large. It is particularly advantageous with sparse data in a real-time environment because the regression surface is instantly defined everywhere, even with just one sample [43]. The success of the GRNN method depends heavily on the *Spread* factors[25]. The larger *Spread*, the smoother function approximation. The *Spread* values obtaining a minimum of MSE for the test period is considered.

The overall experimental procedure is shown in Figure II.13:

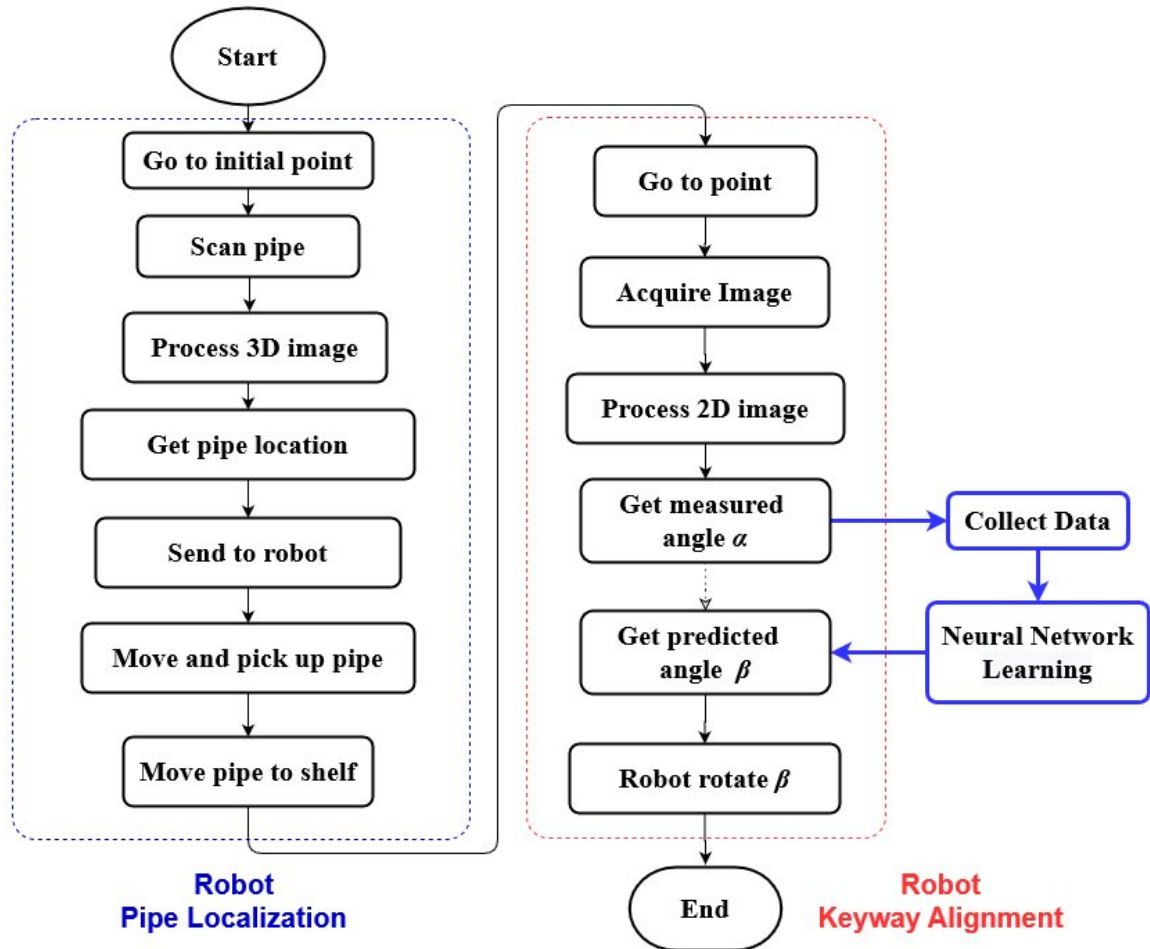


Figure II.13: System Process.

III. EXPERIMENTATION

This chapter includes system configuration and experimental procedures. The detailed procedure and experimental results will be provided.

System Configuration

An overview of the entire system configuration design is shown in Figure III.1:

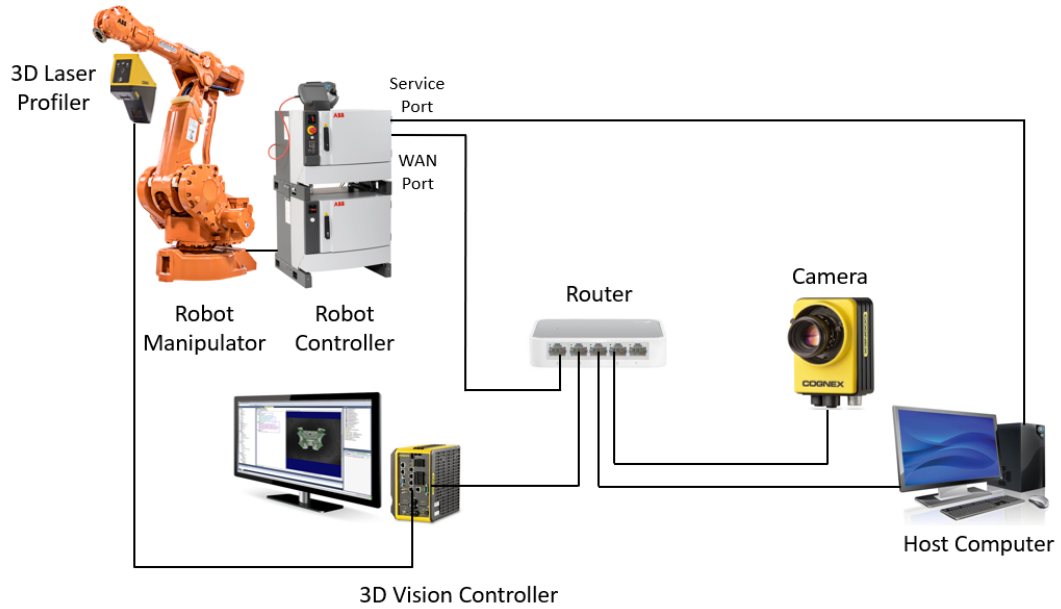


Figure III.1: System Configuration.

Pipe Picking Procedure

In the pipe picking process, the computation device used to capture the 3D image information is Cognex Vision Controller, VC5 with Cognex Designer installed. Cognex Designer does functions as, acquisition, selecting images, optimizing vision effect, and interfacing with the robot.

As mentioned in Chapter II, the pipe picking process should use Cognex DS1300 Laser Displacement Sensor to acquire the picking location of the target pipe. The actual operation is designed and executed in the software, Cognex Designer.

Prior to using Cognex Designer, the first step is connecting devices. Cognex Designer has connection configuration for Cognex DS1300 laser sensor, vision controller VC5 and ABB robot. The communication of DS1300 Laser Displacement Sensor and Vision Controller VC5 is using GigE standard cable. GigE Vision Configuration Tool, a software tool for Cognex Designer.

Once the hardware is set up, the next step is to set up the device connection in the software. In the Designer, there is a section called Devices List, "connect DS1300" choice can be found under "Camera". Clicking it to add DS1300 to the Task Sequence. Under the project environment of Designer, tasks were organized as an execution sequence and displayed as connected block sequence. The task sequence receives a command signal from the robot, executes locating a pipe, and sends back the location of the pipe for the robot to pick up. The finished Task Sequence under Cognex Designer is shown in Figure III.2.

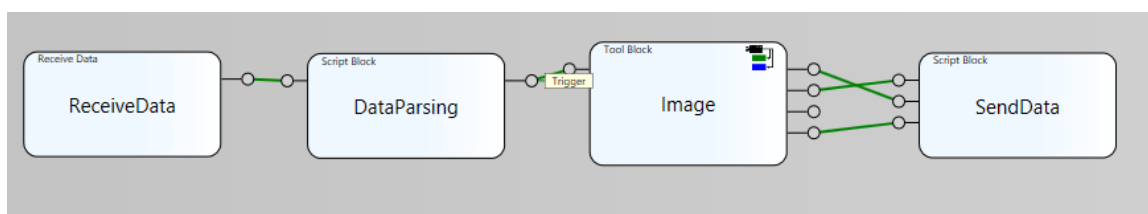


Figure III.2: Pipe Picking Execution Sequence Built in Cognex Designer.

The sequence comprises of four general blocks, which are

1. *ReceiveData*, 2. *DataParsing*, 3. *Image*, and 4. *SendData*. *ReceiveData*, *DataParsing* and *SendData* are single layer blocks, while *Image* block has multiple sub-blocks under it. Each block plays a role as well as has its input and output.

Block 1. ReceiveData

This is the interface block that manages the function of connecting with outside devices. The setup includes selecting the signal type, device source, and which transmission method to receive the signal data. Here we use to receive a String signal from the ABB robot.

Block 2. DataParsing

DataParsing is a script block simply connecting to the next tool block *Image* from the last block. The block converts String signal to Boolean signal.

Block 3. Image

This block is the main block. It contains a series of sub-blocks to configure the image processing procedure for locating the pipes. The scanned image result is from the function, *CogAcqFifoTool1*, which is to acquire image from the DS1300 laser sensor. Figure III.3 is a screenshot of part of the settings for the image acquisition. The image gotten from DS1300 sensor is in Figure III.4.

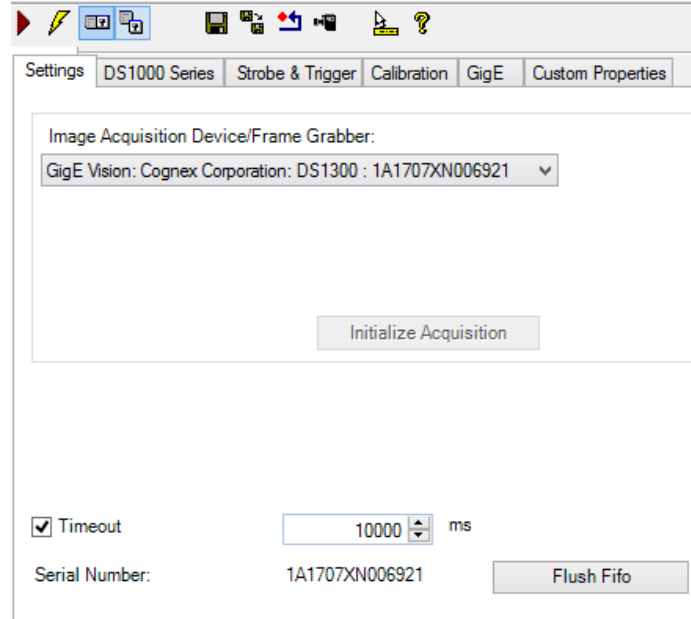


Figure III.3: *CogAcqFifoTool1* Settings.

