MULTI-AGENT COLLABORATION WITH HUMANOID NAO ROBOTS

ECE 486 Final Proposal

by

Stephen Lamczyk (CpE)
Jacob Strother (EE)
Zephaniah Amonoo-Harrison (CpE)
Seth Cummings (EE)

PhD Student Advisor:
Megan Witherow

Faculty Advisor:
Dr. Khan M. Iftekharuddin

Old Dominion University
Department of Electrical and Computer Engineering
December 6, 2021

ODU Honor Pledge

"I [we] pledge to support the Honor System of Old Dominion University. I [we] will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I [we] am aware that as a member of the academic community it is my responsibility to turn in all suspected violation of the Honor Code. I [we] will report to a hearing if summoned."

ABSTRACT

The purpose of this project is to use humanoid NAO robots to simulate a problem setting where one robot, Robot A, will model two or more behavioral states, and another robot, Robot B, will communicate and perform behaviors to move Robot A toward a target state. This project has implications in improving human/robot relations and interactions. The Vision Lab has two NAO robots, which will both be used in this project, each equipped with two five-megapixel cameras, four omnidirectional microphones, temperature sensor, gyroscope, accelerometer, and multiple position, tactile, force-resistive, and ultrasonic sensors. The NAO robots are fully programmable using Python 2.7 and the NAOqi API; there is also a GUI application called Choregraphe that can be used to make block diagrams with programmable blocks. These two programming methods will be used in tandem to establish the robot to robot interaction. Controllers for the behaviors of each robot will be designed and implemented. Each controller will require a behavior module that manages behaviors and communications performed by the robot, a recognition module to classify the behaviors of the other robot, and a reinforcement module to adjust behaviors based on feedback from the other robot. The overall performance goal for the recognition module should be close to 80% accuracy, with timing results of 30 seconds for robot interactions that consist of behaviors that change eye color and speech. Preliminary testing results have shown that our project plan is realistic and achievable. The expected outcomes of this project are to successfully implement the robot to robot interaction as described above.

Table of Contents

Page

LIST OF FIGURES . . . v

LIST OF TABLES . . . vi

Chapter

1. 1. INTRODUCTION . . . 1
   1.1 DESIGN PROBLEM . . . 1
   1.2 THE FUNCTION(S) OF THE DESIGN . . . 1
   1.3 QUANTITATIVE PERFORMANCE OBJECTIVES . . . 2

2. 2. DESIGN APPROACH . . . 2
   2.1 REALISTIC CONSTRAINTS . . . 2
   2.2 ALTERNATIVE DESIGNS . . . 4
   2.3 ANALYSES USED TO SELECT AMONG THESE ALTERNATIVE DESIGN CONCEPTS . . . 5
   2.4 TEAM ORGANIZATION AND PERFORMANCE . . . 6

3. 3. PROJECT DELIVERABLES . . . 6
   3.1 TASKS AND RESPONSIBILITIES . . . 6
   3.2 TEAM ORGANIZATION . . . 7
   3.3 MILESTONE TIMELINE . . . 7

4. 4. DESIGN SPECIFICATIONS . . . 8
   4.1 DETAILS OF THE ENGINEERING DESIGN . . . 8
   4.2 PARTS LIST / SOFTWARE COMPONENTS . . . 11
   4.3 ENGINEERING STANDARDS . . . 11

5. 5. PRELIMINARY DESIGN PERFORMANCE . . . 12
   5.1 BEHAVIOR MODULES . . . 12
   5.2 RECOGNITION MODULES . . . 13

6. 6. PROJECT BROADER IMPACTS . . . 15
   6.1 ETHICAL IMPLICATIONS OR ISSUES OF THE PROJECT . . . 15
   6.2 CONSIDERATION PUBLIC HEALTH, SAFETY, AND WELFARE . . . 16
   6.3 GLOBAL, CULTURAL, AND SOCIAL, FACTORS IN THE ENGINEERING DESIGN PROCESS . . . 16
   6.4 ENVIRONMENTAL FACTORS IN THE ENGINEERING DESIGN PROCESS . . . 17
   6.5 ECONOMIC FACTORS IN THE ENGINEERING DESIGN PROCESS . . . 17

