上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

# 学士学位论文

## BACHELOR'S THESIS

论文题目： BuilderX Prototype of Excavator Automatic Obstacle Avoidance


学生姓名：刘世纪, 刘金杰, 涂勋, 张一苇, 黄鹏远

学生学号：517370910170, 517370910142, 517370910149, 517370910161, 517370910025

专业：电子与计算机工程

指导教师：马澄斌

学院(系)：密西根学院

# 上海交通大学

# 学位论文原创性声明

本人郑重声明：所呈交的学位论文《BuilderX Prototype of ExcavatorAutomatic Obstacle Avoidance》，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：刘世纪 [illegible] 张一苇

涂勋 黄鹏远

日期： 2021 年 8 月 3 日

# 上海交通大学
# 毕业设计（论文）版权使用授权书

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

**保密**☑，在 2 年解密后适用本授权书。

本论文属于

**不保密**□。

（请在以上方框内打“√”）

作者签名：刘世纪 [illegible] 张一帆
涂勋 黄鹏远

指导教师签名：[illegible]

日期：2021 年 8 月 3 日

日期：2021年 8 月 3 日

# BuilderX Prototype of Excavator Automatic Obstacle Avoidance

## 摘要

挖掘机在很长时间内都在建筑及房地产行业中占据重要地位。挖机系统的发明极大的提升了建筑工人的工作效率。然而，现代工业环境下挖机操作手可能会遭遇复杂或不稳定地形，这种环境下的挖机操作有着较高的事故及伤亡几率，并且需要熟练的操作手才能达到较高的生产力。故而，用自动化机械来解决这个问题不失为一个明智的解决方案。这个项目由拓疆者公司赞助，旨在开发一套挖机臂自动避障的挖掘机系统。它能够优化一条挖机臂不与障碍物发生碰撞的运动路径。这个原型机通过一个旋转的红外摄像机感知周围的环境并生成一套标注出所有障碍物的点云信息。而后，A* 路径规划算法确定一条通向用户在可视化界面上指定的目标挖斗位置的无碰撞路径。相应的指令传送至挖机模型使其遵循规划出来的路径完成指定任务。

**关键词：** 路径规划，物体检测，避障，自动化系统

# BUILDERX PROTOTYPE OF EXCAVATOR AUTOMATIC OBSTACLE AVOIDANCE

## ABSTRACT

Excavators have long played a critical role in the field of construction and real estates. The invention of excavation systems substantially boosts the working efficiency of construction workers. Nevertheless, operating excavators risks high possibility of accidents and injuries under modern industrial circumstances where operators can encounter complex or unstable terrain, and it demands experienced operators to maintain productivity. As a result, it is sensible to resort to autonomous robotics to deal with this issue. Sponsored by BuilderX, this project aims at developing an excavator prototype with autonomous arm obstacle avoidance. It is able to optimize a trajectory for excavator arm to move without collision with obstacles. This prototype perceives its surroundings by rotating an infrared camera to generate a set of point cloud representation of all obstacles. A* path planning algorithm is then applied to determine a collision-free path to the user defined target bucket position over a simple user interface. The corresponding commands are generated and sent to the excavator model to follow the planned trajectory and finish the task.

**Key words:** Path Planning, Object Detection, Obstacle Avoidance, Autonomous System

# Contents

# 1 Introduction

In the real world, an autonomous excavator has many benefits, such as improving efficiency and reducing injuries. In this project, our main goal is to develop an excavator system that can move its bucket to a user-specified coordinate and avoid the obstacle in the path automatically. In order to build such a system, there are at least two problems that remain to be solved. One is that the system should be able to detect the obstacle in the surroundings and determine their locations. The other is that the system should be able to move its excavator arm in a sensible manner and not to collide with the obstacle.

To solve these problems, our group is divided into two parts. One is the Perception part, which is going to focus on the first problem. Pengyuan Huang and Yiwei Zhang are responsible for this part. Their job is to use the camera and acquire the visual data in the surroundings. The other one is the Motion Planning part, which is going to focus on the second problem. Xun Tu, Shiji Liu and Jinjie Liu are responsible for this part. Their job is to realize precise control of the excavator model and to develop the obstacle avoidance algorithm. To demonstrate our works, all the features are based on a prototype of the excavator provided by our sponsor (See Figure 1–1). Generally, we firstly build our works on the model separately for around a month. And then all the works are merged and fine-tuned to generate a final product.

**Illustration 1–1 Excavator model**

# 2 Design Specifications

Our engineering specifications are summarized as in Table 2–1.

**Table 2–1 Summary of engineering specifications**

| Engineering Specs. | Target Values | Units |
|---|---|---|
| Joint angle precision (boom & stick) | 5 | deg. |
| Joint angle precision (base) | 4 | deg. |
| Bucket position accuracy | 8 | cm |
| Perception accuracy and precision | 80 | % |
| Perception range | 1.3 | m |
| Success rate of obstacle avoidance | 85 | % |
| Cost | 4000 | ¥ |

We use QFD to generate our engineering specifications, which is also included as Figure 2–1. Our customer requirements could be summarized into following bullet points.

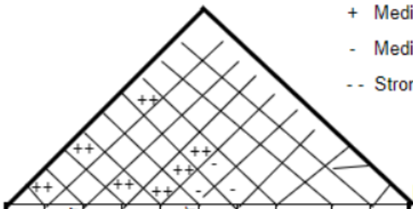
- Basic functionalities / motions of excavator base / boom / stick
- Autonomous excavator arm obstacle avoidance within workspace
- Reliance on camera vision to the greatest extent
- Validation tests under different conditions
- Cost control

We can see that the engineering specifications are closely related to our customer requirements. For arm obstacle avoidance, we choose joint angle precision to measure the arm motion control. The angles between bucket, stick and boom respectively should be the exact values calculated by our motion planning algorithm. The deviation for stick and boom should be within 5 degree, and the deviation for base should be within 4 degree. For perception part, we regard perception accuracy as the success rate of eliminating out the excavator arm. Under that case, the target value is required to be above 80%.

To ensure excavator arm obstacle avoidance within working area and maximum utilization of visual information, the visual perception range should be large enough to cover the whole working zone. Given the size of the model, the radius of the perception range should be no less than 1.3 m. To ensure that our system has good adaptability, the following two specifications should be satisfied under all testing environments. The first one is the success rate of arm obstacle avoidance. The value should be above 85%. Another is the bucket position error. The distance between the given coordinate and final bucket position in real life should be within 5 cm. Besides, under the requirement of our sponsor, the total cost should be under 4000 RMB.

Quality Function Development (QFD)

Correlation between eng. Specs
++ Strong Positive
+ Medium Positive
- Medium Negative
- - Strong Negative




Benchmarks

| | Weight (1-10) | Joint angle precision | Bucket position accuracy | Control response time | Success rate of obstacle | Perception accuracy & pr | Perception range | Vision algorithm runtime | Cost | | | *AES(Baidu)* | *ALS* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Basic functionalities/motions of excavator base/boom/stick | 10 | 9 | 9 | 3 | | | | | | | | 5 | 5 | |
| Autonomous excavator arm obstacle avoidance within workspace | 10 | 3 | | 9 | 9 | 9 | 1 | 3 | | | | 3 | 3 | |
| Reliance on camera vision to the greatest extent | 5 | | | | 3 | 9 | 3 | 9 | 9 | | | 3 | 1 | |
| Validation test under different conditions/environment | 7 | | | | 9 | 3 | 3 | | | | | 5 | 5 | |
| Obstacle avoidance w/ arm in motion | 3 | | | 9 | 9 | 9 | 1 | 3 | | | | 2 | 2 | |
| Cost control | 4 | | | | | | | | 9 | | | 1 | 1 | |
| **Measurement Unit** | | ° | cm | s | % | % | m | s | ¥ | | | | | |
| **Target Value** | | 2 | 5 | 0.3 | 85 | 80 | 1.3 | 0.1 | 4k | | | | | |
| **Total** | | 120 | 90 | 147 | 195 | 183 | 49 | 84 | 81 | 949 | | | | |
| **Normalized** | | 0.13 | 0.09 | 0.15 | 0.21 | 0.19 | 0.05 | 0.09 | 0.09 | 1.00 | | | | |
| *AES(Baidu)* | | | 10 | | | 60 | | 0.2 | >30k | | | | | |
| *ALS* | | 1 | 10 | 0.1 | | | | 0.1 | >30k | | | | | |
| *Rank Order of ECs* | | 4 | 5 | 3 | 1 | 2 | 8 | 6 | 7 | | | | | |

**Competitive benchmarks**

5 = satisfies perfectly
4 = satisfies "mostly"
3 = satisfies "somewhat"
2 = satisfies "slightly"
1 = doesn't satisfy at all

Excavator System
Loading System

Correlation between customer reqs and eng. Specs
9 = Strong Relationship
3 = Medium Relationship
1 = Small Relationship
(blank) = Not Related

**Illustration 2–1 QFD table**

# 3 Concept Analysis

## 3.1 Concept Overview

### 3.1.1 Concept generation for motion planning and control

The mission for the motion planning and control part is to

- Plan a collision-free path for the excavator arm to put its bucket to the target
- Control the excavator base, boom and stick to control follow the planned path

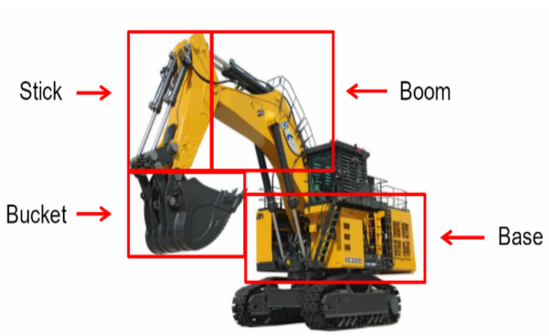



**Illustration 3–1 Base, boom and stick of the excavator**

To fulfill the mission, we need

- An algorithm for generating collision-free path
- A method to measure joint angles of the base, boom and stick joints
- A method to control the excavator base
- A method to control the excavator arm

For each of the 4 needs, we generated 3 concepts, which is shown in the figure below:

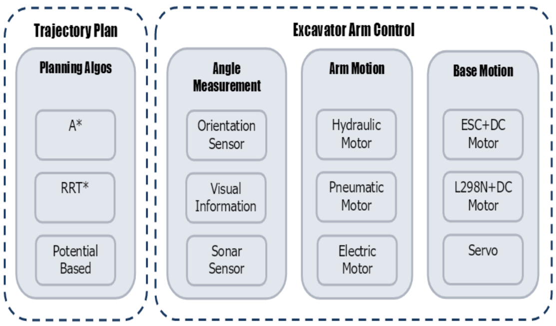



**Illustration 3–2 Generated concepts for motion part**

### 3.1.2 Perception

To fulfill the mission of the motion control part, our perception part is required to

- Perceive the surroundings of our excavator system
- Generate a point cloud representation of the surroundings
- Detect the obstacles to avoid

To fulfill this requirement, we need

- A method to perceive the the surrounding information
- An algorithm to generate a point cloud representation from the perceived information
- An algorithm to detect obstacles within the point cloud representation

For those three needs, we generated 3 concepts, which is shown in the Figure 3–3:

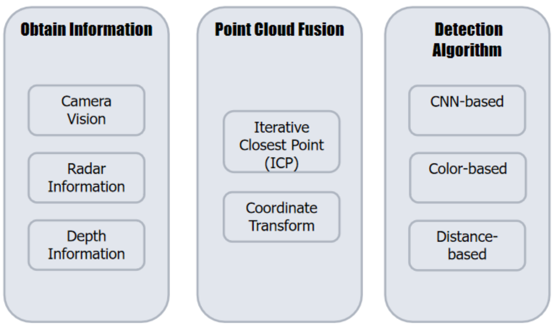



**Illustration 3–3 Generated concepts for perception part**

We will discuss the generated concepts in detail in the selection process section, and select the concepts we believe to be the most suitable for our project.