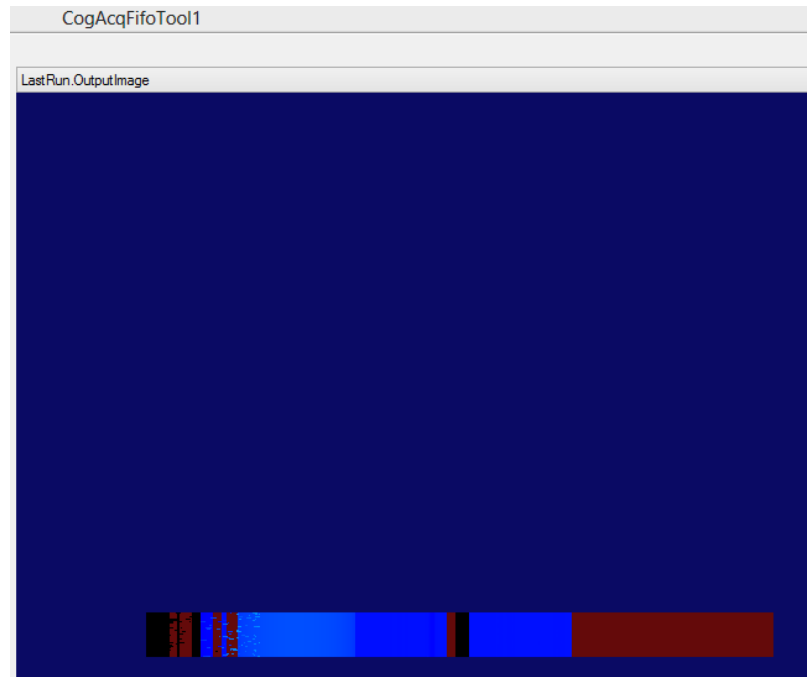


Figure III.4: Image Result from *CogAcqFifoTool1*.

The second function sub-block, *CogPixelMapTool1* converts the original image to an 8bit image as a required format for *CogToolBlock1* to use.

The *CogBlobTool* is the major function block in *CogToolBlock1*. Better processing the image using *CogBlobTool* requires enabling the Color Mapping function to enhance contrast. The color differences indicate the different heights from the laser sensor to the objects in AOI. The red-orange color is the highest area among the objects. A square blob is set to capture the highest area in the image. The center point is the one defined to be picked up. The results of *CogBlobTool1* with the enhanced contrast mode effect is shown in Figure III.5. The light green square is the highest area of the pipe. Figure III.6 is the result tag of *CogBlobTool1* getting the point of the pipe. In the Figure III.6, X, Y are the position coordinates of the center of the square.

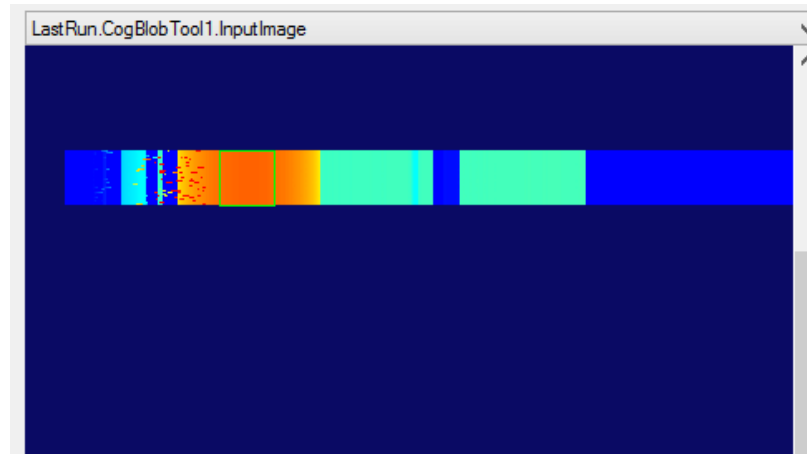






Figure III.5: Output Image from *CogBlobTool1*.

Cog3DRangeImageHeightCalculatorTool1 calculates the height from the sensor to the point we get from the last step. Figure III.7 is the screenshot of part of settings of *Cog3DRangeImageHeightCalculatorTool*.

Tools





Inputs/Outputs

Graphics



Inputs

Name	Type	Value
OutputImage	Cognex.VisionPro.ICogImage	Cognex.VisionPro.CogImage8Grey



Outputs

Name	Type	Value	Force Changed Event
Y	System.Double	-50.2648	<input checked="" type="checkbox"/>
X	System.Double	15.0024	<input checked="" type="checkbox"/>

Figure III.6: Getting the Pipe Point.

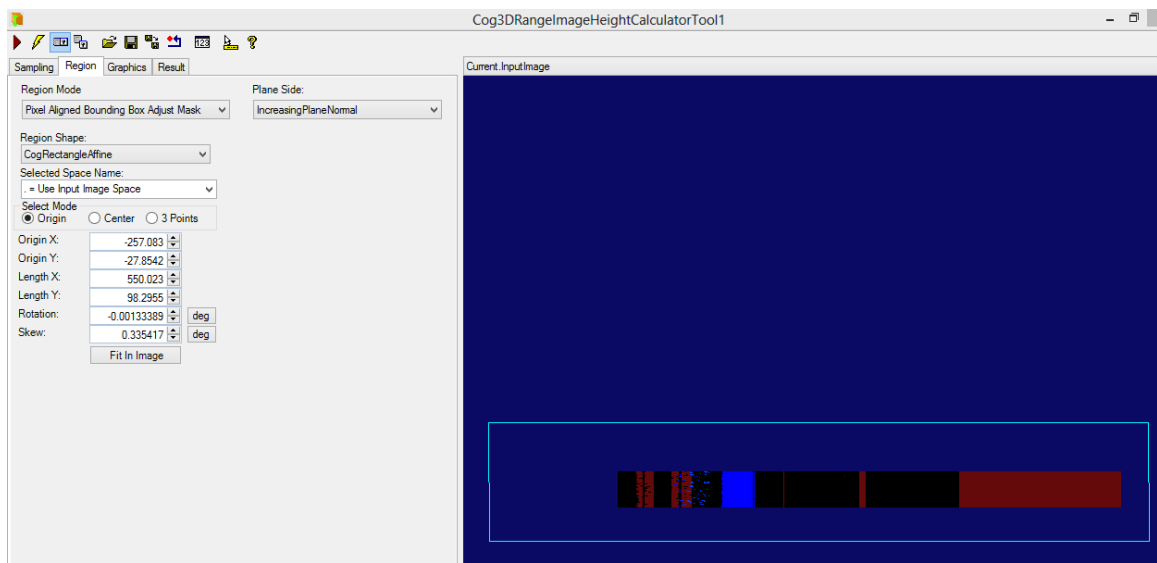


Figure III.7: Sub-block *Cog3DRangeImageHeightCalculatorTool* Part Settings.

Figure III.8 is an output interface of *Image* block. When the blocks have been run, the sensed result of the center point is shown in the output area.

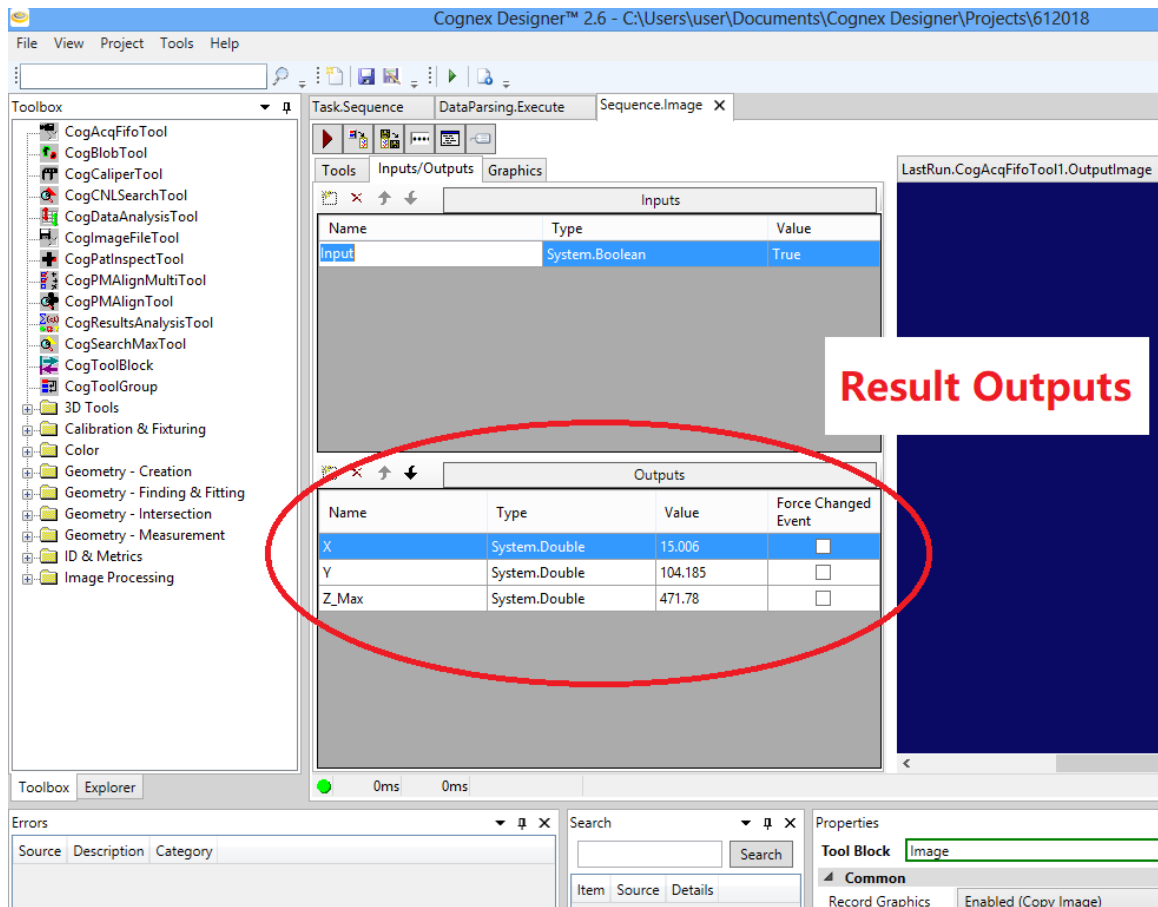


Figure III.8: Final Result Outputs.

The finished *Image* block with sub-blocks in Designer – Figure III.9.