6.6 LIFELONG LEARNING 17
6.7 KNOWLEDGE OF CONTEMPORARY ISSUES 18

REFERENCES 21

APPENDICES
A. EXECUTIVE SUMMARY 22
B. APPENDIX 24

List of Figures

Figure Page

1 Robot B Reinforcement Module Flow Chart. . . . 10

2 Robot A Reinforcement Module Flow Chart. . . . 11

3 Choregraphe Block Diagram Used to Change the Eye Color of the NAO Robot to Blue 13

4 Real-time Image of NAO Robot with Blue Eye Color Which Symbolizes a Neutral Mood 13

5 Choregraphe Block Diagram Used to Recognize Speech 14

6 Choregraphe Block Diagram Used to Recognize Surrounding Objects with Vision 15

7 Example of Robot A and Robot B Interacting With One Another 23

8 Gantt Chart Representing our Milestones and Goals 26

List of Tables

Table Page

1 Robot A Moods 8

2 Robot B Moods 8

3 Past Meeting Schedule 24

4 NAO 6 Robot Specifications 25

1. Introduction

1.1 DESIGN PROBLEM

The objective of this project is to investigate robot-to-robot interaction by designing and implementing a problem setting where two NAO robots communicate and perform behaviors to reach a target state. One potential use case for this design is in autism intervention, where Robot A models a child receiving intervention and Robot B models a robot providing scaffolding to the child.

The design approach will involve using Choregraphe and the NAOqi API to build each module's controller. There will be three modules: behavior, reinforcement, and recognition. Choregraphe is a GUI-based programming application that can be used to create programmable block diagrams to be executed by the NAO robot. Each programmable block contains code from the Python programming language that can be modified where necessary. NAOqi is a Python library that contains useful functions and blocks that can be used by the robot. The NAO robot is a humanoid programmable robot that has cameras, LEDs, speakers, motors, sonar, speech recognition, and more; many of the NAO robot's sensing technologies will also be employed to accomplish the design task.

1.2 THE FUNCTION(S) OF THE DESIGN

This design will use two NAO robots (Robot A (Child Robot) and Robot B (Therapist Robot)), each connected to a separate computer. The main components on the NAO robot that will be used are the cameras, speakers, motors and voice recognition. The cameras will be used to monitor what the NAO robot can see and to collect training data for object recognition. The images taken from the camera on the NAO can be uploaded to a built-in vision recognition database in Choregraphe. The speakers will be used to say certain phrases associated with the current behavior, and also for debugging and testing purposes. The NAO's motors will also be manipulated when a different pose is required by design. There is also a voice recognition block in Choregraphe that will be used to recognize another NAO robot's voice. Choregraphe will be used to build block diagrams with Python code to be executed on the NAO robot. The programs built in Choregraphe will be uploaded to each robot through an Ethernet connection between the computer and robot. Each robot will have their own block diagram program designed in Choregraphe tailored to what the robot will be doing; robot A will display a random initial behavior and robot B will try to recognize the mood/state of Robot A and move robot A toward a target state. Both robots will adjust their behaviors based on what the other robot does. In short, the behavioral module will directly manipulate the robot behavior, the recognition module will recognize the behavior of the other robot, and the reinforcement module will provide a state machine for the robots to change states based upon feedback from the recognition module. The moods for the behavioral module are discussed in tables 1 and 2. The behavioral, recognition, and reinforcement modules will be created using Choregraphe and the NAOqi API.

1.3 QUANTITATIVE PERFORMANCE OBJECTIVES

Robot A will be moved to the target state in a realistic amount of time based on how complex the set of behaviors are. Generally, human reaction time to a stimulus is around 250 ms for visual reaction time, and 225 ms for audio reaction time [1]. Knowing this, an interaction with three behaviors that involve changing eye color and speech should take no longer than 30 seconds, giving ample amount of time for the robots to run their programs and react to each other. The actual recognition module should have an overall accuracy of about 80% just based on qualitative observations from testing in Choregraphe, although, this is subject to change upon running more tests. We felt that in our testing that 80% recognition was obtainable given our observed accuracy of the recognition module interacting with our behavioral modules.