## 3.2 Concept Selection Process

### 3.2.1 Motion planning and control part

As described in the concept generation section, the motion planning and control part needs to fulfill 4 needs to complete its mission:

- An algorithm for generating collision-free path

- A method to measure joint angles of the base, boom and stick joints

- A method to control the excavator base
- A method to control the excavator arm

We generated 3 concepts for each of the needs, as shown in Fig. 3–2.

In this section, we will discuss the concepts in detail, and select the most suitable concepts for our project.

### (1) Concept selection for planning algorithm

We found three path planning algorithms that can be applied to our task:

- A* algorithm
- RRT* algorithm
- Potential field method

For our project, only three degrees of freedom are involved, which can be handled by the A* algorithm. Also, the control precision of each degree of freedom in our system is around 4 - 5 degrees, which means a discretion of the joint space is a must for our system. Therefore, the drawbacks for the A* algorithm have been avoided in our case, and we choose to use A* algorithm, which can generate optimal results. Our detailed selection matrix is in Table 3–1.

**Table 3–1 Selection matrix for path searching concepts**

| **Criterion** | **Weight** | **A*** | | | **RRT*** | | | **Potential Field** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Val | Score | Rate | Val | Score | Rate | Val | Score | Rate |
| Optimum | 0.1 | T | 10 | 1 | F | 0 | 0 | F | 0 | 0 |
| Path Quality | 0.3 | A | 10 | 3 | C | 0 | 0 | A | 10 | 3 |
| Stability | 0.3 | T | 10 | 3 | T | 10 | 3 | F | 0 | 0 |
| Runtime | 0.2 | Med | 5 | 1 | Fast | 10 | 2 | Slow | 0 | 0 |
| Simplicity | 0.1 | T | 10 | 1 | T | 10 | 1 | F | 0 | 0 |
| **Overall** | | | | **9** | | | 6 | | | 3 |

### (2) Concept selection for Angle measurement

We proposed 3 methods to measure the angle of the base, boom and stick joints:

- Orientation sensor
- Visual information
- Sonar sensor

Among these methods, using the orientation sensor is clearly the best choice. There exists many off-the-shelf solution for using orientation sensors to measure the angles. The angle measurement by visual information requires adding a camera as well as some special markers on the excavator system, which is much complicated compared to use orientation sensor. And measuring the angles by using a sonar sensor is also difficult and involve errors. Data provided by sonar sensor is a distance from the sensor to the obstacle. We need to apply geometry knowledge to derive the angles, which is difficult and involving errors. Therefore, we choose orientation sensor to measure angles. Our detailed selection matrix is in Table 3–2.

**Table 3–2 Selection matrix for angle measurement concepts**

| **Criterion** | **Weight** | **Orientation Sensor** | | | **Visual Info** | | | **Sonar Sensor** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Val | Score | Rate | Val | Score | Rate | Val | Score | Rate |
| Accuracy | 0.5 | Med | 5 | 2.5 | Low | 0 | 0 | Low | 0 | 0 |
| Availability | 0.3 | T | 10 | 3 | T | 10 | 3 | F | 0 | 0 |
| Ease to install | 0.2 | T | 10 | 2 | F | 0 | 0 | T | 10 | 2 |
| **Overall** | | | | **7.5** | | | 3 | | | 2 |

## (3) Concept selection for Excavator Arm Control

We proposed 3 methods to measure the angle of the base, boom and stick joints:

- Hydraulic Motor
- Pneumatic Motor
- Electric Motor

Among these three methods, we choose to use the hydraulic motor. The hydraulic motor system is the pre-installed system on the excavator model we received. Using the pneumatic motor or electric motor method both requires uninstalling the hydraulic motor and re-install the new motor system. None of our group members have mechanical engineering background, and this kind of hardware change is beyond our ability. Therefore, we choose the hydraulic motor. Our detailed selection matrix is in Table 3–3.

**Table 3–3 Selection matrix for arm control concepts**

| **Criterion** | **Weight** | **Hydraulic Motor** | | | **Pneumatic Motor** | | | **Electric Motor** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Val | Score | Rate | Val | Score | Rate | Val | Score | Rate |
| Ease to install | 0.4 | T | 10 | 4 | F | 0 | 0 | F | 0 | 0 |
| Stability | 0.3 | T | 10 | 3 | T | 10 | 3 | T | 10 | 3 |
| Ease to control | 0.2 | T | 10 | 2 | T | 10 | 2 | T | 10 | 2 |
| Accuracy | 0.1 | F | 0 | 0 | F | 0 | 0 | T | 10 | 1 |
| **Overall** | | | | **9** | | | 5 | | | 6 |

#### (4) Concept selection for Excavator Base Control

We proposed 3 methods to control the base of the excavator base:

- Pre-installed ESC + DC Motor
- L298N + DC Motor
- Servo Motor

Among these methods, we choose to use "L298N + DC Motor". The pre-installed "ESC + DC Motor" is the hardware choice provided by the excavator model. However, the ESC was designed for teleportation and we cannot find the right signal to control the ESC. The servo motor method, on the other hands, requires uninstall the installed DC motor on the base and re-install a servo motor. We are not sure whether we can fulfill this task. Therefore, we choose L298N + DC Motor, which is the DC Motor control method we are most familiar with. Our detailed selection matrix is in Table 3–4.

**Table 3–4 Selection matrix for base control concepts**

| **Criterion** | **Weight** | **ESC + DC Motor** | | | **L298N + DC Motor** | | | **Servo Motor** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Val | Score | Rate | Val | Score | Rate | Val | Score | Rate |
| Ease to use | 0.5 | F | 0 | 0 | T | 10 | 5 | T | 10 | 5 |
| Ease to install | 0.4 | T | 10 | 4 | T | 10 | 4 | F | 10 | 0 |
| Accuracy | 0.1 | F | 0 | 0 | F | 0 | 0 | T | 10 | 1 |
| **Overall** | | | | 4 | | | **9** | | | 6 |

In summary, for the motion planning and control part, the selected concepts we have now are

- Use A* to search for a collision-free path
- Use orientation sensor to measure angles
- Use hydraulic system to control excavator arm
- Use L298N + DC Motor to control excavator base

### 3.2.2 Perception Part

For the first concept (how to perceive surroundings), we decided to use binocular camera vision required by the sponsor. The path planning algorithm works upon a point cloud representation of the overall surroundings. But what our camera perceives is just from one angle. We have to rotate the camera to capture the surroundings from multiple perspectives.

How to fuse point clouds from multiple angles to form a overall representation of the environment becomes a question. The concepts we generate for that are Iterative Closest Point (ICP) algorithm and the simple coordinate transform. The ICP algorithm is basically to compute and apply a coordinate transform as well but it doesn't require any information given by us which is an advantage. But it has to compute the feature description of multiple pairs of point cloud, filter out noises, find the best matching ones and calculate a transform matrix. So it's relatively slow and its accuracy relies upon the property of the objects in the practical situations. However, during the camera rotation, we are able to get the servo's rotation angle and measure the distance between the camera and the rotation axis. Such information is useful to calculate a transform matrix. But it's obvious that the fusion result relies on the precision of the angle and distance measurement. We decide to use the latter approach because we are pretty confident on the precise control of the camera rotation. The detailed decision making is also shown in Table 3–5.

**Table 3–5 Decision matrix for point cloud fusion algorithm**

| Design Criterion | Weight Factor | ICP | | CT | |
|---|---|---|---|---|---|
| | | Score | Rating | Score | Rating |
| Accuracy | 0.6 | 7 | 4.2 | 9 | 5.4 |
| Efficiency | 0.2 | 5 | 1.0 | 7 | 1.4 |
| Precision Tolerance | 0.2 | 8 | 1.6 | 4 | 0.8 |
| **Overall** | | | 6.8 | | **7.6** |

If we successfully get a point cloud representation of the overall environment, we should allow our excavator to detect obstacles. Since in our case anything except the excavator arm itself can be considered as obstacles, the problem can be interpreted as identifying and removing the excavator arm while the rest is all the obstacle. There are various detection algorithms that we can choose from, such as CNN-based method, color-based method and distance-based method. Even though the CNN-based method might be relatively accurate, it requires us to collect hundreds of data, manually annotate them and train a CNN network which is quite time-consuming. Since the excavator arm itself has an unique color and it's very close to the camera, color-based and distance-based methods will perform well and be easy to implement. The detailed decision making is also shown in Table 3–6.

**Table 3–6 Decision matrix for arm detection algorithm**

| Design Criterion | Weight Factor | CNN | | Color+Distance | |
|---|---|---|---|---|---|
| | | Score | Rating | Score | Rating |
| Accuracy | 0.6 | 9 | 5.4 | 7 | 4.2 |
| Efficiency | 0.2 | 4 | 0.8 | 8 | 1.6 |
| Precision Tolerance | 0.2 | 4 | 0.8 | 8 | 1.6 |
| **Overall** | | | 7.0 | | **7.4** |

In summary, for the perception part, the selected concepts we have now are

- Use binocular camera vision to perceive surroundings

- Rotate the camera and fuse point clouds from multiple angles by coordinate transformation to form a representation of the surroundings
- Use color-based and distance-based approaches to eliminate excavator arms from the overall representation

# 4 Final Design

## 4.1 Overview of Chosen Concepts

In order that our excavator is able to move its robotic arm to the specified location without colliding with any obstacle in the path, we hope to combine the two individual parts,**Perception** and **Motion Control** into one compatible system, where the two subsystems could interact and cooperate efficiently. The hierarchy of the whole system is shown in Figure 4–1.