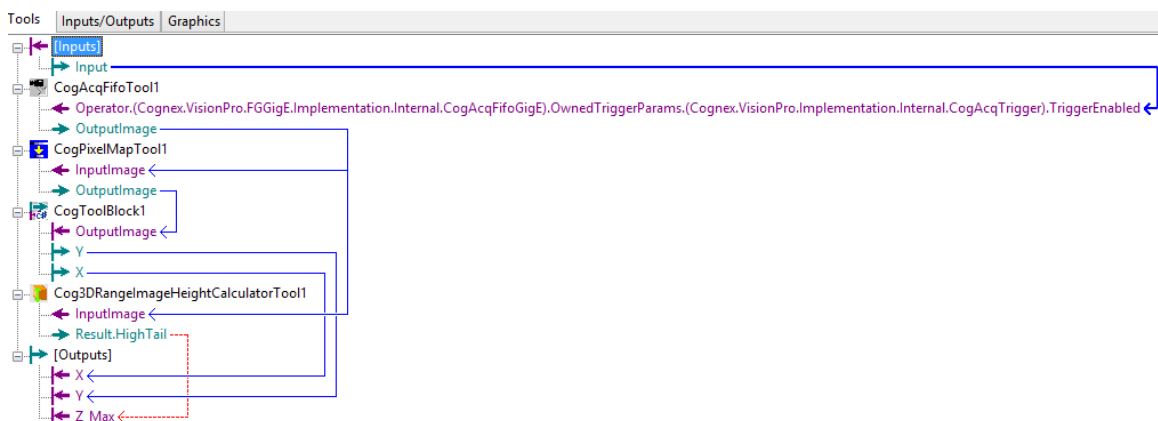


Figure III.9: *Image* Block with Sub-blocks.

Block 4. SendData

This is another interface block. It is for connection configuration. The pipe position " X, Y, Z " is converted from Double type to String and sent to the robot. The robot side receives data and guides the gripper to above the location of the pipe.

In the end, we add a user interface, which is shown in Figure III.10. The whole Task Sequence can be run by pressing the "run sequence" button in the menu of Designer or "start" button in the user interface. With this user interface, image processing of pipe handling can be viewed in the run.

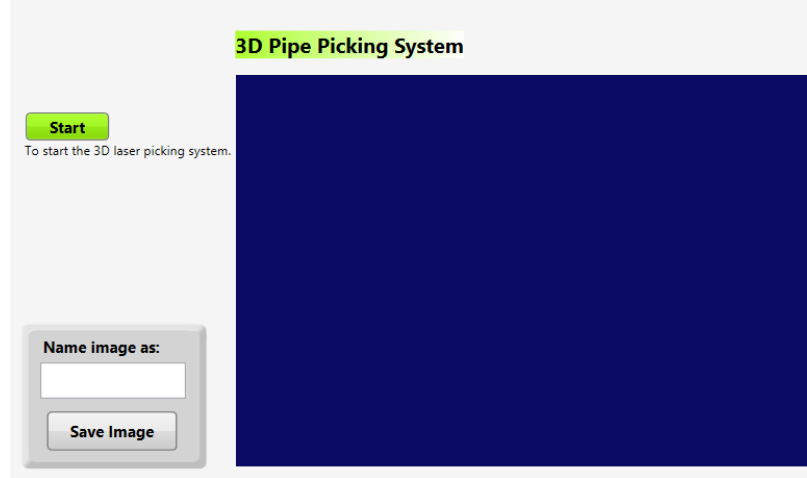


Figure III.10: User Interface.

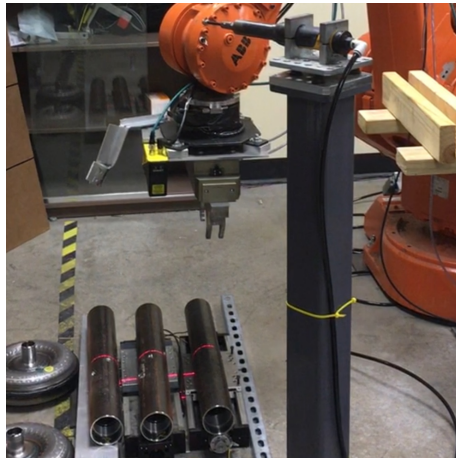
This pipe position from laser sensor $P_{sensed}(x_{sensed}, y_{sensed}, z_{sensed})$ is sent to the robot after a software sequence run. The $x_{sensed}, y_{sensed}, z_{sensed}$ are output variables of X, Y, Z from the above blocks. Since the robot frame and the laser sensor frame are not the same, the calculated pipe picking position under the robot frame is represented as $P_{pickup}(x_{robot} - (y_{sensed} - y_o), y_{robot} - (x_{sensed} - x_o), z_{robot} - z_o)$.

Figure III.11 is the photos of the procedure of a full pipe-picking process from recognition to loading a pipe to a fixed location. Since the proposed method is

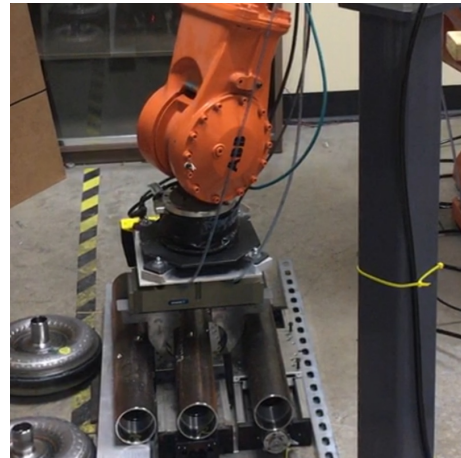
always searching for the highest point within the vision range, it should be fine with multiple layers of pipes. However, the gripper is best utilized for one-layer pipe grasping.

Summary

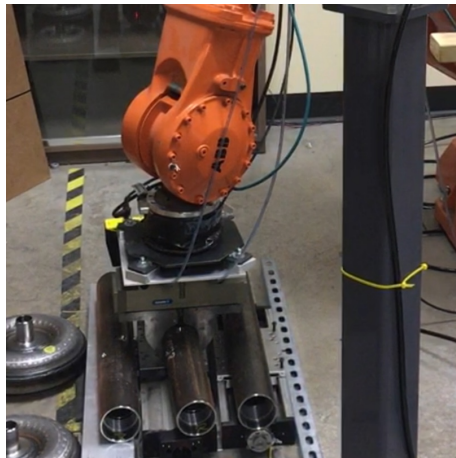
The 3D pipe picking system is verified and meets the requirements of pipe handling process. Once there is more than one pipe within the working area of laser displacement sensor camera, the tool block can output all the possible results in the sequence. The pipe picking process has been tested fifty times, 90% of them works well. The rest fail to pick pipes up due to operation miss. The result means that the proposed method can handle a pipe correctly.



(a) Recognizing a pipe.



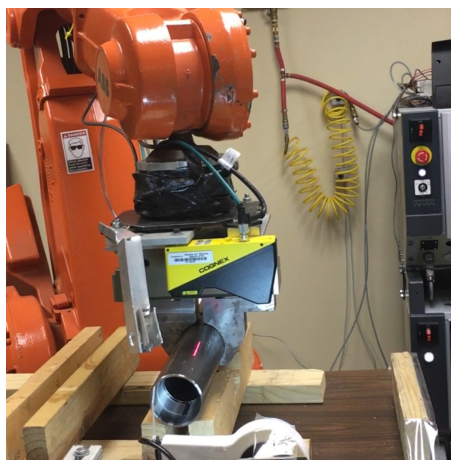
(b) Reaching the pipe.



(c) Picking up the pipe.



(d) Moving the pipe towards a fixed location.



(e) Placing the pipe to the fixed location.

Figure III.11: Final Pipe Picking.

Keyway Alignment Procedure

According to system configuration shown in Figure III.1, we have the 2D device, Cognex In-sight 7000 camera connected with the ABB robot for keyway alignment experiment.

The communication with each components of the system uses wired Ethernet. Two Ethernet ports of the robot controller are used. The service port is connected to host computer, in which RobotStudio is installed to access internal programs and files in the robot controller. The WAN port is connected with the Cognex In-sight 7000 camera. The robot software part of the communication mechanism uses socket programming in RAPID language. The overall 2D system is shown in Fig III.12.

After the communication is tested, the next step is to utilize this system to do keyway alignment experiments and get angle data for modeling.

The functions in Cognex In-sight Explorer Spreadsheet is encapsulated and provides some interfaces. We use some functions to process image from capturing original gray image to extracting and sending the angle to the robot. The process is as follows:

1. Image Acquisition: In step 1, we use "Image" function to get the original gray image;
2. Circle center point searching: "FindCircle" function is used to recognize a circle pattern on the gray image gotten from Step 1. Once the circle is found, the coordinate of the pixel point (*Row*, *Column*) of the circle center is shown in the spreadsheet cells;
3. Keyway pattern and edge line searching: The functions "FindPatMaxPatterns" and "TrainPatMaxPattern" are used to recognize the keyway pattern, Figure III.13 shows a gray image of pipe keyway (on the left