2. Design Approach

2.1 REALISTIC CONSTRAINTS

2.1.1 PUBLIC HEALTH AND SAFETY

The main constraint regarding the health and safety of the general public is the manner in which the robots will be used around people. This means that the safety of those around the robots must always be prioritized. The design of the project has in mind the potential use case of one robot acting as a caretaker for the child robot. If implemented in a scenario with a human child, the NAO robot must pose no threat towards the subject or person involved. On a larger scale, the NAO robots can be used in settings of education, healthcare, and even tourism. Consequently, we are limited in our implementation of different behaviors. The different behaviors implemented must not be erratic and pose a risk to those around it. One example of a potentially risky behavior could be moving the arms of the NAO. This may hurt a child surrounding the robot if the child is very close. Therefore, we will be rather limited in the expression of the robot's behavior.

2.1.2 ECONOMIC/COST ANALYSIS

The project makes use of two NAO V6 robots which were both purchased from RobotLAB. RobotLAB serves as Softbank's exclusive North American distributor as well as an educational robotics market. The robots retail for $10,500.00 each before the application of an educational discount [2]. All the components used in this project (including the robots and their software) have been purchased and provided prior to the beginning of the project. Due to this, no additional budget has been allocated for the project.

2.1.3 ENVIRONMENTAL

The programming of the NAO robots took place in the Vision Lab. The robots can be ran wirelessly, however, if Ethernet wires are applied then cable management should be taken into account. The wires used to establish a connection between the robots are of a certain length. This means that there is a restriction in the amount of movement the robots can perform. To accommodate this, the two robots are always used in close proximity.

The NAO robots make use of camera sensors in their eyes that can detect movement. In this project, these sensors are used to establish communication between the two robots. In a more complex design, these sensors can also be used be used by the robots to locate and navigate to one another. For these components to function properly, the environment the robots are placed in must be free of obstruction such that each NAO robot's camera and sensors can identify the other. Another constraint is the sturdiness of the robots. The robots happen to be fragile and must be operated on a flat surface. For this project, both robots were placed adjacently in the center of a large table.

2.1.4 SOCIAL AND CULTURAL

There are no social or cultural constraints associated with this project. However, the future implementation of these robots in society may present a few. One of the NAO robots has been given instructions to provide care for the other. This robot could also be made to stand in as a caretaker for a human child. This is where the issues arise. Based off demographics and cultural backgrounds, not every parent would be comfortable having a child interact with the NAO robots which is understandable.

2.1.5 MANUFACTURABILITY

NAO robots are still in production and the NAOqi library used in programming them is duly updated. All components used in this project have already been purchased. With that being stated, the completion of this project will not rely on any manufacturability constraints.

2.1.6 SUSTAINABILITY

One huge constraint for this project is that the NAO robots and NAOqi do not support the latest version of Python. Python 2.7 is the only version of Python supported [3] . Python 2.7 went out of support on January 1, 2020. Therefore, many libraries do not support it. This limits what we can do with this project without introducing extra complexity to support Python 3 code. In addition, the NAO robots are lacking GPUs in their hardware and only have an Intel Atom CPU. This additionally limits the complexity and size of any potential neural networks that could be used for the recognition module unless other methods are implemented such as streaming data to a nearby GPU.

2.1.7 TIME CONSTRAINTS AND COST ANALYSIS

As stated previously in the economic section, these robots were purchased beforehand and were not purchase solely for this project. Given that these robots combined would cost around $20,000, it would most likely not be justifiable to purchase these solely for this project. It may be justifiable to purchase these with the knowledge that they would be used in future senior design projects.

In addition, there are some natural time constraints that occur when working with the robot. Firstly, the robot motors tend to overheat after an hour and the robot battery only lasts about an hour. This means we can only work in one hour sprints.

2.2 ALTERNATIVE DESIGNS

One initial decision is the choice between virtually or directly programming the robots. Virtual robots would be programmed through the WeBots software while the real robots could be programmed through either Python 2.7, Choregraphe, or C++. It is important to note that code written for the virtual robots will not work for real robots due to incompatibilities in the API. This choice of languages for the real robots also represents another set of alternative designs. Different languages or software have different strengths and weaknesses that are discussed in the next section.