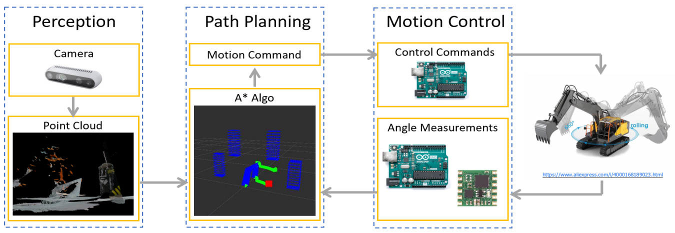



**Illustration 4–1 Final design concepts**

### 4.1.1 Motion Control Part

The final selected concepts to be implemented in the Motion Control part of our final prototype is shown in Figure 4–2. In general, to control the motion of the robotic arm of the excavator and do

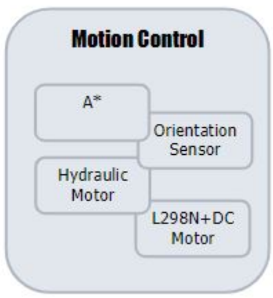



**Illustration 4–2 Selected concepts for motion control part**

path planning, we have included the design concepts in mainly four aspects.

- Motion Planning Algorithm - A*
- Angle Measurement Method - JY901 Orientation Sensor
- Arm Motion Method - Hydraulic System
- Base Motion Method - L298N + DC Motor

### 4.1.2 Perception Part

The final selected concepts to be implemented in the Perception part is shown in Figure 4–3.

In general, to perceive surroundings and detect excavator arms, we have included the design concepts in mainly four aspects.

- Camera Vision
- Camera Rotation
- Point Cloud Fusion - Coordinate Transform
- Arm Detection Algorithm - Color & Distance Segmentation

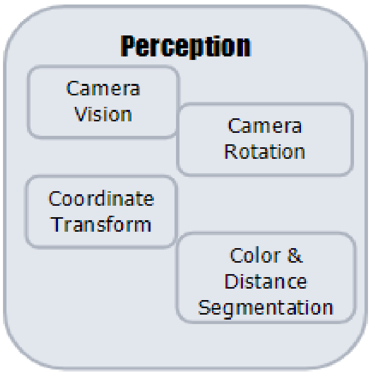



**Illustration 4–3 Selected concepts for perception part**

## 4.2 Engineering Design Analysis

### 4.2.1 Motion Control Part

The final engineering specifications of Motion Control Part that we are going to implement in the system are listed in Table 4–1.

**Table 4–1 Engineering specifications in motion control part**

| Engineering Specs. | Target Values | Units |
|---|---|---|
| Joint Angle Precision (boom & stick) | 5 | deg. |
| Joint Angle Precision (base) | 4 | deg. |
| Bucket Position Accuracy | 8 | cm |

To satisfy the specifications, we need to analyze the physical properties and complete the tasks in the following three aspects.

**Angle Measurement Precision & Orientation Sensor Selection** To choose a good orientation sensor that can meet the requirements, the most primary job is to satisfy the required angle measurement precision. At first, since our whole system is constructed and developed on the excavator model provided by our sponsor, there are already some intrinsic physical constraints. There only three Degrees of Freedom: the angle $\gamma$ between the current orientation of the base and the front of the excavator, the angle $\alpha$ between the boom and the horizontal plane, the angle $\beta$ between the stick and the horizontal plane. From experiments, we have determined the range of the three DoFs:

$$\begin{cases} -\pi/6 < \alpha < \pi/3 \\ -\pi < \beta < \pi/3 \\ -\pi/2 < \gamma < \pi/2 \end{cases} \tag{4–1}$$

To determine the appropriate precision of $\alpha$, $\beta$, $\gamma$, we need to involve **Probability Theories**.
Now, we assume that the angle would locate in the interval $[\mu - \delta, \mu + \delta]$, around the desired target value $\mu_0$, where $\mu$ is the average value. Due to symmetric properties, we further assume that $\mu_0 = \mu$. Then, we assume that the location of the angle would follow the normal distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{4–2}$$

To argue that we can control the motion of the arm precisely, we want to satisfy the following two criteria.

1. The probability for the angle to fall in the interval $[\mu - \delta, \mu + \delta]$ should not be too small. Quantitatively, we want $P(\mu - \delta < x < \mu + \delta) > 95\%$
2. The range of the total variance $2\delta$ should not be too large. Quantitatively, we want $2\delta/\Delta\theta < 12.5\%$, where $\theta$ refers to $\alpha$, $\beta$, or $\gamma$, $\Delta\theta$ is the corresponding total range.

According to the properties of the normal distribution, we have $P(\mu - 2\sigma < x < \mu + 2\sigma) \approx 95\%$. Therefore, we have

$$\delta > 2\sigma \tag{4–3}$$

Combing the listed criterion 2) and the inequality (3), together with our own academic experience, we have finally determined the engineering specifications for angle measurements in Table 4–1
Apart from meeting the angle precision conditions, the sensor should also address issues in another two fields: **Electromagnetic Disturbance** and **Signal Filtering**. Basically, the sensor is working in a surrounding that is filled with electromagnetic waves and metals. So, the components in the sensor may be magnetized and malfunctioned. Besides, the sensor works in a dynamic environment, so there is no chance that its input would be stable. How to filter out unnecessary noises and generate a stable output is also important.

After searching and comparing, we have eventually selected JY901 orientation sensor [1].It has a built-in integrated component to measure the three angles of the orientation of the sensor: Roll, Pitch and Yaw in a nearly purely mechanical way (see Figure. 4–4.

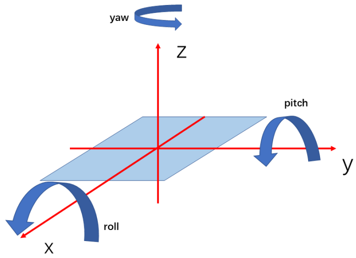



**Illustration 4–4 Angle measurements of the orientation sensor**

So, the influence of the electromagnetic disturbances are small.And it adopts Kalman Filtering algorithm (see [5]) to reduce the errors in processing the data. According to its manual, the general precision of the angle measurements can reach around 5°, so it serves as a perfect choice for our orientation sensor.

**Bucket Position Precision & Arm Motion Method** According to our engineering specifications, the error of the bucket position should be constraint into a range of 8 cm. This needs our power system to provide a sufficient and stable force to move the robotic arm of our system. From our experiments, we estimate that the weight of the whole stick is no smaller than 8 kg. To lift such a heavy robotic arm and make sure that it would not collapse easily during experiments, we can only select from two methods: hydraulic system or electric system. Here, the model adopts a hydraulic system to move its robotic and a DC motor to rotate its base. Notably, the way the model to rotate its boom and stick is a little tricky (See Figure 4–5 ).

**Illustration 4–5 Hydraulic cylinder to drive the excavator stick**

It is not using the hydraulic power directly to rotate the arm, but rather to provide the hydraulic pressure to a piston to control the contraction or extension of it. By moving the piston, it is able to pull or push the related part and make the boom or stick rotate around the pivot. This has applied another constraint on our design, that we could only consider ways to pull or push the arm as the model does initially to reduce workload and unnecessary troubles.
After we search on Internet, we noticed that the commercial electric motors online had a fatal flaw.

They could not be fixed onto our robotic arm. All the electric motors had two parameters: one is the initial length L and the other is the travel distance S. The range of the tip is given by $[L+S, L+2S]$. Now, the full range of the hydraulic bar on our model is [195,350] [cm]. To cover this range, we need

$$\begin{cases} L+S < 195 \\ L+2S > 350 \end{cases} \tag{4–4}$$

But all the items we found online does not satisfy the two inequalities, so finally we decided to use the hydraulic system on the model for the motion of boom and stick.

**Choices in Path Planning Algorithm** To implement the A* algorithm (see [2]), we need to specify several details of the whole algorithm to increase its efficiency and accuracy. Notably, the implementation of A* algorithm is to go through possible nodes and aims to minimize the function

$$f(n) = h(n) + g(n) \tag{4–5}$$

where n is the next node into consideration, which is to take out from the data structure called ”open list”, g(n) is the cost of the node with respect to the start point following the current path, and h(n) is the heuristic function that estimates the cost of the node n with respect to the goal node, following the potentially cheapest path. Besides the traditional A* components, in our system we also need to introduce the concept of ”Obstacle” so that our excavator arm can avoid the possible collisions. Therefore, we have to specify the following five attributes of the whole A* algorithm.

- Representation of Obstacles - KDTree. The raw data of the obstacle is obtained from the pointCloud. The works in our perception part will generate a pointCloud indicating the possible coordinates occupied by the obstacle. Then, we would like to use a KDTree structure (see [4]) to store all the data, because it is a very efficient data structure used in high-dimensional calculations.
- Implementation of Open List - Fibonacci heap. Generally, we need to take an element m out from the open list and evaluate the value f(m). To find out the most appropriate element in the open list to be added into the current path in a quick way, we have decided to implement the open list using a Fibonacci heap (see [3]). Typically, the Fibonacci heap is an efficient way to implement a priority list so we can find the node n with lowest f(n) value in the list directly without traversing the whole open list.
- Heuristic function h(n). Notice that we need to process several points in 3D space, so we hope to simplify the calculations. As a result, in Heuristic function, we simply use the L1 cost defined by

the following equation as Heuristic function.

$$L1(\vec{r}_1, \vec{r}_2) = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| \tag{4–6}$$

- Cost function g(n). To accommodate with our scenario with obstacles, our cost function g(n) consists of two parts. One is the direct L1-cost between the current node n and the start node. This measures the distance between the two nodes. The other is the so-called "close penalty" p, which is related to the minus three power of the closest distance d between the obstacle and the robotic arm as $p \propto d^{-3}$. In this way, the smaller d is, the higher the penalty is, and the larger the cost would be, and it would be less likely for the robotic arm to continue moving in the direction.
- Minimum Collision distance. To avoid collision, we have to set up a "collision distance" serving as the lower bound of the shortest distance between the obstacle and the robotic arm. It actually consists of two parts. One is due to the geometric properties of robotic arm. This is to ensure that the robotic arm would not collide with the obstacle no matter what the form of the obstacle is. After our experiments, we determine that the distance for this part is 0.12 m. The other is due to the inaccuracy of our control. Notice that there are existing deviations in our controlled DoFs, so the bucket position also has some deviations that needs to be considered. According to our engineering specification, the distance for this part is 0.08 m. In total, we set up **the minimum collision distance as 0.2 m**.

### 4.2.2 Perception Part

The engineering specifications related to perception are the perception range, the perception accuracy and the successful rate of obstacle avoidance. For the perception range, it must be no smaller than the workspace that excavator arm can reach. So the radius is no less than 1.3m. And the Intel RealSense D435 camera we use is capable of capturing objects within 0.3m ~ 10+ m so it satisfies this requirement perfectly.