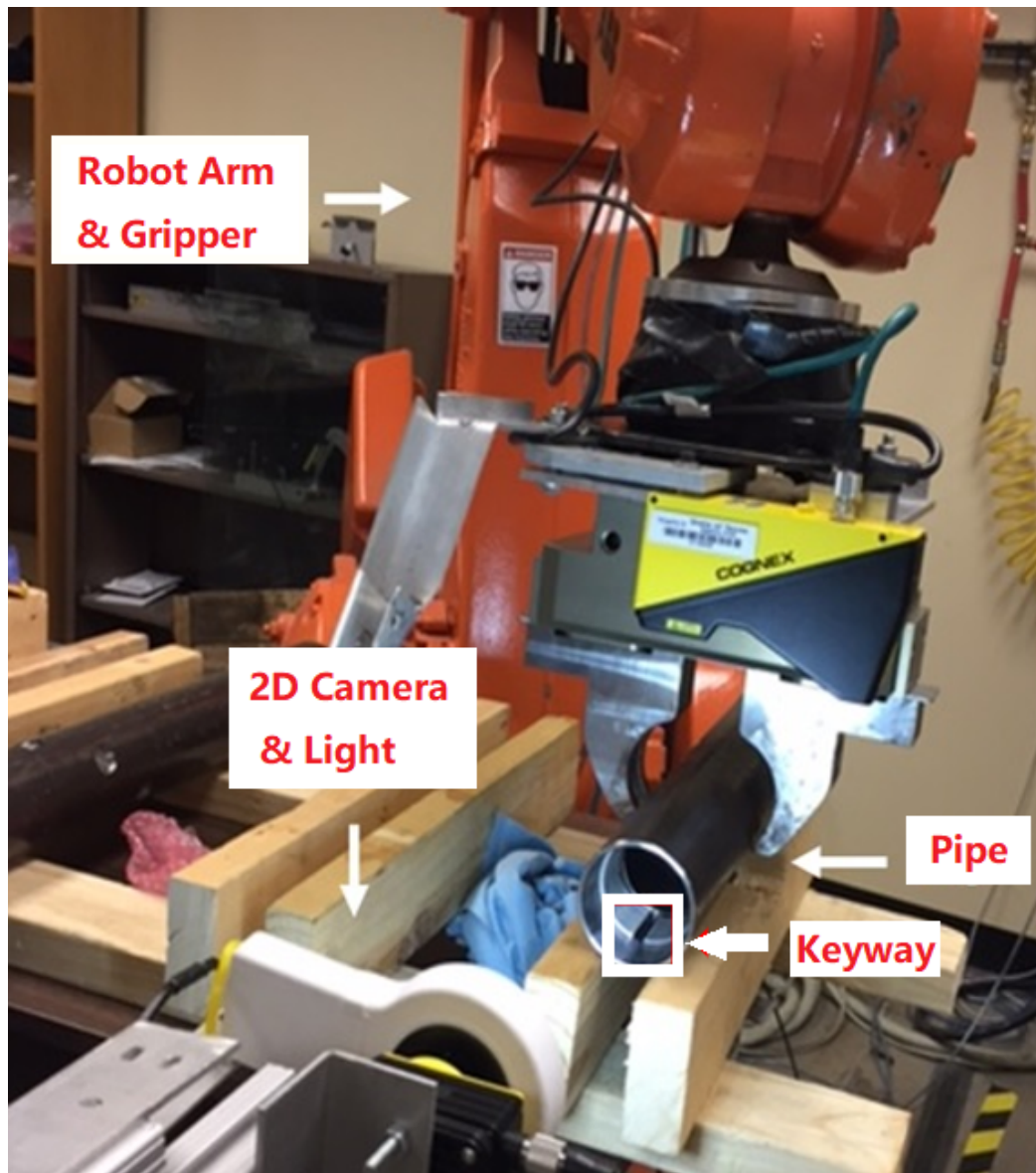


Figure III.12: The Integrated System Overview during the Experiment.

side), and the pattern of keyway found by "FindPatMaxPatterns" (on the right side). It captures the defined center of the keyway, recorded as $(Row, Column)$ pixel point coordinate in the spreadsheet cells;

4. Keyway angle calculation: In this step, we use the tool "LineToLine" to measure the keyway angle formed by two rays: one is from the circle center point to the keyway line, and the other one is the base line starts from the center point;
5. Data sending to the robot: Function "FormatString" formats the measured angle to a String. "TCPDevice" manages the devices and port information. The host is In-sight Camera, the client is the ABB robot, and the port is set to 23.

The screenshot of the finished process via the Cognex Explorer spreadsheet is shown in Figure III.14.

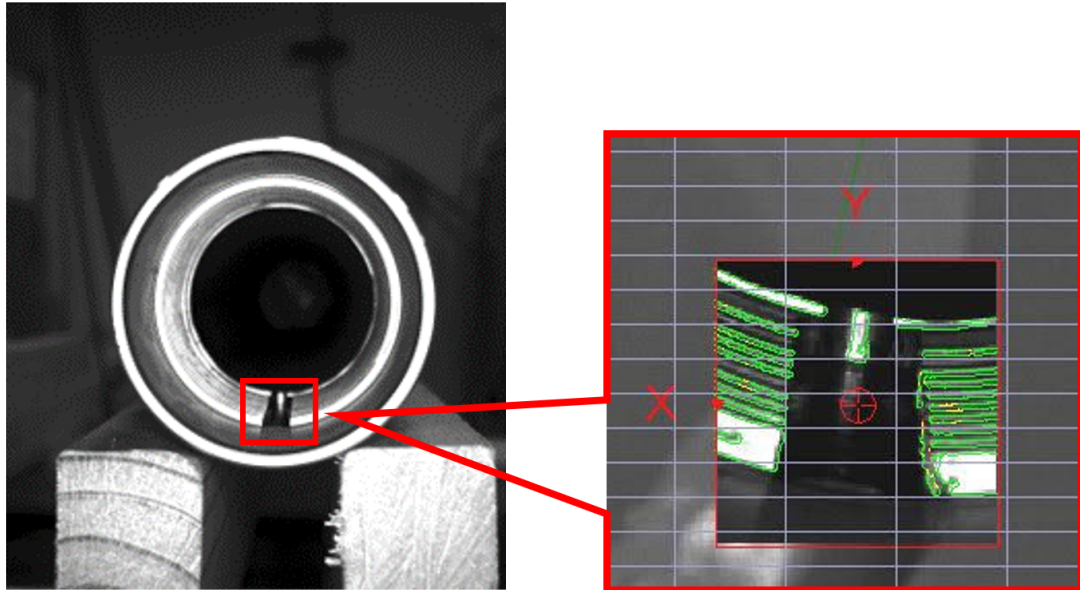


Figure III.13: Keyway Pattern Captured from "FindPatMaxPatterns" Function.

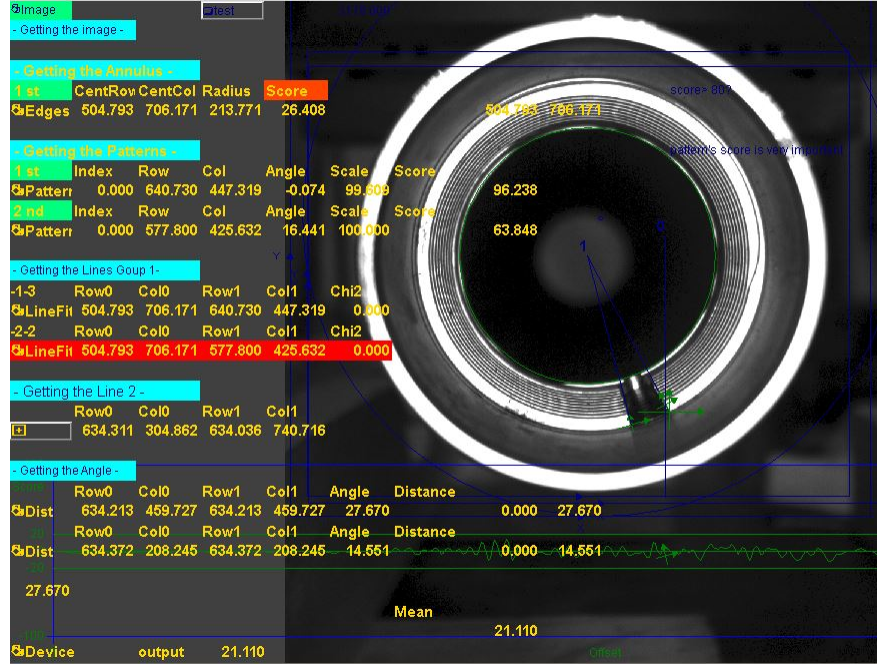


Figure III.14: Execution Using Cognex In-sight Explorer.

Data Collection

Ideally, the rotation angle would be the same as the detected angle, or the relationship would be linear with an offset[44]. However, after experimentation, it is found that the angle before rotation α and the angle after rotation β has a nonlinear relationship[45]. More data were collected and analyzed to address this nonlinear error problem.

Here is a little elaboration about the way of measuring two sets of angles. The angle is a representation of keyway orientation. The angle data are collected by repeating the same pick and rotating routine and with random keyway orientation. 130 experiments are performed. The original measured angle α and the angle after robot rotation β in about -20.0 to +20.0 degrees are recorded. 120 datasets are recorded in Table III.1. It is found that the robot rotation had error ranging from -3.0 to 3.0 degrees. The data sets show a nonlinear trend. However, in the research requirement, the final angle after rotating should be as small as possible.

Table III.1: Data collected (Unit: Degrees).

No.	α	β	No.	α	β	No.	α	β	No.	α	β
1	-25.58	-28.55	11	-18.9	-20.22	21	-13.81	-15.73	31	-12.79	-13.69
2	-25.4	-27.59	12	-17.86	-19.08	22	-13.67	-14.94	32	-11.93	-12.99
3	-23.19	-25.24	13	-17.61	-18.80	23	-13.51	-14.88	33	-10.11	-12.99
4	-23.15	-25.18	14	-17.36	-18.56	24	-13.46	-14.51	34	-9.85	-10.75
5	-21.16	-22.6	15	-17.33	-18.88	25	-13.37	-14.92	35	-8.81	-9.49
6	-20.87	-22.27	16	-17.32	-18.5	26	-13.31	-14.4	36	-8.53	-9.59
7	-20.63	-22.01	17	-15.74	-16.93	27	-13.28	-14.41	37	-8.31	-8.88
8	-19.45	-20.78	18	-14.09	-15.41	28	-13.25	-14.08	38	-8.06	-9.59
9	-19.41	-21.04	19	-13.87	-15.39	29	-12.85	-13.82	39	-8.04	-8.78
10	-19.3	-20.66	20	-13.86	-15.78	30	-12.81	-14.02	40	-7.99	-8.74
41	-7.82	-8.26	51	1.26	1.49	61	4.37	5.27	71	5.9	6.4
42	-7.09	-7.45	52	2.37	2.75	62	4.38	5.15	72	5.98	6.22
43	-6.14	-6.69	53	2.41	2.8	63	4.51	5.08	73	6.14	6.7
44	-3.85	-4.18	54	2.98	2.43	64	4.76	5.31	74	6.16	6.73
45	-3.84	-4.55	55	3.16	3.63	65	4.89	5.4	75	6.19	6.93
46	-3.56	-4.16	56	3.17	3.52	66	5.29	5.57	76	6.23	7.56
47	-3.26	-3.73	57	3.23	3.64	67	5.43	5.97	77	6.29	7.27
48	-2.14	-2.38	58	3.33	3.7	68	5.48	6.26	78	6.36	6.83
49	-1.25	-1.58	59	3.96	4.47	69	5.64	5.99	79	6.42	7.24
50	-0.26	-0.34	60	4.26	4.73	70	5.87	6.98	80	6.65	6.83
81	6.19	6.93	91	7.24	7.39	101	8.78	9.65	111	13.93	15.39
82	6.23	6.36	92	7.26	8.02	102	9.14	9.83	112	14.22	15.33
83	6.29	6.93	93	7.31	8.28	103	10.10	11.09	113	14.50	15.67
84	6.36	7.56	94	7.32	8.00	104	10.70	11.92	114	14.65	15.80
85	6.42	7.27	95	7.38	7.76	105	10.71	11.74	115	14.83	15.57
86	6.65	6.83	96	7.55	8.39	106	11.75	12.8	116	15.47	16.65
87	6.68	7.27	97	7.57	8.66	107	12.99	13.98	117	16.62	18.62
88	6.79	7.90	98	7.62	8.16	108	13.35	14.48	118	17.43	19.06
89	6.92	7.79	99	7.74	8.29	109	13.69	14.97	119	19.54	21.33
90	7.07	7.61	100	8.02	8.86	110	13.71	14.73	120	21.33	23.92