Within the two different robots controllers, there are competing designs for the modules. For example, there are a large selection of capabilities available to express robot behavior. These include motion, pose, eye color, voice, etc. This ties into the structure of the recognition module as this module must also classify the other robot's behavior. The structure of the recognition module is also another thing to consider. It could utilize the simple built-in NAO capabilities or we could develop our own method to perform these classification tasks. In addition, there are many possible structures for a reinforcement module. The reinforcement module could be made using concepts from reinforcement learning or just resemble a simple state machine. Further details are discussed in the next section about choosing between a state machine and reinforcement learning.

2.3 ANALYSES USED TO SELECT AMONG THESE ALTERNATIVE DESIGN CONCEPTS

For the virtual vs real robot programming decision, we decided to program the real robot. This is because a virtual robot would not have much real world use and would be practically abandoned at the conclusion of this project.

Following the choice to use the real robots, we decided to use Choregraphe to program the robots. This meant we could already utilize built-in modules for programming the robots. We could then fallback to Python if we had to write a custom module. C++ was avoided due to the difficulty of the language and the lack of functionality included in Choregraphe.

For the behavior module, we decided to utilize voice and eye color to express behavior. This is because we could utilize the built-in object recognition capabilities of the NAO robots. If we were to use motion to communicate behavior, then a custom neural network would be required. This network would have to be built to handle time-series data which would be very complex and outside the scope of this project. Implementation of such a network is also limited by the robot only supporting Python 2.7. This also explains our reasoning for the recognition module. We decided to solely use the object recognition and the voice recognition modules built into the NAO robot to classify the behavior of the other robot. Our experiments so far prove that these built-in modules are capable of performing the required recognition tasks.

For the reinforcement module, we decided that a simple state machine most effectively handles the communication between a robot's recognition and reinforcement modules without adding excess complexity. While a reinforcement learning approach could be used, this greatly increases the complexity of the modeling and the group members do not have an existing background or understanding of reinforcement learning.

2.4 TEAM ORGANIZATION AND PERFORMANCE

When splitting up the work of the 6 modules, we naturally considered each team member's own skills and interests. Stephen and Jacob chose the recognition modules due to their experience with machine learning. Zeph and Seth chose the behavior modules due to their experience with programming. Stephen and Zeph will work on the reinforcement module for robot B while Seth and Jacob will work on the reinforcement module for robot A. More details are described in the following section.

3. Project Deliverables

3.1 TASKS AND RESPONSIBILITIES

Stephen's responsibilities were researching the feasibility of NAOs recognizing each other's voices. He also developed a detailed potential plan for the fusion of various mobile nets for recognition that never came to fruition due to the decision to switch to the NAO robot's built in object recognition capabilities. Stephen was also coordinating with Seth and Jacob on the possibility and capability of NAO robots classifying another's eye color. Also, Stephen is in charge of building the recognition module for Robot B. Stephen also set up the GitHub page for this project and has had a supervisory role in writing the report and in general.

Zephaniah's responsibilities were looking into NAO robot behavior to determine if the NAO Robots were able to overlap emotions and movement at the same time. Zephaniah was also in charge of building the behavior module for Robot B (Therapist Robot).

Seth's responsibilities were to do research on all the sensors on the NAO Robot and to see how well they responded. Seth was also in charge of building the behavior module for Robot A. Seth coordinated with Stephen and Jacob on building the behavior module for Robot A and made sure Robot B learned all of Robots A behaviors using Choreographe's learn mode which uses Robot's B camera to take pictures of Robot's A eyes to learn its moods.

Jacob's responsibilities were to organize the shared drive and documents, troubleshoot issues with installing Choregraphe and NAOqi, help Stephen test the NAO robots' capabilities, design Choregraphe block diagrams, draft plans and produce figures, and develop a recognition module for Robot A. Jacob also collaborated with Stephen on conducting feasibility tests on the functionality of the NAO voice recognition and object recognition systems.