The definition of perception accuracy shifts from the success rate of detecting one obstacle (DR1) to the success rate of detecting the excavator arm. The scientific field involved is still 3D object detection but for the current definition, because the excavator arm has an unique color and a relatively fixed position to the camera, it's easier for us to detect the arm based upon color information and depth information. But the evaluation of our algorithm is not easy to implement. For each test scenario, we have to manually label all the point clouds on the excavator arm. However, there are many noises generated by the camera because the arm is too close to the camera. Therefore, the accuracy of detecting arm points can not be too high. We define it as the ratio of true positive over all points labelled as excavator arm.

For the point cloud fusion algorithm, we must carefully choose the parameters that can be used to describe the transform. The rotation angle can be read from the servo and the distance between the camera and the rotation axis is 0.06m. Such information is enough to calculate the transform matrix.

For the arm detection algorithm, we utilize the color information and the distance information. Color segmentation requires a lower threshold and a upper threshold in the HSV color space. To segment out the color of the excavator arm, the lower threshold we determine is [100, 110, 46] and the upper threshold is [140, 255, 255]. For the distance information, the threshold is set to be 0.54 in the xy-plane.

### 4.2.3 Integration of Two Parts

The main engineering specification for the integration of perception and motion control part is the successful rate of obstacle avoidance. After we integrate these two parts, we will set up experiments in which the excavator system completes specific tasks and obtain the probability of avoiding obstacles successfully. The probability is expected to be above 85%.

## 4.3 Design Description

### 4.3.1 Motion Control Part

Figure 4–6 shows the design of our excavator prototype based on the 1/12 excavator model provided by the sponsor. See appendix for link to the excavator model. All wirings are omitted for simplicity.

A total of four JY901 modules are used to measure different angles. Three of them are fixed on the excavator arm, with two on the stick and one on the boom. Two JY901 modules are used on the stick, with one placed perpendicular to the other, because the readings are found unstable around -90 degrees. Therefore, another module is placed beside the primary one to compensate under this scenario. The last JY901 is fixed on the base to measure the base rotation.

The infrared camera module, Intel RealSense D435, is connected to the bracket with a servo such that the camera can rotate and sweep horizontally to scan the surroundings, as shown in Fig. 4–7.

The L298N module is used to control the DC motor that turns the base. The original excavator model used an electronic speed controller (ESC) to control the motor, but the L298N served the purpose better.

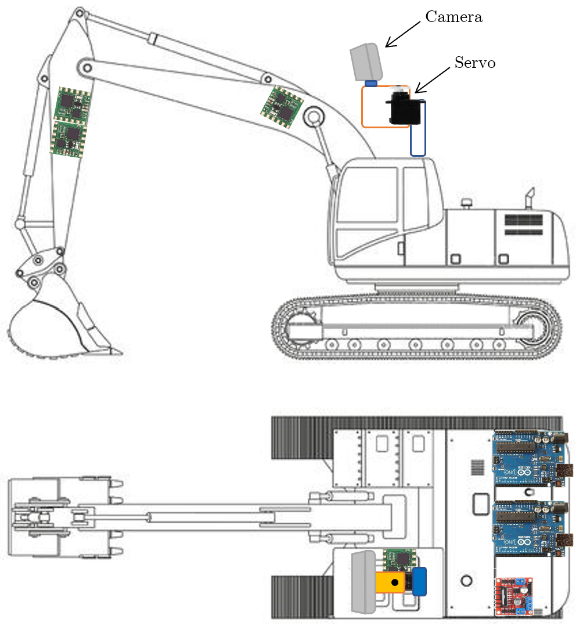



**Illustration 4–6  Side and top view of the excavator model**

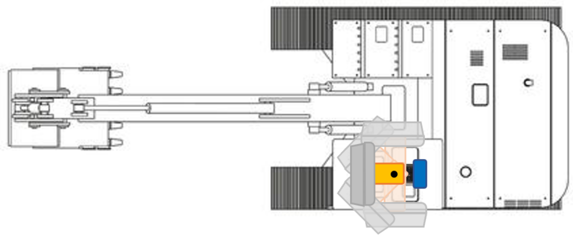


**Illustration 4–7  Camera rotation**

Two Arduino Uno boards are used in this prototype design. One is for obtaining angle readings from all four JY901 modules following the I2C protocol, and the other is used to send out controlling signals to the three servos controlling the selector valve (from the original excavator model), the servo controlling the rotation of the camera, the oil pump (from the original excavator model), and the L298N module.

Figure 4–8 shows the detailed design diagram of the entire system, and the components that come with the excavator model are highlighted in yellow.

Note that although the two servos from the selector valve came with the excavator, we decided to replace the original servos with TBS-K20 servos due to potential slow responses of the specific servo model. The servo under the camera sweeps for 180 degrees from left to right to let the camera scan the surroundings, and all frames are sent back to the host to generate a point cloud. The host applies A* algorithm to search through the point cloud and determines a collision-free path. The host then

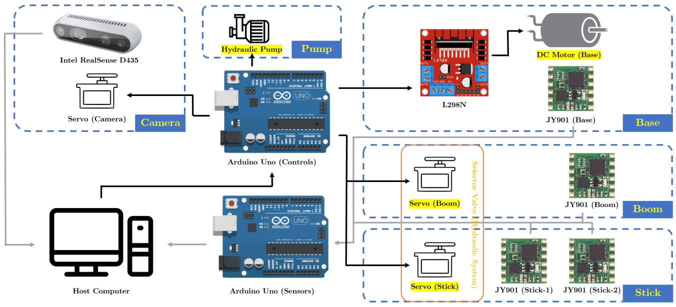



**Illustration 4–8 Detailed design diagram**

calculates actual commands and sends the commands through the Arduino to actuate the pump and selector valve to control the excavator arm. The L298N then receives signal from the same Arduino board to rotate the base to the desired position. By following each target on the planned path, the excavator achieves autonomous obstacle avoidance. The list of items used in this project and their costs are summarized in Table 4–2.

**Table 4–2 Item list and prices**

| Item | Unit Price(¥) | Amount | Total Price(¥) |
|---|---|---|---|
| JY901 Module | 102.00 | 5 | 510.00 |
| L298N Module | 8.99 | 1 | 8.99 |
| Intel RealSense D435 | 1550.00 | 1 | 1550.00 |
| TBS-K20 Servo | 75.25 | 2 | 150.50 |
| **Total** | | | **2219.49** |

### 4.3.2 Perception Part

The perception algorithm consists of two parts

- Point cloud fusion algorithm (coordinate transform)
- Detection algorithm (color & distance approach)

For the first part, Figure 4–9 shows the diagrams from two adjacent point clouds (the first two rows) to a fused point cloud which is iteratively applied to a series of point clouds.

After fusing all the point clouds, the arm detection method will be applied shown as Figure 4–10. The detected arm will be removed.

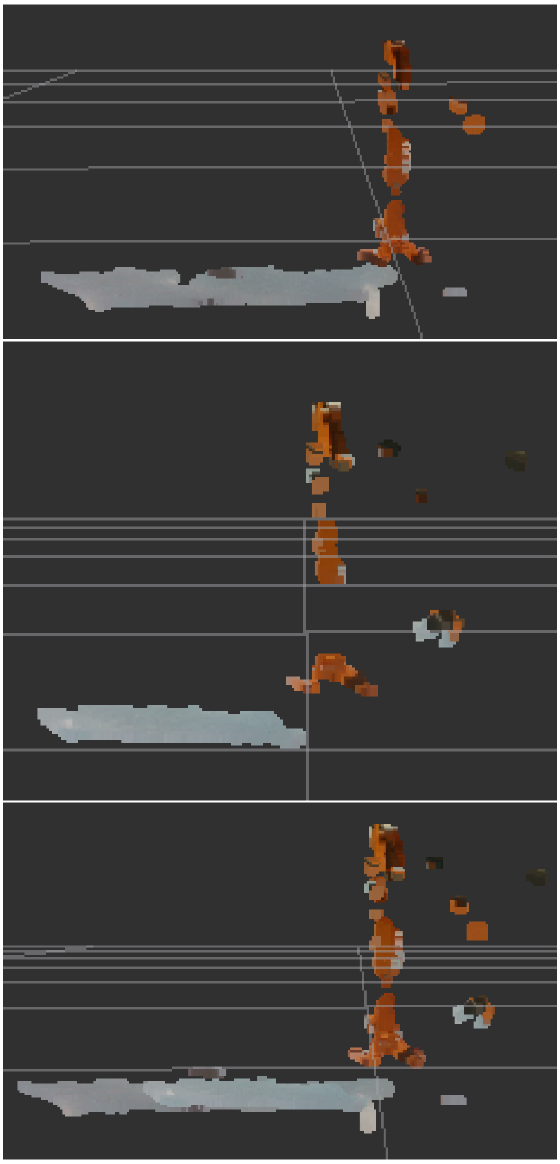


**Illustration 4–9  Point cloud fusion result**

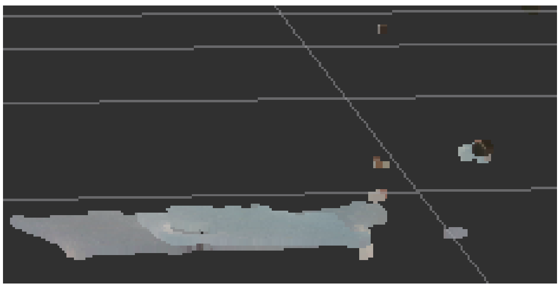


**Illustration 4–10  Excavator arm segmentation**

# 5 Implementation and Validation

## 5.1 Implementation Plan

The excavator model itself is provided by the sponsor, and all other components listed in Table 4–2 can be purchased off the shelf. Currently, the total cost is around ¥2219.49, well below the original budget of ¥4000. Figure 5–1 illustrates our final prototype for the autonomous excavator with obstacle avoidance.

**Illustration 5–1 Final excavator prototype**

The perception algorithm and the path planning A* search algorithm are programmed in Python, and the communication network to handle readings, send commands, and visualize the entire setup is done by Robot Operating System (ROS), and the whole system runs on Ubuntu 18.04 system.

## 5.2 Validation Plan

### 5.2.1 Motion Control Part

To validate that our system meet the engineering specifications, we have adopted two different methods to deal with them separately.

**Validation on Base, Boom, Stick Angles** The precision of the angle measurements are 5° for boom and stick angles and 4° for the base angle. Now, it is easy to see that the values of the angles can only locate in a discrete number of intervals. For example, the range of the boom angle is [0, 60°], with a precision of 5°. Involving the potential deviation, the boom angle can only locate in the intervals [0, 5°], [5°, 10°] ... [55°, 60°]. Our plan to validate this part is to **traverse each DOF and each**

**possible interval of value** and use protractor to verify that the arm and the base have successfully reach the desired precision. For example, we hope to command the boom angle to 42.5° and then use the protractor to verify that it locates in the interval [40°, 50°]. In such way, we need to test 10 values for boom, 24 values for stick together, and then test the 45 values for base separately since the motion of the base is relatively independent. In total, we want to test 10*24+45 possible values of the angles to validate that the system works successfully.