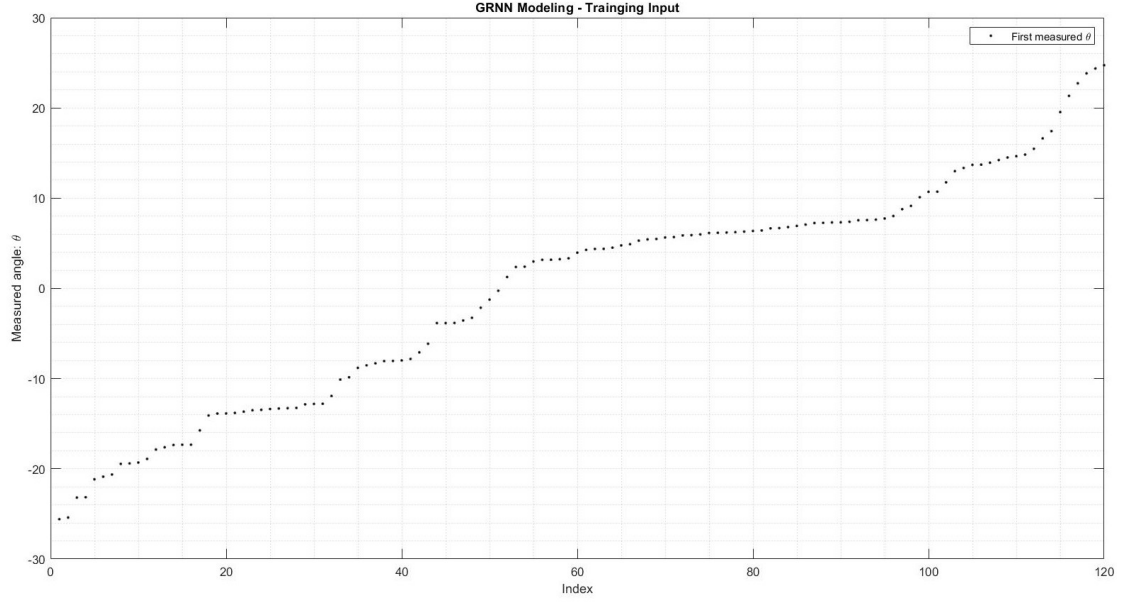


Figure III.15: Measured angle α_i .

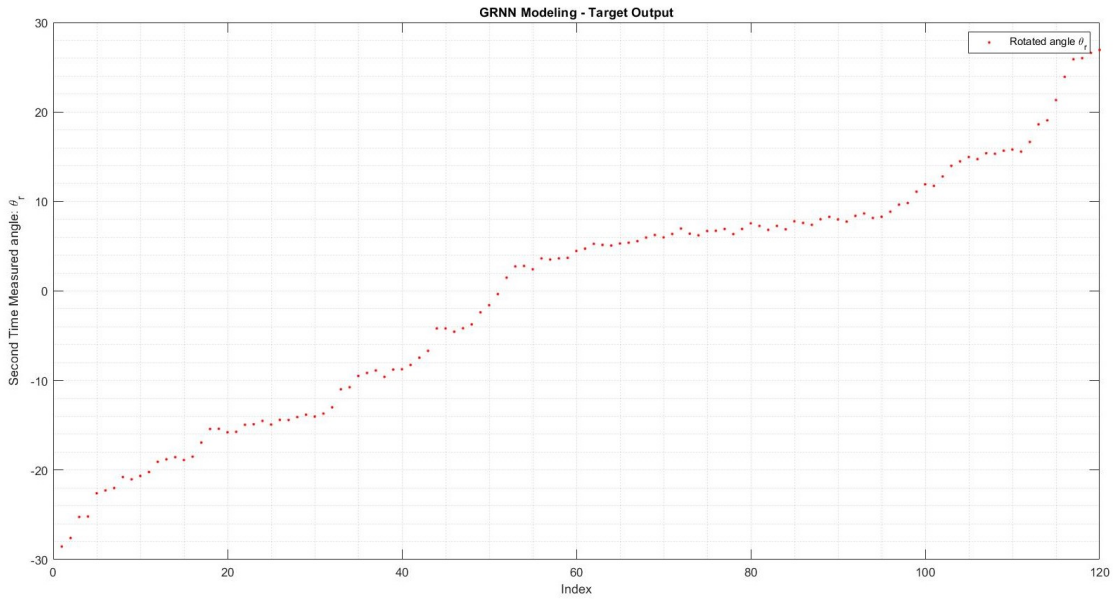


Figure III.16: Measured angle β_i .

Modeling

The data analysis and model building process were accomplished in MATLAB Ver.9.5.0 (R2018b). The training process used MATLAB *newgrnn* function: $net = newgrnn(\alpha, \beta, Spread)$, where α is assigned by the set of α , β is assigned by

the set of β , and *Spread* is assigned by σ . The total sample size is 120.

As mentioned in Chapter II, k -fold Cross-validation was used for pre-selecting the corresponding σ value of the training sample. The smoothing parameter σ was arranged by certain increments, which are the value of $\Delta\sigma$ for a predefined range $[\sigma_{min}, \sigma_{max}]$. The range is from 0.1 to 2.0, and the step is 0.1.

In the Cross-validation process, the mean value of the accuracy of the four results was used to estimate the modeling accuracy. Sample datasets were randomly divided into four different subsets. The samples were divided into 3 folds for training, 1 fold for validation. The program switched the training folds, and the test fold in turns for 4 times. The corresponding pseudo-code of Cross-validation in program is written as below:

Cross-validation method to find GRNN σ
1: Initialize σ , MSE
2: For σ from 0.1 to 2, step is 0.1, do
3: For k from 1 to 4, do
4: Remove subset k from dataset
5: Build net=newgrnn from remaining dataset
6: Use subset k for testing, get the output of GRNN y of the input
7: Get the MSE for testing data subset k
8: Record σ and MSE
9: End for
10: Select the case k with the lowest overall MSE value and the corresponding σ
11: End for
12: Select the case k with the lowest overall MSE

From running the program for forty times to collect different σ , MSE under the corresponding training sets, we get Table III.2 and Table III.3 of those forty results.

Table III.2: Different σ of the GRNN model results (Part1)

σ	MSE_{cv}	train error	test error	σ	MSE_{cv}	train error	test error
0.3	0.575	0.280	0.307	0.8	0.443	0.233	0.517
0.8	0.33	0.148	0.690	0.8	0.293	0.185	0.430
1.1	0.800	0.356	0.769	1.0	0.515	0.247	0.349
0.8	0.625	0.447	1.575	0.7	0.444	0.246	0.502
0.5	0.562	0.319	0.389	0.7	1.079	0.096	0.655
0.9	0.291	0.165	0.223	0.6	0.445	0.237	0.380
0.8	0.438	0.144	0.357	0.9	0.420	0.506	0.816
1.1	0.438	0.251	0.490	1.2	0.664	0.463	0.665
0.8	0.390	0.150	0.226	1.1	0.358	0.189	0.350
1.0	0.405	0.281	0.290	1.1	0.650	0.339	0.333

Table III.3: Different σ of the GRNN model results (Part2)

σ	MSE_{cv}	train error	test error	σ	MSE_{cv}	train error	test error
0.3	0.424	0.393	0.415	1.0	0.346	0.212	0.164
0.9	0.415	0.189	0.294	1.1	0.325	0.352	0.169
0.6	0.949	0.489	0.423	0.7	0.569	0.453	0.437
0.9	0.421	0.565	0.815	1.2	0.664	0.587	0.663
1.1	0.476	0.323	0.378	0.8	0.349	0.472	0.405
0.6	0.901	1.564	2.380	0.8	0.667	0.812	1.871
0.5	0.703	0.598	0.619	0.5	0.479	0.398	0.548
1.2	0.511	0.613	0.659	1.2	0.478	0.297	0.288
0.6	0.466	0.533	0.549	0.8	0.392	0.219	0.226
1.2	0.520	0.565	0.697	0.3	0.424	0.468	0.415

In one run, the $\sigma = 1.0$, the overall Cross-Validation MSE was 0.346, the test error was 0.164, and the training error was 0.212. This run had the smallest test error out of forty runs. In addition, the Cross-Validation MSE and training error were also among the smallest out of these tests. The three error values being close to each other, the training data shows a stable and even distribution. This was thanks to the application of Cross-Validation in this process.

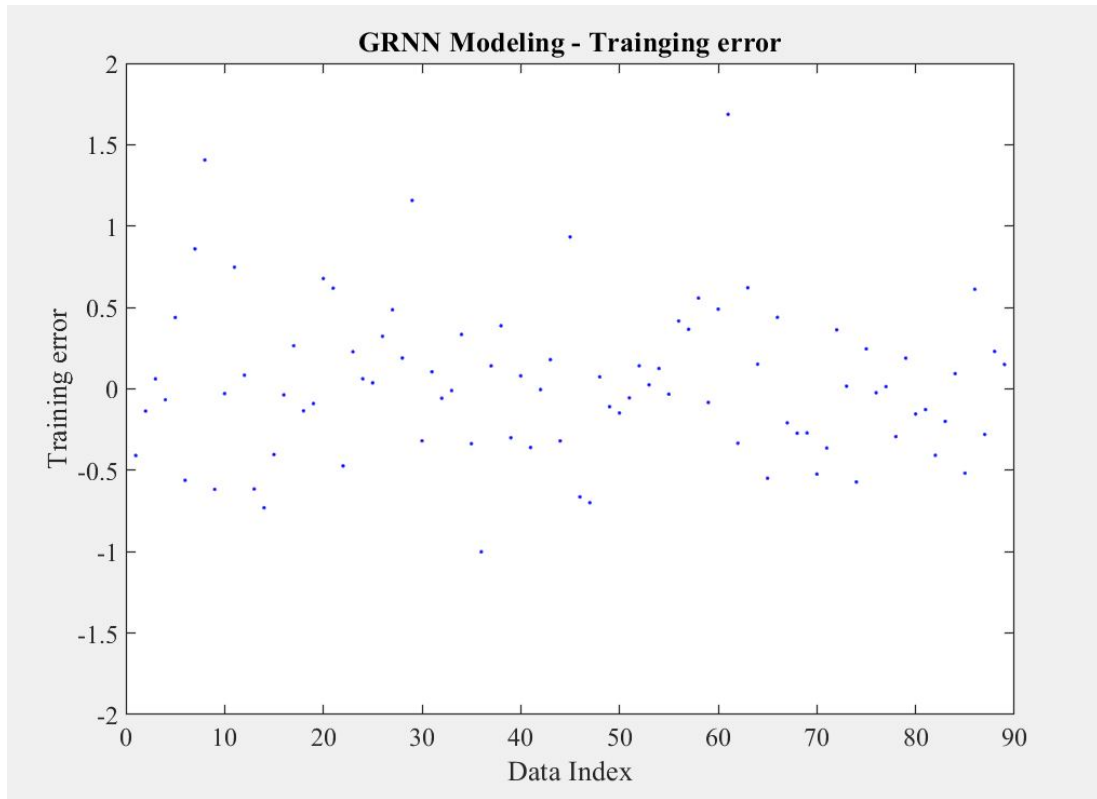


Figure III.17: Distribution of training errors.