Overall, all team members worked together closely on all tasks and responsibilities. We worked together over Zoom meetings to complete all assigned tasks on blackboard on time and in-person meetings to test the NAO Robots functions. We have also all coordinated and collaborated in writing the proposal with no one person having a larger or smaller share of work than the others. After everyone has completed their assigned module, we will work in groups of two on both reinforcement modules (for Robot A and B). Seth and Jacob will work on Robot A's reinforcement module while Stephen and Zephaniah will work on Robot B's reinforcement module.

3.2 TEAM ORGANIZATION

Together as a team we met every Wednesday at 2PM via Zoom along with Dr. Khan's PhD student, Megan. Dr. Khan assisted us when needed. Also, as a team, we met in person one to two times a week to work on the NAO Robots. We kept a shared Google Drive open between all group members to complete assignments in a timely manner. Furthermore, the team has a group text chat to be able to get into contact with one another when needed. Table 3 can be found in the appendix section documenting our history of meetings. Unless explicitly stated, it can be assumed that we also worked on that week's assignment during the weekly meeting on Wednesdays. Every meeting lasted approximately an hour.

3.3 MILESTONE TIMELINE

Our full size Gantt chart of all tasks we completed/will complete can be found in the appendix, figure 8. The start date is when we started or intend to start on a specific task and the end date is when we plan to finish the following task. It is also split up into the different milestones: design, behavioral module, recognition module, and reinforcement module.

4. Design Specifications

4.1 DETAILS OF THE ENGINEERING DESIGN

There are two robots. Robot A will start at a random state and robot B will perform behaviors to move Robot A to a target state. Each robot has three modules: behavior, recognition, and reinforcement.

4.1.1 BEHAVIOR MODULE

There will be three behaviors per robot that will each correspond to the current mood of robot A. These behaviors include three levels of responsiveness for robot A and three levels of intervention that will be provided by robot B. The behavior for robot A will be implemented in the form of a mood variable ranging from -100 to 100 with -100 being negative mood, 0 being neutral mood, and 100 being positive mood. Thresholds will be established at the -50 mark between negative and neutral mood as well as the 50 mark between neutral and positive mood. Robot A's behavior will be exhibited in several different ways according to table 1. These are implemented in Choregraphe.

TABLE 1: Robot A Moods

Mood	**Eye Color**	**Voice**
[-100, -50) (negative)	red (0xFF0000)	Unresponsive
[-50, 50) (neutral)	blue (0x0000FF)	Responds to prompts normally
[50, 100] (positive)	green (0x00FF00)	Responds to prompts positively

As stated before, the behavior for robot B will be based upon the behavior and mood of robot A. This is as follows in table 2. The thresholds between moods are the same as before.

TABLE 2: Robot B Moods

Recognized Robot A Mood	**Eye Color (Hexcode)**	**Voice**
-100 (negative)	green (0x00FF00)	Tries to console robot A
0 (neutral)	blue (0x0000FF)	Tries to cheer up robot A
100 (positive)	red (0xFF0000)	Makes small talk with robot A

4.1.2 RECOGNITION

The recognition module for each robot will be implemented by using the voice recognition functionality in the Choregraphe as well as the object recognition module. The voice recognition module in NAOqi has the ability to recognize predetermined words. The object recognition module has a similar ability to recognize predetermined classes of objects. Each robot will be trained to recognize the other's eye color as well as speech. For robot A, the robot will monitor robot B for the appropriate speech and respond if it is correct. Additionally, robot B will also monitor robot A's eye color to provide the intervening behavior. If both conditions are met, then the robot A's mood will improve by +10.

Similarly, robot B will monitor robot A for the appropriate mood. It will monitor robot A's responses to it's statements as well and eye color through the use of the voice recognition and object recognition modules respectively. It will then return the expected mood of robot A, which determines the appropriate behavior for robot B.

4.1.3 REINFORCEMENT MODULE

The reinforcement module for each robot will be implemented as a state machine and is the main function that calls other functions for each robot. Figure 1 and 2 are flow charts for how each robots' reinforcement should be implemented. These will be converted into Choregraphe block diagrams and/or Python programs to be executed on the NAO robots.