**Validation on Bucket Position** The possible locations of the bucket may be too many to handle one by one. Theoretically, the bucket position can locate at any point in the space. After our calculation and estimation, we think that we may need to test over 1000 values. Also, the bucket itself occupies some bulk that may influence our experiments.
Hence, we want to validate our bucket position accuracy by direct theoretical calculation. Check the illustration in Fig. 5–2

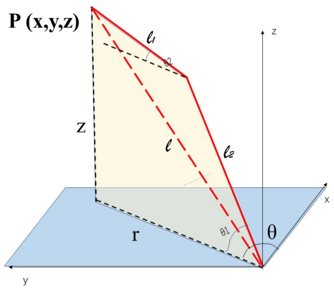



**Illustration 5–2 Relationship between spatial coordinates and DoFs**

Thus, we can have a relationship between the coordinate in space (x,y,z) and the three DoFs ($\alpha$, $\beta$, $\gamma$). Then, we can obtain the deviation of the bucket position along three directions with the following formulas:

$$\begin{cases} \Delta x = \frac{\partial x}{\partial \alpha}\Delta\alpha + \frac{\partial x}{\partial \beta}\Delta\beta + \frac{\partial x}{\partial \gamma}\Delta\gamma \\ \Delta y = \frac{\partial y}{\partial \alpha}\Delta\alpha + \frac{\partial y}{\partial \beta}\Delta\beta + \frac{\partial x}{\partial \gamma}\Delta\gamma \\ \Delta z = \frac{\partial z}{\partial \alpha}\Delta\alpha + \frac{\partial z}{\partial \beta}\Delta\beta + \frac{\partial x}{\partial \gamma}\Delta\gamma \end{cases} \tag{5–1}$$

Finally, using the formula below, we can obtain the deviation of the bucket position in terms of the deviation length.

$$\Delta L = \sqrt{\Delta^2 x + \Delta^2 y + \Delta^2 z} \tag{5–2}$$

After calculations, if the final $\Delta L$ is smaller than 8 cm, which is specified in our engineering specifications, then we will say that our design has met the criteria. The detailed calculation and the corresponding matlab codes can be seen in **Appendix**.

### 5.2.2 Perception part

As speculated in the engineering logic part, our main focus on the perception part has changed to eliminate excavator arm from the point cloud representation.

**Validation on Perception Accuracy** In Section "Engineering Design Analysis", we has define the perception accuracy to be the ratio of the true positive over all the points labelled as excavator arm. In the integral point cloud representation, we will manually label out all the excavator arm points, which are defined as the ground truth points for the excavator arm. And after we segment out the excavator arm points using the combined color-based method and distance-based method, we would got a set of points being eliminated. The true positive is thus defined as the intersection between the eliminated set and the ground truth set. Define the intersection set as $I$, and the ground truth set as $G$. And our overall accuracy is defined as follow:

$$\text{Perception Accuracy} = \frac{|I|}{|G|} \tag{5–3}$$

We expect this value to be over 80% in every test scenario.

**Validation on Perception Range** Since we use Intel RealSense D435 as our perception tool, from its official website, we could get the parameters of the camera, listed as in the above table.

**Table 5–1 Camera depth parameters**

| Features | Values | Units |
|---|---|---|
| Maximum range | 3 | m |
| Minimum depth distance at max resolution | ~28 | cm |

And our expected perception range is within the maximum range of the camera, thus satisfying our needs.

# 6 Validation Results

In this section, we will introduce the validation and testing of our model in more details. Validation tests are conducted for motion controls and perception respectively.

## 6.1 Motion Control & Path Planning

For the excavator model controls, we have traversed all valid angles of the base, boom and stick of the excavator and measured the difference between the assigned and actual angles to check against the identified engineering specifications in Table 4–1. The table is reproduced below for reader's convenience. Meanwhile, the success rate of obstacle avoidance is also counted to validate the target success rate of 80%.

**Table 6–1 Engineering specifications in motion control part**

| Engineering Specs. | Target Values | Units |
|---|---|---|
| Joint Angle Precision (boom & stick) | 5 | deg. |
| Joint Angle Precision (base) | 4 | deg. |
| Bucket Position Accuracy | 8 | cm |

Refer to **Appendix** for all collected data during the tests. The average, maximum and minimum angle differences are summarized in Table 6–2.

**Table 6–2 Summary of motion control test results**

| Control Parts | Target Values | Mean Dev. | Max. Dev. | Min. Dev. | Units |
|---|---|---|---|---|---|
| Base | 4 | 1.81 | 3.82 | 0.16 | deg. |
| Boom | 5 | 1.71 | 2.70 | 0.65 | deg. |
| Stick | 5 | 1.96 | 4.84 | 0.06 | deg. |

As indicated in the table, we have satisfied the requirements of base, boom and stick joint angle precision listed in Table 6–1. By following the process discussed in section 5.2.1, we obtain the final mean bucket position accuracy to be 2.8 cm, which also falls below the 8 cm requirement under the engineering specifications.

To validate our success rate of obstacle avoidance, a total of 12 random target bucket positions are sent to the system, and 11 out of the 12 operations are successful. The bucket slightly touches an obstacle in one specific test due to the set target being in close proximity of an obstacle and limited control precision of the hydraulic system. This leads to an approximate success rate of 90%, well above the targeted 80%.

## 6.2 Perception

The engineering specs on the perception range is already satisfied by the Intel RealSense D435 camera and its built-in parameters are shown above.

The remaining specs required to be validated is only the perception accuracy. The evaluation metric is defined as Eq(5-3). And for five validation scenarios, we calculate the accuracy results shown in Table 6–3.

**Table 6–3 Validation Result for Perception Accuracy**

| **Trial** | 1 | 2 | 3 | 4 | 5 | **Avg** |
|---|---|---|---|---|---|---|
| **Accuracy** | 80.5% | 84.3% | 82.1% | 88.4% | 81.3% | 83.3% |

All the accuracy we calculate for those test scenarios satisfy the engineering specification.

# 7 Discussions and Conclusion

## 7.1 Discussions

### 7.1.1 Achievements

At the time of writing, we have realized precise control of the provided excavator model, developed full set of algorithms to perceive the surroundings and detect obstacles and to plan a collision-free path for the excavator to follow. Our excavator prototype is capable of accepting user input for a bucket target position and autonomously finish the operation to reach the target.

### 7.1.2 Potential Problems

Since our project involves actuating a real 1/12 excavator model, most of the potential problems we are facing with are hardware problems. Table 7–1 summarizes possible hardware problems and our preparation for such problems.

**Table 7–1 Potential problems**

| Problems | Solutions |
|---|---|
| Slow response on servos for hydraulic selector valve | Prepare spare servos |
| JY901 sensors malfunctioning | Prepare spare JY901 sensors |
| Loose connection of wires | Use glue to stabilize wires |
| Hydraulic pipeline leakage | Prepare spare parts for the hydraulic system |

## 7.2 Conclusion

In conclusion, we have created an excavator system that can capture static surrounding environment and put the bucket of the excavator to desired target without collision. Our work has met the customer's requirement. We are highly confident in the promising future of our project idea as it reduces unnecessary casualties due to complex terrain or inexperience of excavator operators.

## 7.3 Future Work

Considering our current progress and achievements, we recommend the following aspects to be considered for further research and exploration.

### 7.3.1 Real-time Obstacle Avoidance

Our current prototype scans the surrounding environment merely at the beginning of each operations, yet the possibility of sudden intruders in the working area should not be overlooked. Due to the hardware and time limitations, we are unable to implement real-time obstacle avoidance, but it would be a substantial improvement to have the feature integrated to the system.

### 7.3.2 Control Precision

Due to the nature of the hydraulic system on the excavator model we have received, it makes little difference to apply more complicated control algorithms such as PID to strive for a higher control precision. Nevertheless, each a couple inches of error can mean a lot when our system is adapted and applied to real-world excavators. Therefore, additional control algorithms are recommended to be added to the system for use under real working environment.

# 8 References

[1] Corporate. Wit Motion. Manual for JY901(S) sensor [EB/OL]. 2016[2021-06-10]. https://dl.wit-motion.com:2103/index.html#/markdown/preview?document=37ec3ecfdc804a79a1fb99ea215b3913

[2] Shushanghoujue. A-Star Algorithm [EB/OL]. 2019.10.09[2021-07-03]. https://blog.csdn.net/qq_36931982/article/details/102465846

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. T. Introduction to Algorithms [DB/OL]. 1990.03[2021-07-03]. http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap21.htm

[4] Quansilang. Detailed Explanation on KDTree [EB/OL]. 2014.11.26[2021-07-03]. https://blog.csdn.net/silangquan/article/details/41483689

[5] Tongxinchengxuyuan. Intro to Kalman Filtering [EB/OL]. 2016.07.30[2021-07-15]. https://blog.csdn.net/u011285477/article/details/52070900

[6] M. Young, The Technical Writer’s Handbook [M]. Mill Valley, CA: University Science, 1989.

# Acknowledgements

To begin with, we would like to present our gratitude towards Zishi Li from BuilderX and Professor Chengbin Ma from the Joint Institute, without whom our project would not have been possible.

Besides, we would like to extend our greatest appreciation to Dr. Yu Sun from the Circuit Lab of the Joint Institute. His insightful suggestions and assistance to our hardware issues along the way have been an indispensable contribution to our project. The experience and inspiration he brought about are something we will forever treasure. We would also like to thank Dr. Mingjian Li from the mechanical engineering lab for his kind suggestions for our hydraulic system.

In addition, we would like to thank the rest of VE450 faculty team for all the knowledge they have delivered in terms of large-scale engineering project managements.

We would like to thank everyone who has helped and made this project possible.

Moreover, thanks to SJTUTHESIS Team for providing the LaTeX template for this thesis document.