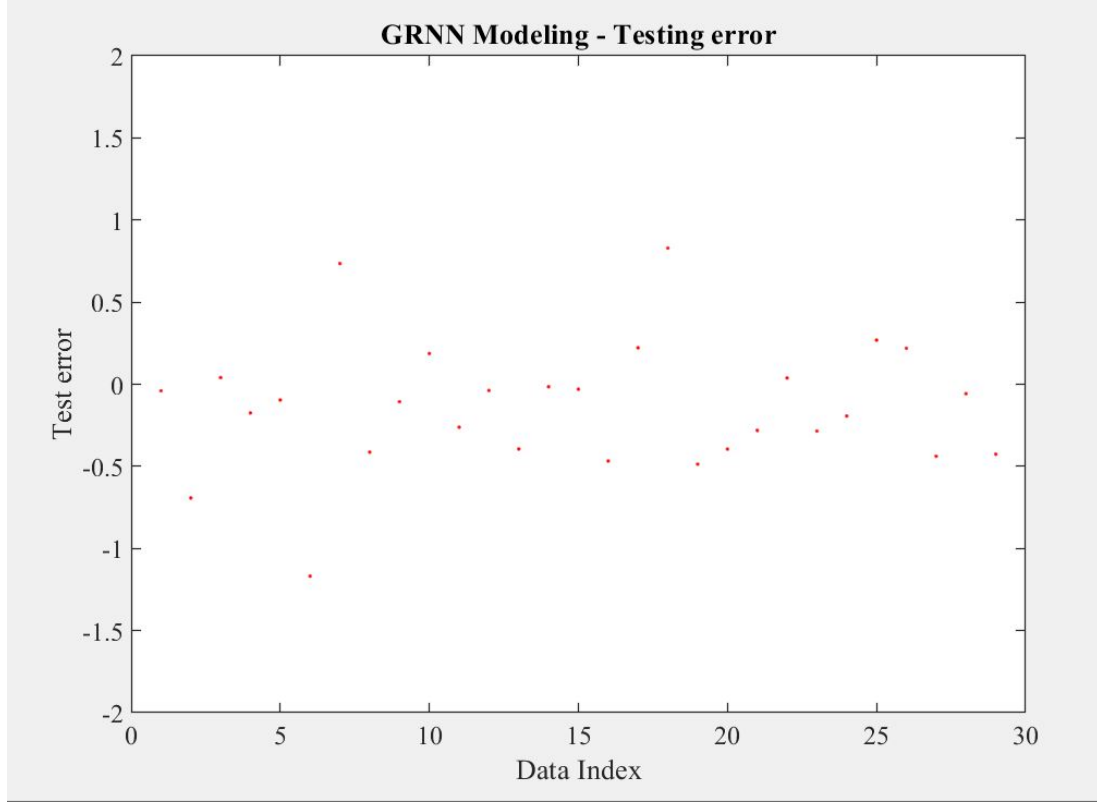


Figure III.18: Distribution of testing errors.

The test error shows there are twenty-six out of thirty points are within $[-0.5, 0.5]$, three points are within $[-1, -0.5)$ and $(0.5, 1]$, and one point is beyond -1.

Experiment Validation Using Predicted Angles

We tried to use the predicted value to control the robot operation. A 400x2 table was made. The detailed angle number is shown in Table Appendix A.1 and Table Appendix A.2. Column 1 contains the angles of all the possible cases from -20 to +20 degrees. Column 2 includes the corresponding predicted angles. The overall data plot is shown in Figure III.19.

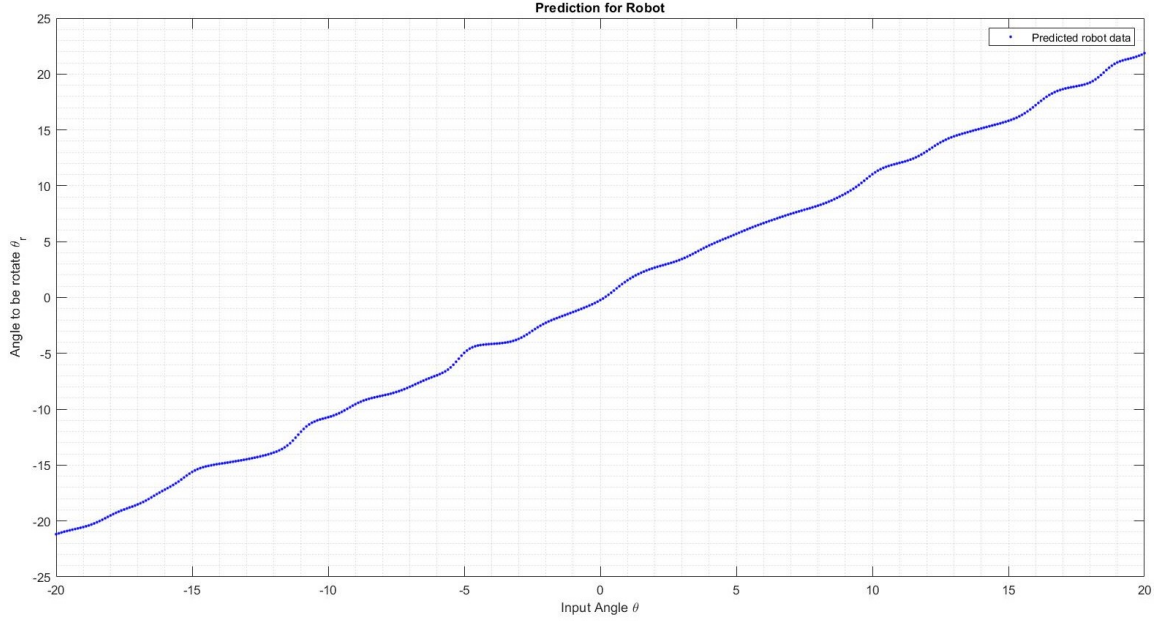


Figure III.19: Predicted Angle from -20 degrees to 20 degrees.

We inserted the prediction data to the robot control program through ABB RAPID. The leftover angles were measured to see whether they met the requirement or not. Once the keyway angle was measured, the robot used the predicted angle to rotate the pipe. These measured initial keyway angles represented as "Ini". The measured corresponding leftover keyway angles after rotation, represented as "Lef", We executed two sets of experiment, the data were collected and recorded in Table III.4 and Table III.5.

Table III.4: Result 1 (Unit: Degrees).

No.	Ini	Lef	No.	Ini	Lef	No.	Ini	Lef
1	-19.603	-0.257	11	-3.019	0.366	21	8.904	0.625
2	-13.808	0.455	12	-5.169	-0.178	22	6.812	0.461
3	-2.351	0.063	13	-7.725	-0.363	23	6.603	0.404
4	-6.029	-0.395	14	-9.881	-0.354	24	5.075	-0.175
5	-4.084	-0.038	15	-12.255	0.342	25	1.234	-0.494
6	-4.698	0.099	16	-12.221	0.287	26	3.101	-0.585
7	-6.935	0.397	17	-16.71	-0.187	27	2.561	-0.387
8	-5.287	-0.503	18	-19.885	-0.362	28	2.234	-0.125
9	-4.792	-0.402	19	19.735	0.49	29	-2.837	0.427
10	-4.505	-0.091	20	12.103	0.555	30	-3.655	0.436

Table III.5: Result 2 (Unit: Degrees).

No.	Ini	Lef	No.	Ini	Lef	No.	Ini	Lef
1	-19.970	-0.226	11	9.067	0.409	21	-0.265	-0.154
2	-17.178	-0.094	12	12.016	0.550	22	-1.993	-0.551
3	-13.338	0.252	13	14.575	0.612	23	-2.925	0.436
4	10.500	0.042	14	16.521	0.471	24	-3.617	-0.611
5	-8.836	0.110	15	18.886	0.606	25	-7.089	0.025
6	-6.639	-0.151	16	9.478	0.508	26	-8.468	0.057
7	-1.980	-0.159	17	7.758	0.420	27	-13.166	0.010
8	0.887	0.172	18	5.228	0.210	28	-15.068	-0.003
9	3.981	0.433	19	3.354	0.574	29	-16.570	-0.241
10	6.104	0.462	20	1.346	0.304	30	-17.851	-0.153

In the leftover angle column in the table above, we can see the absolute maximum value is 0.625, the absolute minimum value is 0.003, and the absolute average angle is 0.322.

Summary

In the process of establishing the model, a few essential parameters need to be taken care of: data size, the *Spread* value σ and the k number in k -fold Cross-validation. Once the data size is fixed, the bigger k is, the fewer data in each fold. If the k is big, but the whole data size is small, then it might result as a big bias, and not enough data for each shift case to use.

From the observation of training error and testing error, we can see the error with the corresponding σ and training, testing sets still have cases haven't met the requirement as being within the range $[-0.5, 0.5]$. We analyze the error that occurs in this validation experiment. The following are some possible factors:

1. robot assembly error;
2. TCP measurement error;
3. positioning error;
4. workpiece position calibration error.

Nevertheless, it is an alternative way from the current angle to address the problem. We are working on further research to lead to a better result.

IV. CONCLUSION AND FUTURE WORK

Conclusion

After analyzing the problem and performing a review of the relevant literature, we proposed using a generalized regression neural network with machine vision devices to address the problem of keyway alignment.

Our work includes setting up the keyway alignment robot system: integrating the robot with 2D and 3D vision devices, researching the General Regression Neural Network algorithm and applying it to solve the keyway alignment problem.