The flow chart diagram in figure 1 shows the flow of execution for robot B. The robot will be initialized with some random state and then will attempt to locate robot A. Once robot A is located, its behavior will be recognized and the recognition module will check if it is the target state. The target state and target behavior can be interpreted as synonyms.

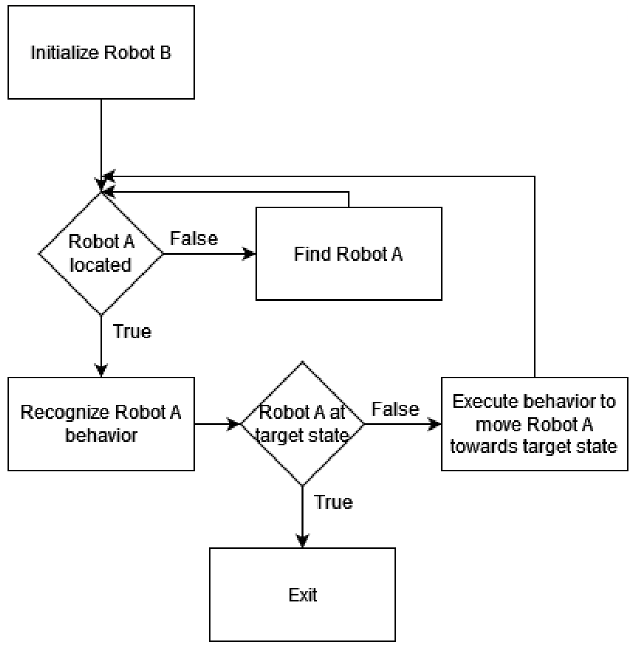

Fig. 1: Robot B Reinforcement Module Flow Chart.

The flow chart diagram in figure 2 shows the flow of execution for robot A. Similar to the previous diagram described, robot A will be initialized with some random state, and then robot B will need to be located and its behavior will need to be recognized. If robot A prefers the behavior of robot B, then it will move closer to the target state, else it will move away from the target state.

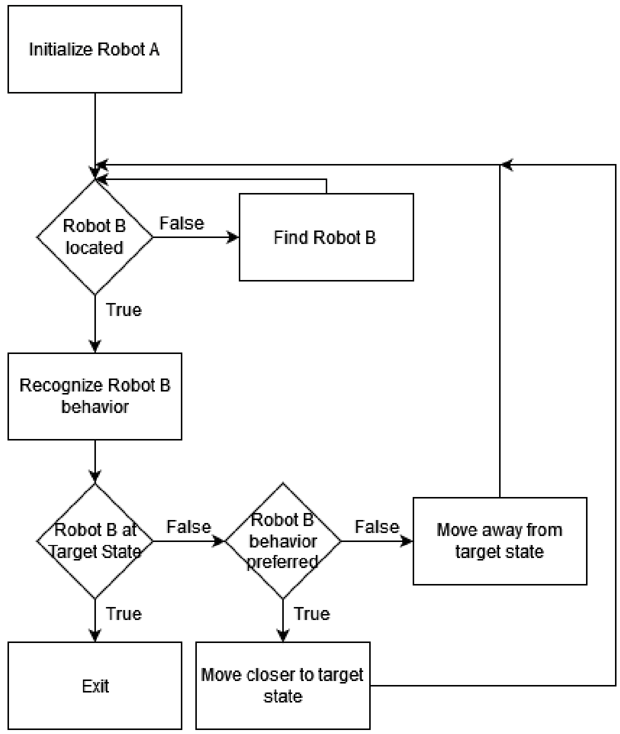

Fig. 2: Robot A Reinforcement Module Flow Chart.

4.2 PARTS LIST / SOFTWARE COMPONENTS

Required components for this project are rather limited.

- 2 NAO Robots
- 2 Laptops with Choregraphe installed
- 2 Ethernet Cords

Specifications for the NAO robot used are located in table 4 in the appendix. Table taken in part from [4].