# Appendix A Appendix

## I.1 Validation Test Measurements

**Table I–1 Joint angle precision (base) test results (deg.)**

| Target Angles | Measured Angles | Deviations |
|---|---|---|
| -90.00 | -92.30 | 2.30 |
| -80.00 | -81.43 | 1.43 |
| -70.00 | -70.25 | 0.25 |
| -60.00 | -61.42 | 1.42 |
| -50.00 | -50.20 | 0.20 |
| -40.00 | -42.25 | 2.25 |
| -30.00 | -32.16 | 2.16 |
| -20.00 | -22.76 | 2.76 |
| -10.00 | -13.42 | 3.42 |
| 0.00 | -0.68 | 0.68 |
| 10.00 | 13.82 | 3.82 |
| 20.00 | 19.84 | 0.16 |
| 30.00 | 33.68 | 3.68 |
| 40.00 | 40.88 | 0.88 |
| 50.00 | 52.66 | 2.66 |
| 60.00 | 61.57 | 1.57 |
| 70.00 | 70.47 | 0.47 |
| 80.00 | 82.26 | 2.26 |
| 90.00 | 91.93 | 1.93 |
| | **Dev. Mean** | 1.81 |
| | **Dev. Max.** | 3.82 |
| | **Dev. Min.** | 0.16 |

**Table I–2 Joint angle precision (boom) test results (deg.)**

| Target Angles | Measured Angles | Deviations |
|---|---|---|
| 20.00 | 19.35 | 0.65 |
| 25.00 | 23.15 | 1.85 |
| 30.00 | 28.70 | 1.30 |
| 35.00 | 32.85 | 2.15 |
| 40.00 | 38.33 | 1.67 |
| 45.00 | 42.30 | 2.70 |
| 50.00 | 48.36 | 1.64 |
| | **Dev. Mean** | 1.71 |
| | **Dev. Max.** | 2.70 |
| | **Dev. Min.** | 0.65 |

**Table I–3 Joint angle precision (stick) test results (deg.)**

| Target Angles | Measured Angles | Deviations |
|---:|---:|---:|
| 30.00 | 31.12 | 1.12 |
| 35.00 | 31.12 | 3.88 |
| 40.00 | 40.56 | 0.56 |
| 45.00 | 45.83 | 0.83 |
| 50.00 | 50.70 | 0.70 |
| 55.00 | 57.34 | 2.34 |
| 60.00 | 57.66 | 2.34 |
| 65.00 | 68.30 | 3.30 |
| 70.00 | 78.15 | 1.85 |
| 75.00 | 76.83 | 1.83 |
| 80.00 | 77.40 | 2.60 |
| 85.00 | 86.20 | 1.20 |
| 90.00 | 88.00 | 2.00 |
| 95.00 | 95.73 | 0.73 |
| 100.00 | 98.66 | 1.34 |
| 105.00 | 104.94 | 0.06 |
| 110.00 | 105.40 | 4.60 |
| 115.00 | 118.76 | 3.76 |
| 120.00 | 120.30 | 0.30 |
| 125.00 | 120.16 | 4.84 |
| 130.00 | 130.90 | 0.90 |
| | **Dev. Mean** | 1.96 |
| | **Dev. Max.** | 4.84 |
| | **Dev. Min.** | 0.06 |

## I.2 MATLAB Code for Bucket Position Accuracy Calculation

**Code**:

```
clear;clc
l1=0.482; l2=0.255;
d=0.04;

alpha_min = pi/9;
alpha_max = pi*5/18;
beta_min = -13/18*pi;
beta_max = pi/2;
gamma_min = -pi/2;
gamma_max = pi/2;

num_steps = 100;

step_alpha = (alpha_max - alpha_min) / num_steps;
step_beta = (beta_max - beta_min) / num_steps;
step_gamma = (gamma_max - gamma_min) / num_steps;
alpha = [alpha_min:step_alpha:alpha_max];
```

```
beta = [beta_min:step_beta:beta_max];
gamma = [gamma_min:step_gamma:gamma_max];
d_angle = pi/90;
x_alpha = cos(gamma).*(-sin(alpha)).*l1;
x_beta = cos(gamma).*(-sin(beta)).*l2;
temp = l1*cos(alpha)+l2*cos(beta)+d;
x_gamma = (-sin(gamma)).*temp;
dev_x = (x_alpha+x_beta+x_gamma)*d_angle;
y_alpha = (-sin(alpha)).*sin(gamma).*l1;
y_beta = (-sin(beta)).*sin(gamma).*l2;
y_gamma = cos(gamma).*temp;
dev_y = (y_alpha+y_beta+y_gamma)*d_angle;
dev_z = l1*cos(alpha)+l2*cos(beta);
dev_z = dev_z * d_angle;
dev = sqrt(dev_x.^2 + dev_y.^2 + dev_z.^2);

fprintf("max deviation: %f \n",max(dev))
fprintf("avg deviation: %f \n",sum(dev)/length(dev))
fprintf("min deviation: %f \n",min(dev))
```

**Output:**

```
max deviation: 0.034109
avg deviation: 0.028484
min deviation: 0.015324
```

# Appendix B Bios

## II.1 Shiji Liu


| | |
|---|---|
| **Affiliation** | B.S. ECE at JI and CE at UM |
| **Stud. ID.** | 517370910170 |
| **Email** | shijiliu@sjtu.edu.cn |
| **Phone** | +86 13801965280 |
| **Programming** | C++, Python, MATLAB |


**Short Bio** Shiji Liu received his B.S. from the University of Michigan in Computer Engineering in 2021, and is currently an undergraduate student studying Electrical & Computer Engineering at UM-SJTU Joint Institute. Shiji is mainly responsible for developing the ROS network and implementing the path planning algorithm in this project.

**Future Plan** Shiji Liu will work as a Research and Development engineer at BuilderX.

## II.2 Jinjie Liu


| | |
|---|---|
| **Affiliation** | B.S. ECE at JI and CE at UM |
| **Stud. ID.** | 517370910142 |
| **Email** | jj-jdfz@sjtu.edu.cn |
| **Phone** | +86 13636488100 |
| **Programming** | Python, C/C++, Javascript |


**Short Bio** Jinjie Liu received his B.S. from the University of Michigan in Computer Engineering in 2021, and is currently an undergraduate student studying Electrical & Computer Engineering at UM-SJTU Joint Institute. Jinjie is mainly responsible for working on the hardware issues and implementing the path planning algorithm in this project.

**Future Plan** Jinjie will continue to work towards his master's degree in ECE at the University of Michigan and step into industry afterwards.

## II.3 Xun Tu



| | |
|---|---|
| **Affiliation** | B.S. ECE at JI and CE at UM |
| **Stud. ID.** | 517370910149 |
| **Email** | ramseytaler@sjtu.edu.cn |
| **Phone** | +86 15221307531 |
| **Programming** | C++, C, Python, Matlab |



**Short Bio** Xun Tu received his B.S. from the University of Michigan in Computer Engineering in 2021, and is currently an undergraduate student studying Electrical & Computer Engineering at UM-SJTU Joint Institute. Xun is mainly responsible for angle measurements and implementing forward and inverse kinematics for the path planning algorithm in this project.

**Future Plan** Xun is pursuing master degree in Electrical and Computer Engineering in University of Michigan, Ann Arbor, and then search for a job in industry

## II.4 Yiwei Zhang



| | |
|---|---|
| **Affiliation** | B.S. ECE at JI and CS at UM |
| **Stud. ID.** | 517370910161 |
| **Email** | stupager@sjtu.edu.cn |
| **Phone** | +86 13918772259 |
| **Programming** | C++, Python, Java |



**Short Bio** Yiwei Zhang received his B.S. from the University of Michigan in Computer Science in 2021, and is currently an undergraduate student studying Electrical & Computer Engineering at UM-SJTU Joint Institute. Yiwei is mainly responsible for developing excavator perception in this project.

**Future Plan** Yiwei will finish her master degree and secure a job in the industry

## II.5 Pengyuan Huang

| | |
|---|---|
| **Affiliation** | B.S. ECE at JI and CS at UM |
| **Stud. ID.** | 517370910025 |
| **Email** | 1140833585hpy@sjtu.edu.cn |
| **Phone** | +86 18621812875 |
| **Programming** | C++, Python, Go |

**Short Bio** Pengyuan Huang received his B.S. from the University of Michigan in Computer Science in 2021, and is currently an undergraduate student studying Electrical & Computer Engineering at UM-SJTU Joint Institute. Pengyuan is mainly responsible for developing excavator perception in this project.

**Future Plan** Pengyuan is pursuing a Master of Science in Computer Science in Georgia Technology Institute, and then probably looking for a job in industry.

# 上海交通大学 毕业设计（学士学位论文）
# 单独工作报告

**SHANGHAI JIAO TONG UNIVERSITY**

**CAPSTONE DESIGN (BACHELOR’S THESIS)**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name：Shiji Liu

Student ID Number：517370910170

Major: Electrical and Computer Engineering

## II.6 Shiji Liu's Individual Contribution Report

### Contributions

I played two roles in the team:

- team leader
- technical expert

As the team leader, I was responsible for

- holding regular laboratory sessions for the team to work together on the project
- working out progress plan for the team, and supervise the progress of the team
- holding weekly meetings with my team members and Professor Chengbin Ma to report our progress
- offering regular report to our sponsor BuilderX

As the technical expert, I participated in almost all technical achievements in the project:

- dividing the project into perception and motion planning & control part
- working out the excavator arm hydraulic control system
- working out the excavator base turning system
- working out the motion planning algorithm selection and implementation (selected and implemented A* algorithm)
- working out the installation plan for the camera
- working out the fusion algorithm for local pointclouds
- working out the integrated autonomous excavator obstacle avoidance system

### Inspiration and Exploration

As stated before, I participated in working out almost all technical achievements in the project. Many of them involve inspiration and exploration.

One of the key tasks for our project is to utilize the rgbd camera to perceive the surrounding environment. To perceive the environment, two sub-tasks must be finished:

- install the camera, such that the camera can perceive the surrounding
- get the point-cloud representation of the environment based on the information captured by the

camera

Since the angle of view of the camera we used is limited, we cannot get the point-cloud representation of the whole environment with only one camera shot. Our sponsor Zishi Li from BuilderX offered us with an idea that we can overcome this issue by rotating the excavator base, allowing the camera to view the whole work space. Since directly rotate the excavator base may lead to collision, we rejected this idea. However, inspired by the 'rotation' idea, I proposed a plan that we can add a servo-motor under the camera. By rotating the servo-motor, the camera will rotate accordingly and being able to capture the information of the whole work-space.

After we have got the local point-cloud representations by rotating the camera, we need to align these representations to form a global representations of the environment. Initially, my team-mates Pengyuan Huang and Yiwei Zhang tried the "Iterative Closest Points" algorithm and "RANSAC" algorithm. However, the algorithms work poor in our case. Inspired by the fact that the point-cloud representation taken by the camera can be considered as precise, and by the fact that the pos of the camera when taking each shot is known with high precision, I proposed an idea to directly fuse the environment by using the coordinate transformation. It turns out that the simple coordinate transformation can produce a result much better than the ICP + RANSAC method.

Another key task for our project is to choose an algorithm that can form the collision free path. Initially, we selected RRT*, a random-based path planning algorithm. However, judging the fact that our excavator system only involves 3 degree of freedom, I proposed that we can apply A* instead of RRT*. A* is an optimal-first search algorithm, which can guarantee an optimal output, and is fast when the dimension of the search space is low. My teammates agreed with my idea, and Jinjie Liu and Xun Tu and I implemented different components of the A* algorithm. It turns out that my idea was successful: A* can find us with a collision free path within a tolerable amount of time.

