A close-up of a logo

Description automatically generated

ENG5325: Robotics Team Design Project M (2023-24)

Technical Report

Team #8

ZhixinMai (2937965)

Chaoxu Ji (2940201)

Zirui Wei (2816739)

Hengming Zhang

Yifei Zhao

Xinyuan Wu

## 1. Abstract

## 2. Introduction

## 3. Methodology

### 3.1 Introduction

### 3.2 Motion Part

#### 3.2.1 Selection of Simulation Software

In the methodology section of our project, a rigorous evaluation was undertaken to select the most suitable simulation software, culminating in the decision to employ Webots over other contenders such as Gazebo and Choregraphe. This choice was substantiated by several critical factors:

Advanced Simulation Capabilities: Webots outperforms with its superior physics engine and sensor simulation accuracy, essential for realistic robotic behavior modeling. While Gazebo also offers commendable physical simulation, Webots excels in user experience and simulation efficiency.

Extensive Model and Environment Library: The comprehensive library of ready-to-use robot models and environments in Webots significantly expedited our design and testing processes. Despite Choregraphe's specialized advantages for NAO robot programming, it falls short in simulating multi-robot interactions and environments as effectively as Webots.

Ease of Integration and Scalability: The support for integration with various robotics frameworks like ROS in Webots facilitates potential project expansions. Moreover, its open architecture and robust API enable customization and further development, aligning with our project's flexibility requirements.

Community and Documentation Support: The active developer community and extensive documentation of Webots provided invaluable assistance during technical challenges. While Gazebo also has a supportive community, Webots offers more specialized support in the domain of robot simulation that better suits our project needs.

Therefore, despite each software's unique strengths and application scenarios, Webots emerged as the preferred choice for our project based on its overall performance in simulation accuracy, user experience, resource availability, and scalability potential.

#### 3.2.2 Introduction to Webots

Webots serves as a powerful simulation tool that provides a professional environment for modeling, programming, and simulating robots. It offers a vast library of robot models and environments, facilitating the rapid development and testing of robotic algorithms. The user-friendly interface and realistic physics engine enable accurate and efficient simulations, making it an ideal choice for educational and research purposes in robotics.

#### 3.2.3 Creation of motion in Webots

### 3.3 Communication Part

#### 3.3.1 Inter-Robot Communication

In addressing the crucial aspect of inter-robot communication within our project, we adopted an integration approach between Webots and the Robot Operating System (ROS). This strategic decision was informed by ROS's extensive capabilities in facilitating seamless message passing and service invocation between distributed nodes in a robotic system. Through this integration, we established a robust framework enabling our simulated robots to communicate and coordinate effectively, mirroring potential real-world multi-agent interactions. This section delineates the methodologies employed to harness the combined strengths of Webots and ROS in achieving sophisticated inter-robot communication mechanisms, essential for orchestrating collaborative behaviors and strategies among the robotic soccer team members.

#### 3.3.2 Overview of ROS

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. ROS provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. Its modularity and tools-driven approach offer a scalable and reusable solution for robot software development.

#### 3.3.3 Communication with ROS

### 3.4 Vision Part

#### 3.4.1 Object detection

#### 3.4.2 Depth Measurement

### 3.5 Strategy Part

### 3.6 Summary of This Chapter

**Message Board第三章留言板**

（这里大家可以说下自己的建议，比如觉得哪些地方不必描述，哪些地方需要着重描述，或者有哪些需要添加的内容）

## 4. Results and Analysis

## 5. Team Performance Analysis

## 6. Conclusions and Further Work

## 7. References

## 8. Appendices