4.3 ENGINEERING STANDARDS

4.3.1 IEEE 7007-2021

An IEEE standard that applies to this project is the Standard for Ethically Driven Robotics and Automation Systems (IEEE 7007-2021). With having the capability to program the NAO Robots to do almost any task, we must make sure to ethically guide the

NAO Robots in the correct direction while also ensuring that no harm is done to humans surrounding the robot. Throughout the project the NAO Robots must complete many tasks such as changing emotions, communicating with one another, assisting one another and doing specific body emotions such as moving the robot's arms and legs. For this to happen we must make sure the recognition module and reinforcement module of the robot is designed flawlessly; if the design of these modules is not done correctly, the risk of the NAO Robot doing something unethical increases significantly. Two subdomains that we chose for this IEEE standard is, Transparency and Accountability (TA) which formalizes the concepts and relationships necessary to enable ethical autonomous systems and Norms and Ethical Principles (NEP) which formalizes aspects of ethical theories and principles. This is how the Standard for Ethically Driven Robotics and Automation Systems applies to this project.

4.3.2 PYTHON 2.7.18

As this project requires the use of Python 2.7, either by using Python itself or by using Choregraphe which relies on Python 2.7, Python 2.7 is an important and relevant engineering standard even if it is not an IEEE standard. Python 2.7.18 is the latest version of Python 2.7 that will be used on this project. Even though Python 2.7 is antiquated and out of support, it must be used for this project as the newer (and better) versions of Python such as 3.10 are not supported on the NAO robot. This also ties into project constraints as most libraries support Python 3 with limited or no support for Python 2. If the standard version of Python 2.7.18 was not applied to this project, this would make using libraries very difficult as they rely on the standard Python version. Additionally, porting the written code to another platform would also be more difficult and it would make our results harder to reproduce. [3]

5. Preliminary Design Performance

We are making steady progress towards developing our design and have fulfilled our first milestones. We have made substantial progress implementing the behavior and recognition module for each robot. Details are described in the sections below.

5.1 BEHAVIOR MODULES

We have been able to successfully implement the robot behaviors outlined in tables 1 and 2. These so far have been: successful modulation of eye color to express mood and successful manipulation of the voice of the NAO robots. We have also been able to change the tone of voice to make it easier for the other robot to recognize the voice. Figure 3 shows a block diagram in Choregraphe that changes the eye color of the NAO robot; there is a block that controls the specific color, and a block that receives the color value and applies it to the LEDs. The results from running this program are shown in figure 4.

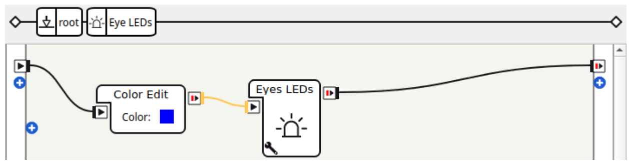

Fig. 3: Choregraphe Block Diagram Used to Change the Eye Color of the NAO Robot to Blue

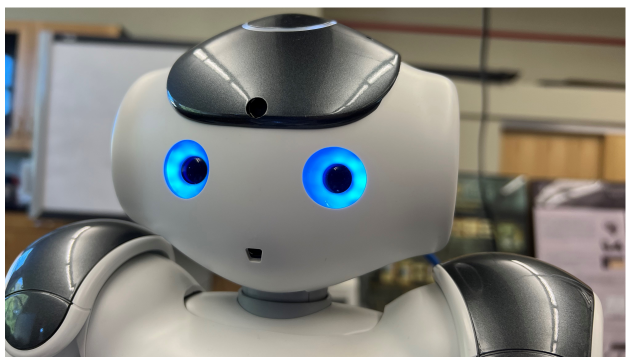

Fig. 4: Real-time Image of NAO Robot with Blue Eye Color Which Symbolizes a Neutral Mood

5.2 RECOGNITION MODULES

Though the behavior module functions properly, the success of the project stems from if the NAO robots can recognize each other. For this, the recognition module focuses on the specific actions the robots will look for and how they will respond to each them. Using the NAO robots' built-in object recognition functionality, we have been able to develop a simple classifier to identify the three different eye colors of the NAO robots. The NAO robots have also been able to detect when certain words are spoken from their counterparts.