One of the contributions of this thesis is to use the angle information extracted from images and to establish the GRNN prediction model to achieve a shorter robot pipe handling cycle. The GRNN method requires a smaller training set than many methods, which means it requires less time to collect training data, thus it can reduce manufacturing costs. This method allows the pipe keyway's repositioning in one step rather than several steps, resulting in a faster process. Simultaneously, the GRNN method retains sufficient accuracy for the overall process to be successful. Even the result was not as expected, but it was close to our goal, and the error analyzing was given out. Another contribution is to establish a 3D image processing method for locating pipes, which makes it automatic and flexible to recognize and grasp metal pipes with robots.

There had been many difficulties that were handled. During setting up the system, the difficulties were, for example, image-related problems: such as light reflections, the object not being found in the image, or the work-piece not being in view. Mechanical problems were like interference between robot gripper and frame

or pipes, the robot reachability, collision with other objects in the laboratory. For both data collection and testing, experiments were repeated hundreds and thousands of times for testing and verifying if the operations were correct and repeatable, and data were usable.

GRNN has a quick training process, and only one primary parameter *Spread* needs to be determined. From experiment execution, GRNN is proven to be flexible, practical, and easy to use.

Future Work

There are lots of experiments that have been done when we address the thesis problem. But further investigation can be carried on for future research. The work will concern different methods of data collection and extraction. These are a few we can propose:

1. In the data collection, the data range selected is between -20° and $+20^\circ$. However, the keyway could be in any location. Therefore, future research could include a wider range, or even from -180° to $+180^\circ$.
2. Image processing plays a great role in angle measurement research. How accurate the angle can be detected affects the prediction result. For the keyway pattern search, we can see the keyway shape is slightly different each time, due to the light reflecting on the metal pipe. Also, when the keyway is not in the middle of the camera, the locations of shadows and over-reflections are different. If the shape changes dramatically, the camera will not recognize the keyway pattern very well. This increases the error rate of pattern finding. The image processing method can be improved. A new spreadsheet tool called "lineMax" has been developed by Cognex. It is used to precisely find lines and threads in a low contrast and other confusing environments. If we can use this

tool instead of "PatternMax", it might improve the accuracy, and efficiency compared to "PatternMax".

3. As for data collection, the smaller data size usually leads to a higher bias. However, the variance decreases with the data volume. We can improve to check the data with enough points for getting the low bias, but not too many points to cause big variance - to find the bias-variance balance. Also, checking the irreducible error maybe useful for analyzing the data characteristics before modeling.
4. In the end, we can search and compare a more advanced algorithm to the presented one. In the GRNN model, the parameters that need to be adjusted are very few. What we can do is to try other GRNN methods. Using the collected data, we can compare if different algorithms are better or worse.

APPENDIX SECTION

APPENDIX A

Table Appendix A.1: Predicted angle using for rotation (Unit: Degrees).(part1)

No.	α	β	No.	α	β	No.	α	β	No.	α	β	No.	α	β
1	-20.0	-21.386	11	-19.0	-20.340	21	-18.0	-19.418	31	-17.0	-18.318	41	-16.0	-17.337
2	-19.9	-21.280	12	-18.9	-20.237	22	-17.9	-19.115	32	-16.9	-18.219	42	-15.9	-17.235
3	-19.8	-21.175	13	-18.8	-20.134	23	-17.8	-19.015	33	-16.8	-18.120	43	-15.8	-17.137
4	-19.7	-21.070	14	-18.7	-20.031	24	-17.7	-19.015	34	-16.7	-18.022	44	-15.7	-17.038
5	-19.6	-20.965	15	-18.6	-19.928	25	-17.6	-18.915	35	-16.6	-17.923	45	-15.6	-16.940
6	-19.5	-20.860	16	-18.5	-19.826	26	-17.5	-18.715	36	-16.5	-17.824	46	-15.5	-16.841
7	-19.4	-20.756	17	-18.4	-19.724	27	-17.4	-18.615	37	-16.4	-17.727	47	-15.4	-16.742
8	-19.3	-20.652	18	-18.3	-19.622	28	-17.3	-18.516	38	-16.3	-17.628	48	-15.3	-16.643
9	-19.2	-20.548	19	-18.2	-19.520	29	-17.2	-18.516	39	-16.2	-17.530	49	-15.2	-16.544
10	-19.1	-20.444	20	-18.1	-19.418	30	-17.1	-18.417	40	-16.1	-17.431	50	-15.1	-16.445
51	-15.0	-16.346	61	-14.0	-15.342	71	-13.0	-14.316	81	-12.0	-13.255	91	-11.0	-12.139
52	-14.9	-16.246	62	-13.9	-15.240	72	-12.9	-14.212	82	-11.9	-13.146	92	-10.9	-12.024
53	-14.8	-16.147	63	-13.8	-15.139	73	-12.8	-14.107	83	-11.8	-13.036	93	-10.8	-11.909
54	-14.7	-15.046	64	-13.7	-15.037	74	-12.7	-14.002	84	-11.7	-12.926	94	-10.7	-11.794
55	-14.6	-15.946	65	-13.6	-14.832	75	-12.6	-13.897	85	-11.6	-12.815	95	-10.6	-11.678
56	-14.5	-15.846	66	-13.5	-14.832	76	-12.5	-13.791	86	-11.5	-12.704	96	-10.5	-11.562
57	-14.4	-15.746	67	-13.4	-14.730	77	-12.4	-13.685	87	-11.4	-12.592	97	-10.4	-11.446
58	-14.3	-15.645	68	-13.3	-14.623	78	-12.3	-13.578	88	-11.3	-12.480	98	-10.3	-11.330
59	-14.2	-15.545	69	-13.2	-14.523	79	-12.2	-13.471	89	-11.2	-12.367	99	-10.2	-11.213
60	-14.1	-15.544	70	-13.1	-14.420	80	-12.1	-13.363	90	-11.1	-12.253	100	-10.1	-11.097
101	-10.0	-10.981	111	-9.0	-9.844	121	-8.0	-8.759	131	-7.0	-7.703	141	-6.0	-6.665
102	-9.9	-10.866	112	-8.9	-9.734	122	-7.9	-8.653	132	-6.9	-7.603	142	-5.9	-6.561
103	-9.8	-10.750	113	-8.8	-9.623	123	-7.8	-8.547	133	-6.8	-7.499	143	-5.8	-6.456
104	-9.7	-10.635	114	-8.7	-9.513	124	-7.7	-8.441	134	-6.7	-7.395	144	-5.7	-6.352
105	-9.6	-10.521	115	-8.6	-9.405	125	-7.6	-8.336	135	-6.6	-7.290	145	-5.6	-6.271
106	-9.5	-10.407	116	-8.5	-9.296	126	-7.5	-8.231	136	-6.5	-7.186	146	-5.5	-6.142
107	-9.4	-10.293	117	-8.4	-9.188	127	-7.4	-8.126	137	-6.4	-7.082	147	-5.4	-6.037
108	-9.3	-10.190	118	-8.3	-9.080	128	-7.3	-8.021	138	-6.3	-6.978	148	-5.3	-5.932
109	-9.2	-10.068	119	-8.2	-8.973	129	-7.2	-7.916	139	-6.2	-6.873	149	-5.2	-5.827
110	-9.1	-9.956	120	-8.1	-8.866	130	-7.1	-7.812	140	-6.1	-6.769	150	-5.1	-5.722
151	-5.0	-5.616	161	-4.0	-4.544	171	-3.0	-3.434	181	-2.0	-2.275	191	-1.0	-1.081
152	-4.9	-5.510	162	-3.9	-4.435	172	-2.9	-3.320	182	-1.9	-2.157	192	-0.9	-0.961
153	-4.8	-5.404	163	-3.8	-4.326	173	-2.8	-3.206	183	-1.8	-2.038	193	-0.8	-0.841
154	-4.7	-5.297	164	-3.7	-4.216	174	-2.7	-3.091	184	-1.7	-1.919	194	-0.7	-0.721
155	-4.6	-5.191	165	-3.6	-4.105	175	-2.6	-2.976	185	-1.6	-1.800	195	-0.6	-0.602
156	-4.5	-5.084	166	-3.5	-3.995	176	-2.5	-2.861	186	-1.5	-1.680	196	-0.5	-0.483
157	-4.4	-4.977	167	-3.4	-3.883	177	-2.4	-2.744	187	-1.4	-1.561	197	-0.4	-0.364
158	-4.3	-4.869	168	-3.3	-3.771	178	-2.3	-2.628	188	-1.3	-1.441	198	-0.3	-0.245
159	-4.2	-4.761	169	-3.2	-3.660	179	-2.2	-2.511	189	-1.2	-1.321	199	-0.2	-0.127
150	-4.1	-4.653	170	-3.1	-3.557	180	-2.1	-2.393	190	-1.1	-1.201	200	-0.1	-0.010

Table Appendix A.2: Predicted angle using for rotation (Unit: Degrees). (part2)