After we have implemented A*, we need to control the excavator arm to follow the planned path. We encountered a problem that the stick angle cannot be stablized within the range of 40-60 degrees.Xun Tu gives a reasonable explanation of the cause of this problem: the angle sensor on the stick may become unstable when its x direction is having a small angle difference with the direction of the gravity vector. Inspired by Xun Tu's explanation, I proposed that he can use two angle sensors to measure the stick angle, when one sensor may become unstable, use another sensor to get the angle. Inspired by my idea, Xun Tu came up with the final solution that we install 2 angle sensors perpendicular to each other, use the reading from one sensor when it is stable, and the reading from another sensor when the first one is unstable. The stick issue was resolved after Xun Tu implemented this method.

## Skills and Knowledge Acquisition

This project greatly enhances my ability in engineering management and communication. This is the first time in my undergraduate time that I created a successful project in the role of the team leader. By having this project, I understood the importance of

- working out a feasible plan for the project
- checking whether the progress meets the plan regularly, and regularly adjust the original plan
- communicating with other engineers in the field and looking for their advice on the project
- distributing labor to each teammates

This project also greatly enhances my ability in dealing with hardware issues. I and Jinjie Liu, Xun Tu spent more than a month to actuate the excavator model. We

- replaced the pre-installed ESC on the model to the ESC we are familiar with
- found the user guide for the pump motor and understood the usage of the hydraulic system by experiment
- select angle measurement sensors and install sensors on the model

By going through this process, I summarized several experiences that may be useful for future work with hardware:

- Keep patience and keep caution
- Find user guide for each hardware components and read them thoroughly. Many hardware problems can be resolved or avoided by reading the user guide thoroughly.
- Apply control variable method to find explanation to issues
- Do not stick to a single solution to the problem, try different solutions, and come up with a solution that may not induce the problem if necessary.

In conclusion, I consider my work for this capstone design project to be successful. I have greatly enhanced my technical skills and soft skills, which should be beneficial for my work in the future.

# 上海交通大学 毕业设计（学士学位论文）
# 单独工作报告

**SHANGHAI JIAO TONG UNIVERSITY**

**CAPSTONE DESIGN (BACHELOR’S THESIS)**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name：Jinjie Liu

Student ID Number：517370910142

Major: Electrical and Computer Engineering

## II.7 Jinjie Liu's Individual Contribution Report

Due to the nature of this project, the work is mainly divided into two halves in our group, namely the perception part and the motion control & path planning part. Yiwei Zhang and Pengyuan Huang are assigned to the perception half due to their stronger computer science background, and Shiji Liu, Xun Tu and I are responsible for both bsic motion controls of the provided excavator model and the path planning algorithm implementation and realization.

Shiji is our team leader and takes perhaps the heaviest workload. He takes charge of the communication between the team, the faculty advisors and the sponsor. He is also responsible for constructing the underlying framework of ROS that handles the communication of perception, motion controls and path planning. Xun devotes most of his energy to the testing and adaptation of orientation sensors, as well as the code implementation of forward and inverse kinematics for our path planning part. Yiwei and Pengyuan take care of testing different camera models and implementation of point cloud fusion algorithms. Additionally, Yiwei prepares the testing environment of the system.

For me, I participate mostly in the motion control and path planning parts. Since the excavator model our team has received from the sponsor lacked proper documentation, we struggled a lot during the starting phase of the project. Our first major task was to drive the hydraulic pump. The pump appeared to be connected to and controlled by an electronic speed controller (ESC), but we could not drive the pump as we sent PWM signals to the ESC. The starter code provided by our sponsor indicated a range of 5.0% to 10.0% of PWM signal for pump control, yet none of the values between 4.0% to 8.0% worked. It occurred to me that other users might have also encountered similar issues, so I searched online. An online post recorded an identical problem and reported that only the maximum PWM signal successfully powered on the pump. Sure enough, we also succeeded in starting the pump with maximum PWM.

Our next challenge was to control the rise and fall of the excavator arm (boom & stick). Our first hypothesis was that the positions of servos on the selector valve determine the exact positions of the boom and stick. However, it didn't appear to be the case after we successfully turn on the pump. The boom and stick kept rising till their extreme positions when the servos were fixed at a position, and to make matters worse, we identified a leaking spot in the hydraulic pipeline. As the selector valve did not have a specific model type, it was much harder to find reference documentations on the usage of the valve. I was eventually able to find an image illustrating the mechanisms inside such valves, which gave rise to a new hypothesis that the angle of the servos controls the rise/fall of the boom and stick. This new hypothesis was verified as I turned on the pump and manually rotate the selectors to see the motions of the arm.

Yet the fluid leakage remained to be resolved. Another connector started to leak and even ended up with a small explosion of hydraulic fluid after we tried to fix the first one with an O-shaped ring. After taking the first leaking connector apart, we identified a broken sealing ring, and the leakage was resolved by replacing the connector. We believed that the explosion resulted from excessive pressure inside the hydraulic pipes, since the pump was rotating at a high rate of speed. We reflected upon our initial attempt to power on the pump, and we agreed that something was still wrong. Theoretically, we should be able to control the speed of the pump. It was much easier to find the user’s manual of the ESC this time, and I finally located the section discussing the setup of the ESC. After following each step carefully, we successfully set up the full range of control signals on the ESC and were able to gain full control over the rotation speed of the pump. With the pump running at minimum working power, both the leakage and explosion issues were settled.

As for the path planning section, Shiji, Xun and I split the work to implement the A* algorithm. Xun implemented forward and inverse kinematics to determine the exact location of the bucket while I implemented the configuration space creation and expanding algorithm of A*. On the other hand, Shiji handled the cost calculation/estimation and fusion of the three parts. The collaboration on the coding side went rather smoothly, and the software part turned out to be much easier compared to our struggles with the hardware issues on the excavator model. It took us about three days to implement the algorithm and conduct validation tests with different test cases to confirm the validity of the path planning algorithm.

When it comes to the integration of perception and motion control/path planning, directly connecting everything to power without any prior tests was risky and to some extent dangerous as the hydraulic system could create excessive strength on the excavator arm, leading to property damage or even personal injuries. As a result, I suggested testing without power first, by connecting controlling signals to an oscilloscope instead of actual components and manually moving the excavator arm to see the commands that the system would send out. It turned out to be a sensible decision and helped us avoid multiple potential test failures.

Overall, this project has profoundly enhanced my abilities in handling large-scale engineering projects. The early struggles we suffered demonstrate the importance of online resources as well as official user manuals. Some of the problems that puzzled us could have been avoided if we had sought help from other resources earlier. In addition, it is always better to ask for advice from as many people as possible. Although our team is divided into two halves for most of the time, the advice from Yiwei and Pengyuan could also be valuable from time to time. The A* algorithm we have implemented is one of the most widely adopted path planning algorithms in robotics and game development. Therefore, this project has been a great help in advancing my knowledge in autonomous robotics.

# 上海交通大学 毕业设计（学士学位论文）
# 单独工作报告

**SHANGHAI JIAO TONG UNIVERSITY**

**CAPSTONE DESIGN (BACHELOR’S THESIS)**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name：Xun Tu

Student ID Number：517370910149

Major: Electrical and Computer Engineering

## II.8 Xun Tu's Individual Contribution Report

The general tasks of our project are divided into two parts: the perception part and the motion control part, based on the individual' s academic background. I am assigned to work in the motion control part, which is responsible for controlling the motion of the robotic arm and the base, as well as designing the algorithm to generate the collision-free path for the excavator system. Though it is already clear that I need to work in control motion part, the concrete jobs to do in practice still vary with respect to the progress of the whole project. Generally, my tasks and contributions can be divided into four stages.

The first stage is to get familiar with the excavator and try to succeed in moving the boom, stick and base. When we had received our model in the first place, we just saw quite a few wires in chaos and several components which we have no idea how and whether they work. So, we had spent several days reading the provided manual from our sponsor, searching reference materials online and carrying out test experiments. At first, we were stuck with the mechanism to provide the power to the system. I proposed my understanding in the existing circuits and helped in redesigning the circuits according to our own needs. Later, after we had figured out the way to provide the power, we were still troubled by the appropriate input signal to control the components. In this part, I searched online and wrote initial programs to control the oil pump. Then, I proposed my model of how the hydraulic system works, which directly helps the partners write the programs to control the motion of the boom and the stick afterwards.

The second stage was to measure the orientation of the robotic arm and the base. I attended our group discussions on how to determine the orientation of the boom, the stick, and the arm, during which I had proposed using orientation sensors. My inspiration originates from my experience in manufacturing a baby stroller with automatic balancing seat, in which we had used an orientation sensor to measure the angle between the seat and the horizontal line. I hoped to replicate the model in our project. What' s more, I had noticed that the orientation sensor could not only measure the inclining angle, but also the yaw angle of our base. Therefore, I bring up my suggestions in the discussions and accepted approval. Then, I was in the part to test the sensor and install them on our excavator, during which I reviewed the knowledge in embedded system and Arduino programming.

Afterwards, our project entered the third stage, and so did my jobs. In the stage, we hoped to build up the programs to schedule the collision-free path. I offered to complete the jobs in forward and inverse kinematics, which is the part to transform between the user input coordinates and the joint space coordinates we hope to control. This required mathematical calculations in the geometric relationships and programming in python. During the process, I had enhanced my abilities in writing and expressing my ideas clearly and precisely, since I needed to explain to my teammates what the mathematical models were and why they made sense. I also learnt several path-searching algorithm, especially the one we were going to use, which was the A* algorithm. What’ s more, I practiced my skills in programming with my cooperators and peer reviewing the codes. With our efforts, we had succeeded in building the whole control programs.

The last stage in our project and my job was to reduce the inaccuracies and finetuning the sensors. After constructing the software and hardware fundamentals, we hoped to test the whole excavator system. The first several experiments had exposed fatal issues in the sensors we had used, with which I was assigned to handle. The first issue was the sensor drift. For the sensor we had used to measure the yaw angle of the base, it suffered severely from the drift. Basically, its output value was neither stable nor accurate. When the base stops moving in practice, the output of the sensor was still changing, which might throw an error in the control program and made the base move again. And the inaccuracy of the sensor led to the failures to satisfy our engineering specifications. After checking the manual and doing test experiments, I thought this was due to the electromagnetic disturbance and proposed to use a protective shell to block out distractions.

However, this did not go smoothly. Even with the designed protective shell, the magnetic disturbances were severe. Then, my partner proposed to use the accelerometer integrated on the orientation sensor to measure the yaw angle. Here came new problems later. When we were rotating the robotic arm with manual force, the output value was correct. But when we put on the power and do experiments, the output value suffered from huge instabilities and was no longer valid. To solve the problems, I had to search reference manuals online and do test experiments online and finally determined that the origin was the coupling between the three axes. We assumed that they were independent, but actually they were not, and this caused problems that we could not explain. By installing another sensor horizontally, we had fixed the problem.