No.	α	β	No.	α	β	No.	α	β	No.	α	β	No.	α	β
201	0.0	0.108	211	1.0	1.256	221	2.0	2.364	231	3.0	3.446	241	4.0	4.511
202	0.1	0.225	212	1.1	1.369	222	2.1	2.473	232	3.1	3.553	242	4.1	4.616
203	0.2	0.341	213	1.2	1.480	223	2.2	2.582	233	3.2	3.660	243	4.2	4.722
204	0.3	0.457	214	1.3	1.592	224	2.3	2.690	234	3.3	3.767	244	4.3	4.827
205	0.4	0.573	215	1.4	1.703	225	2.4	2.799	235	3.4	3.873	245	4.4	4.933
206	0.5	0.688	216	1.5	1.814	226	2.5	2.907	236	3.5	3.980	246	4.5	5.038
207	0.6	0.802	217	1.6	1.925	227	2.6	3.015	237	3.6	4.086	247	4.6	5.143
208	0.7	0.917	218	1.7	2.035	228	2.7	3.123	238	3.7	4.192	248	4.7	5.248
209	0.8	1.030	219	1.8	2.145	229	2.8	3.231	239	3.8	4.299	249	4.8	5.353
210	0.9	1.143	220	1.9	2.255	230	2.9	3.338	240	3.9	4.405	250	4.9	5.458
251	5.0	5.563	261	6.0	6.606	271	7.0	7.645	281	8.0	8.686	291	9.0	9.741
252	5.1	5.668	262	6.1	6.710	272	7.1	7.749	282	8.1	8.791	292	9.1	9.847
253	5.2	5.772	263	6.2	6.814	273	7.2	7.853	283	8.2	8.896	293	9.2	9.954
254	5.3	5.877	264	6.3	6.918	274	7.3	7.957	284	8.3	9.000	294	9.3	10.061
255	5.4	5.981	265	6.4	7.022	275	7.4	8.061	285	8.4	9.106	295	9.4	10.168
256	5.5	6.086	266	6.5	7.126	276	7.5	8.165	286	8.5	9.211	296	9.5	10.276
257	5.6	6.199	267	6.6	7.230	277	7.6	8.269	287	8.6	9.316	297	9.6	10.384
258	5.7	6.294	268	6.7	7.333	278	7.7	8.373	288	8.7	9.422	298	9.7	10.493
259	5.8	6.398	269	6.8	7.437	279	7.8	8.477	289	8.8	9.528	299	9.8	10.601
260	5.9	6.502	270	6.9	7.541	280	7.9	8.582	290	8.9	9.634	300	9.9	10.711
301	10.0	10.820	111	11.0	11.929	121	12.0	13.038	131	13.0	14.115	141	14.0	15.171
302	10.1	10.930	112	11.1	12.041	122	12.1	13.147	132	13.1	14.221	142	14.1	15.277
303	10.2	11.040	113	11.2	12.152	123	12.2	13.256	133	13.2	14.327	143	14.2	15.382
304	10.3	11.150	114	11.3	12.264	124	12.3	13.365	134	13.3	14.433	144	14.3	15.488
305	10.4	11.261	115	11.4	12.375	125	12.4	13.473	135	13.4	14.5380	145	14.4	15.594
306	10.5	11.372	116	11.5	12.486	126	12.5	13.581	136	13.5	14.644	146	14.5	15.700
307	10.6	11.483	117	11.6	12.597	127	12.6	13.688	137	13.6	14.749	147	14.6	15.807
308	10.7	11.594	118	11.7	12.708	128	12.7	13.795	138	13.7	14.855	148	14.7	15.913
309	10.8	11.706	119	11.8	12.818	129	12.8	13.902	139	13.8	14.960	149	14.8	16.020
310	10.9	11.817	120	11.9	12.928	130	12.9	14.009	140	13.9	15.066	150	14.9	16.127
351	15.0	16.234	361	16.0	17.324	371	17.0	18.453	381	18.0	19.633	391	19.1	20.996
352	15.1	16.342	362	16.1	17.435	372	17.1	18.568	382	18.1	19.754	392	19.2	21.122
353	15.2	16.450	363	16.2	17.550	373	17.2	18.684	383	18.2	19.876	393	19.3	21.249
354	15.3	16.558	364	16.3	17.658	374	17.3	18.801	384	18.3	19.999	394	19.4	21.375
355	15.4	16.667	365	16.4	17.770	375	17.4	18.918	385	18.4	20.122	395	19.5	21.501
356	15.5	16.775	366	16.5	17.883	376	17.5	19.036	386	18.5	20.245	396	19.6	21.627
357	15.6	16.884	367	16.6	17.996	377	17.6	19.154	387	18.6	20.369	397	19.7	21.753
358	15.7	16.993	368	16.7	18.109	378	17.7	19.273	388	18.7	20.494	398	19.8	21.879
359	15.8	17.103	369	16.8	18.223	379	17.8	19.392	389	18.8	20.619	399	19.9	22.004
360	15.9	17.213	370	16.9	18.338	380	17.9	19.512	390	19.0	20.870	400	20.0	22.129

REFERENCES

- [1] PWC, “Robot-ready: Adopting a new generation of industrial robots.,” Jun 2018.
- [2] S. B. Gaswirth, *Assessment of continuous oil resources in the Wolfcamp shale of the Midland Basin, Permian Basin Province, Texas, 2016*. Open-file report: 2017-1013, U.S. Department of the Interior, U.S. Geological Survey, 2017.
- [3] *Assessment of undiscovered oil and gas resources in the Eagle Ford Group and associated Cenomanian-Turonian strata, U.S. Gulf Coast, Texas, 2018*. U.S. Geological Survey fact sheet: 2018-3033, U.S. Department of the Interior, U.S. Geological Survey, 2018.
- [4] B. Vance, “Api [american petroleum institute].,” *A Dictionary of Abbreviations*, 2011.
- [5] H. Chen and Y. Liu, “Robotic assembly automation using robust compliant control.,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 293 – 300, 2013.
- [6] K. Feldmann and S. Slama, “Highly flexible assembly-scope and justification.,” *CIRP Annals - Manufacturing Technology*, vol. 50, no. 2, pp. 489–498, 2001.
- [7] B. G. Batchelor, *Machine Vision for Industrial Applications*, pp. 1–59. London: Springer London, 2012.
- [8] J. Li, *Machinery Electronics and Control Engineering III*. No. Volume 441 in Applied Mechanics and Materials, Trans Tech Publications, 2014.
- [9] E. R. Davies, *Computer and Machine Vision : Theory, Algorithms, Practicalities.*, vol. 4th ed. Academic Press, 2012.
- [10] G. Godfrey and A. C. Pinder, *Food process monitoring systems / edited by A.C. Pinder and G. Godfrey*. London ; New York : Blackie Academic Professional, 1993., 1993.
- [11] S. Sahoo and B. Choudhury, “A robotic assistance machine vision technique for an effective inspection and analysis,” *International Journal of Electrical and Computer Engineering*, vol. 5, no. 1, pp. 46–54, 2015.
- [12] A. Toker, *Machine vision : technology and applications*. Clanrye International, 2015.
- [13] H. Colestock, *Industrial robotics : selection, design, and maintenance*. McGraw-Hill, 2005.
- [14] T. M. Anandan, “Intelligent robots: A feast for the senses.,” Jun 2015.

- [15] T. M. Mitchell, *Machine Learning*. McGraw-Hill series in computer science, New York : McGraw-Hill, c1997., 1997.
- [16] P. Donmez, “Introduction to machine learning.,” *Natural Language Engineering*, vol. 19, no. 2, pp. 285 – 288, 2013.
- [17] J. Liang and R. Du, “Thermal comfort control based on neural network for hvac application.,” *Proceedings of 2005 IEEE Conference on Control Applications, 2005.*, p. 819, 2005.
- [18] T. Chow, G. Zhang, Z. Lin, and C. Song, “Global optimization of absorption chiller system by genetic algorithm and neural network,” *Energy and Buildings*, vol. 34, no. 1, pp. 103 – 109, 2002.
- [19] A. Kramer and F. Morgado-Dias, “Applications of artificial neural networks in process control applications: A review,” in *2018 International Conference on Biomedical Engineering and Applications (ICBEA)*, pp. 1–6, July 2018.
- [20] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, “Unsupervised learning based on artificial neural network: A review,” in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 322–327, Oct 2018.
- [21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors.,” *Nature*, vol. 323, pp. 533 – 536, 1986.
- [22] S. Fan, Y. Zhao, H. Xie, and C. Li, “Risk evaluation of rural financial organization operation based on the bp.,” *Journal of Digital Information Management*, vol. 10, no. 5, pp. 341–346, 2012.
- [23] M. J. D. Powell, “Radial basis functions for multivariable interpolation: a review,” in *Algorithms for approximation (Shrivenham, 1985)*, vol. 10 of *Inst. Math. Appl. Conf. Ser. New Ser.*, pp. 143–167, Oxford Univ. Press, New York, 1987.
- [24] J. Moody and C. J. Darken, “Fast Learning in Networks of Locally-Tuned Processing Units,” *NEURAL COMPUTATION*, vol. 1, pp. 281–294, SUM 1989.
- [25] D. F. Specht, “A general regression neural network,” *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.
- [26] D. F. Specht, “Probabilistic neural networks,” *Neural Networks*, vol. 3, no. 1, pp. 109 – 118, 1990.
- [27] X. Wang, M. You, Z. Mao, and P. Yuan, “Tree-structure ensemble general regression neural networks applied to predict the molten steel temperature in ladle furnace,” *Advanced Engineering Informatics*, vol. 30, pp. 368–375, aug 2016.

- [28] A. J. Al-Mahasneh, S. G. Anavatti, M. A. Garratt, and M. Pratama, “Evolving general regression neural networks using limited incremental evolution for data-driven modeling of non-linear dynamic systems,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 335–341, Nov 2018.
- [29] A. Kaur, M. K. Dutta, and J. Prinosil, “General regression neural network based audio watermarking algorithm using torus automorphism,” in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–4, July 2018.
- [30] M. M. Islam, G. Lee, and S. N. Hettiwatte, “Application of a general regression neural network for health index calculation of power transformers.,” *International Journal of Electrical Power and Energy Systems*, vol. 93, pp. 308 – 315, 2017.
- [31] R. Rooki, “Application of general regression neural network (GRNN) for indirect measuring pressure loss of herschel–bulkley drilling fluids in oil drilling,” *Measurement*, vol. 85, pp. 184–191, may 2016.
- [32] Cognex, *DS1000 Series Displacement Sensors Reference*. Cognex, 2017.
- [33] Cognex, *In-Sight 7000 Series Communications and Programming Guide*. Cognex, 2017.
- [34] Cognex, *In-Sight 7000 Series Vision System Installation Manual*. Cognex, 2011.
- [35] Z. Lai, R. Xiong, H. Wu, and Y. Guan, “Integration of visual information and robot offline programming system for improving automatic deburring process,” in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1132–1137, Dec 2018.
- [36] J. Xu, X. Yang, S. Zhu, K. Chen, Y. Wang, and P. Liang, “Error analysis of a 3d ultrasound-based robotic system for hepatic tumors,” in *2007 International Conference on Mechatronics and Automation*, pp. 2651–2656, Aug 2007.
- [37] E. Parzen, “On estimation of a probability density function and mode.,” *The Annals of Mathematical Statistics*, vol. 33, no. 3, p. 1065, 1962.
- [38] T. Cacoullos, “Estimation of a multivariate density,” *Annals of the Institute of Statistical Mathematics*, vol. 18, pp. 179–189, Dec 1966.
- [39] MathWorks, *MATLAB 7.0 (R14SP2)*. 2005.
- [40] Mathworks, “newgrnn, design generalized regression neural network.”
- [41] Mathworks, “Generalized regression neural networks.”
- [42] S. Geisser, *Predictive inference: An introduction*. School of Statistics, University of Minnesota: CRC Press, 2017.

- [43] H. Cigizoglu, “Application of generalized regression neural networks to intermittent flow forecasting and estimation,” *JOURNAL OF HYDROLOGIC ENGINEERING*, vol. 10, no. 4, pp. 336 – 341, n.d.
- [44] Y. Liu, N. Xi, G. Zhang, X. Li, H. Chen, C. Zhang, M. J. Jeffery, and T. A. Fuhlbrigge, “An automated method to calibrate industrial robot joint offset using virtual line-based single-point constraint approach,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 715–720, Oct 2009.
- [45] G. W. Snedecor, *Curvilinear regression.*, pp. 308 – 335. Collegiate Press, 1938.