Another challenge in this stage was the value leap in the sensor’ s output. Generally, when approaching the designed range limit, the output of the sensor was quite unstable and would leap. This meant that it might jump from a value to another, and the interval is beyond the designed precision. This would cause our stick to oscillate around the equilibrium point. After discussion, I had proposed to use two sensors to measure the orientation angle of the stick and helped in fixing the problem in practice.

In conclusion, my role in this project was to fix problems in motion control part, which include fixing the hardware problems on the excavator, testing, installing and finetuning the sensors, and helping in developing the control programs. I have deepened my understanding in related fields, such as the embedded system, path-searching algorithm, and the hydraulic system hierarchy. I have also improved my skills in communicating and sharing my ideas. These are meaningful benefits for me.

# 上海交通大学 毕业设计（学士学位论文）
# 单独工作报告

**SHANGHAI JIAO TONG UNIVERSITY**

**CAPSTONE DESIGN (BACHELOR’S THESIS)**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name：Yiwei Zhang

Student ID Number：517370910161

Major: Electrical and Computer Engineering

## II.9 Yiwei Zhang's Individual Contribution Report

As stated in the former main body of the report, to fulfill the task of excavator arm autonomous avoidance, we are required to finish two parts of sub-part works, which are motion control and planning part along with perception part. From the general perspective, Pengyuan Huang and I are determined to take charge of the perception part, while Shiji Liu, Xun Tu and Jinjie Liu are determined to take charge of the motion control and planning part. To explain it more clearly, motion control part focuses on the job of getting the excavator' s base, boom and stick to move into specified location given by our calculation and algorithms respectively, alongside with simulating and planning out the obstacle avoidance path based on the point cloud representing the surrounding obstacles around the excavator provided by the perception part. Whereas for the perception part, with the Intel RealSense D435i depth camera, they should collect as much surrounding information as possible and find out the best way to represent the obstacles without including the excavator itself.

Shiji Liu is our group leader and also the communicator between our group and the sponsor/instructor team. He bears the heaviest workload and coordinates between every aspect of the working process. He is mainly in charge of the whole project and implements the majority of the modeling, planning and integration algorithms on the ROS system. Jinjie Liu and Xun Tu mainly takes responsibility of the motion control part and forward/inverse kinematics modeling of the excavator arm and sensor controls. Pengyuan Huang works on the perception' s point cloud fusion algorithm with me.

For me, I am mainly responsible of working on the perception part' s point cloud fusion algorithms and other subsidiarity works among all steps. Besides, I also work on the designing and implementing of our final testing ground for exposition. Originally, we are provided with the excavator model by our sponsor, and we are required to perceive the surrounding environments using binocular cameras. Me and Pengyuan discussed with Shiji Liu about how the perception and motion control part should cooperate together. We decided that the input fed into motion planning interface on the ROS platform should be an integral point cloud which covers the front 180-degree view with the excavator itself as the reference point. Since the camera could only cover a certain range of degrees at a time, we are advised by our sponsor that we should rotate the camera several times and then through some point cloud fusion algorithms to fuse each taken point cloud into a whole. Pengyuan advised using Iterative Closest Points (addressed as "ICP" in the following context) algorithm to merge the point clouds, and I started right away testing this algorithm.

After observing the excavator model, we first decided to give it a try using Mynt depth camera. It requires Ubuntu 18.04 system to support the Mynt' s SDKs. I found out that the Mynt depth camera couldn' t function stably on the Ubuntu and often encountered crash, and thus proposed to find out another way. Intel RealSense D435i is a great alternation plan.

According to where we fixed the camera on the excavator model, the excavator arm is too close to the camera thus it introduced a lot of noises in the ICP algorithms. Since the algorithm itself couldn' t distinguish between excavator arm and real obstacles, the outcome wasn' t too reliable. I proposed that we should come up with some new way of fusing point clouds. Pengyuan and Shiji thought of hardcoded coordinate transformation matrix, and I proposed segmentation based on color since the excavator arm possesses a unique color which could clearly separate out themselves from the environment, and also segmentation based on distance since generally the points within certain range would be our excavator points (Our obstacles would not be put around our excavator model too near, since obstacles within in the minimal detection range couldn' t be captured by the camera.) Then I worked with Pengyuan together on the color segmentation algorithm and distance segmentation algorithm.

For the point cloud fusion algorithm, the d coordinate transformation matrix worked reliably and produced a relatively great and stable outcome. Therefore, we adopted it as our main method of fusion in the end. Together with the segmentation, we could now have an integral surrounding environment point cloud with excavator arm removed.

After Shiji Liu integrated motion part and perception part and deployed it on the ROS system, we decided to design and construct a real testing playground to validate our obstacle avoidance system. I am mainly in charge of this part. In order to be as close as possible to the real industrial environment, I want to include obstacles that look alike those which could be seen in the real world. Luckily, we have a construction site inside our building and I collected some stones inside the site, placed some together to form a mountain and scattered the rest around the testing ground. And after a careful discussion with Jinjie, we decided to make some paper models of trucks, electrical box and lamps out of paper bags to mimic real world scene. We have tested several ways of layout. Since we aren' t supposed to change the obstacle layout but only the starting position of the excavator bucket during the exposition, we want it to be as diverse as possible so that audience can try as many positions as possible and fully present the audience our obstacle avoidance function.

During the final test phase, I was in charge of deploying the test. I played as the audience and tested our system through various angles and thus spotted a flaw in our motion planning code which would introduce math domain error when certain target position is chosen. Our autonomous system proved to have obstacle avoidance accuracy over 85% during the whole testing process, thus validating our prototype design.

Due to time constraints and equipment computation power limitations, I could have researched into better obstacle segmentation plan using modern computer vision algorithms such as CNN and more, which might produce better and more reliable outcomes in more complex scene and real industrial environments. And at last, I want to thank all my teammates for their hard work, teamwork spirits, brilliant ideas and consideration. I also want to thank Prof. Chengbin Ma for his guidance, Sponsor Zishi Li for his patience. We could never achieve this without their help.

# 上海交通大学 毕业设计（学士学位论文）
# 单独工作报告

**SHANGHAI JIAO TONG UNIVERSITY**

**CAPSTONE DESIGN (BACHELOR’S THESIS)**

**INDIVIDUAL CONTRIBUTION REPORT**

Student Name：Pengyuan Huang

Student ID Number：517370910025

Major: Electrical and Computer Engineering

## II.10 Pengyuan Huang's Individual Contribution Report

The work is mainly divided into two parts in our group, namely the perception part and the motion control path planning part. Yiwei Zhang and I are assigned to the perception half due to their stronger computer science background, and Shiji Liu,Xun Tu and Jiejie Liu are responsible for both basic motion controls of the provided excavator model and the path planning algorithm implementation and realization.

Shiji is our team leader and takes charge of the communication between the team, the faculty advisors and the sponsor. He is also responsible for constructing the underlying framework of ROS that handles the communication of perception, motion controls and path planning. Xun devotes his effort to the testing and adaptation of orientation sensors, as well as the code implementation of forward and inverse kinematics for our path planning part.

With relatively stronger background on computer vision, Yiwei and I are assigned to take care of the perception part. According to the customer requirement, we decided to use computer vision to perceive the surrounding information. Binocular camera is a good choice because it' s able to calculate depth based on the disparity of the two images captured by the two eyes. I choose to use Intel RealSense D435 binocular camera on our system.

The SDK of this camera is quite helpful so I write a C++ program based upon it to utilize the device and capture images and depth. Point Cloud Library is a third-party library for users like us to convert RGBD information to point cloud. The reason why we want to use point cloud is that such data structure is appropriate for path planning algorithm to process and find a relatively optimal collision-free path.

However, the camera is able to only capture part of the surrounding each time. To find a collision-free path for the excavator arm, a comprehensive and complete point cloud representation works better. We work together with Jinjie Liu to install the camera along with a servo on the excavator model. Thus, the camera can rotate 180 degree to capture a series of point clouds.

The next problem becomes how to fuse them together. One solution is to use iterative closest point (ICP) algorithm. But this algorithm might match points on the ground which won' t lead to any fusion. So I implement a planar segmentation algorithm to segment the ground out. Therefore, we can run the ICP algorithm on the obstacles. However, because of the close distance to the camera, the excavator arm is captured with plenty of noises. The fusion result on two adjacent point clouds of the excavator arm doesn' t look well. In other words, even though the ICP algorithm doesn' t require any additional input, its performance depends upon the property of the obstacles in the surroundings.

So it’ s not very reliable.

Our team leader Shiji Liu and I discuss the alternative way to do point cloud fusion. In addition, the sponsor from BuildX provided us with a binocular camera ZED 2 which has a IMU built in. The IMU is able to capture the transformation and rotation information of the ZED camera. So the SDK of that camera already has a program to fuse the point clouds during the camera motion. This inspires us to capture the motion information during the camera rotation. The rotation angle is provided by the servo and we measure the distance between the camera and the rotation axis and the distance between the mounted point and the center of the excavator model. At first, we aren’ t very confident with the precision of our measurement. Shiji Liu implement a coordinate transform algorithm which requires all the motion information mentioned above. Surprising, the motion information is so precise that the result of this algorithm looks good.

What we have to do next is how to detect obstacles. At Design Review 1, we think such problem can be converted as a 3D object detection algorithm which requires us to find dataset and train CNN networks on it. But there doesn’ t exist a open-source 3D dataset used in excavator scenario online. We had to make our own dataset which is to capture vision data and manually annotate it. This task is pretty time-consuming. So we change our mind and have a discussion with our sponsor and instructor. We realize that the surrounding objects except the excavator arm can be viewed as obstacles. Therefore, the problem now is how to detect and remove the excavator arm from the surroundings. Because the excavator arm itself has a unique color and shape, it’ s much easier for us to detect it. According to the position where we mounted the camera, we are able to detect the excavator arm based upon distance information as well.

The color information we obtain from the capturing program is in the RGB color space which is difficult to do color segmentation. Therefore, I first convert them to the HSV color space which is descriptions of hue, saturation, and luminance. For a particular color region, we have to determine the upper color threshold and the lower color threshold so that the point with a color within that range will be removed out. For the unique color of the excavator arm, the upper threshold we choose is [140, 255, 255] and the lower one is [100, 110, 46]. In addition, we also utilize the distance information to remove arm points. The distance threshold is set to be 0.54 in the xy-plane.

To evaluate our arm-detection algorithm, we define the accuracy to be the ratio of the number of points removed by the algorithm to the number of excavator arm point covered by our manually selected bounding box. Jinjie Liu, Xun Tu and Yiwei Zhang take lots of effort to make obstacle models like car, truck and lamp on their own. And we collect some stones from the nearby construction field. All these items are used to create multiple validation scenarios. For all these scenarios, our algorithm is proved to be reliable. The measured accuracy is above 85