5.2.1 SPEECH RECOGNITION

Figure 5 shows a Choregraphe block diagram that is used to recognize speech. In the diagram, a delay block is implemented. This gives the robot a brief moment to process the data it has received. The speech recognition block detects a keyword and if that keyword is heard, then the NAO robot will say that word and go to a delay block. If the keyword is not recognized, the robot will not respond and will transition to the delay block. The delay block is needed to prevent the NAO robot from being overloaded with commands. Following this design, the robots have very easily been able to recognize the speech from the other robot.

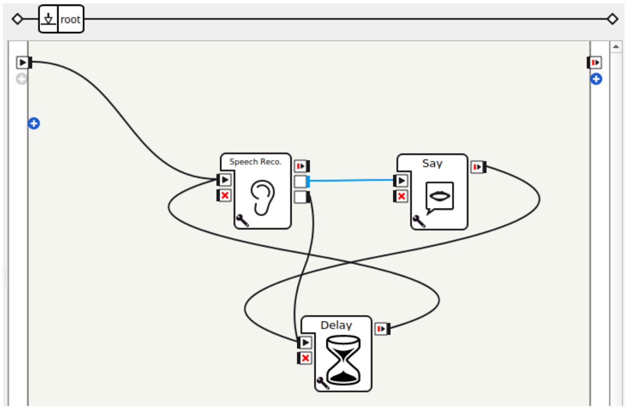

Fig. 5: Choregraphe Block Diagram Used to Recognize Speech

5.2.2 VISION RECOGNITION

Figure 6 shows a Choregraphe block diagram that uses vision recognition to recognize an object. The objects to recognize in this project are the two NAO robots; more specifically, the eye color of the robot. First, a vision recognition database is created in Choregraphe by taking pictures of objects with the NAO robot's cameras and then assigning labels to these images. The vision recognition block will then return the label of each object that it recognizes. Additionally, the NAO robot will also say the object labels. Congruent with the speech recognition diagram in figure 5, there must be a delay block added to the design. Experiments with the vision recognition system have also proved that it works well with minimal wrong classifications.

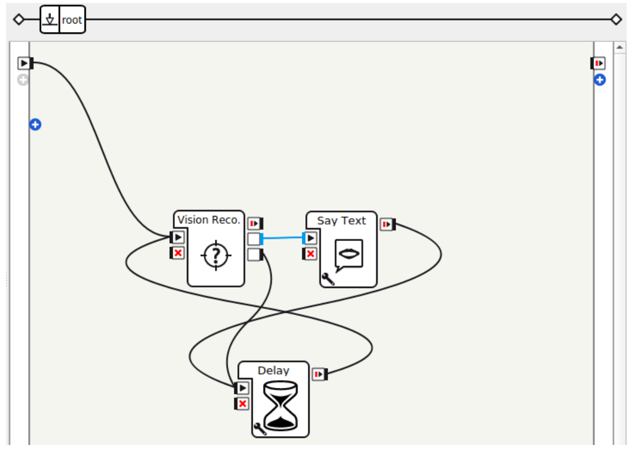

Fig. 6: Choregraphe Block Diagram Used to Recognize Surrounding Objects with Vision

6. Project Broader Impacts

6.1 ETHICAL IMPLICATIONS OR ISSUES OF THE PROJECT

The robots used in this project are very versatile in their ability to perform various tasks. The robots have a face recognition system that allows them to follow the movements of a person they have detected. For this project, the NAO robots are made to recognize and perform actions in response to one another. If this recognition system does not prove to be consistent, it could take incorrect action against the wrong individual. In other instances, NAO robots are utilized in the fields of healthcare and education where they are specifically programmed to interact with people. This is where the consistency in recognition becomes a requirement. If this feature is not accurate, the NAO robots can pose a threat to the safety of those around them by behaving unpredictably.

To prevent the implications of risk and danger to the participants of this project, the IEEE Code of Ethics was thoroughly observed and practiced. Specifically, item 1 of the IEEE Code of Ethics is followed to prevent harm to the health of the public. It is understood that failure to comply with this code could cause harm to students, halt the completion of the project, and ultimately stain the image of the institution.

Overall, the ten different items in the IEEE code of ethics are followed at all times. In addition to the first one as described above, other codes such as the sixth item are being followed by us naturally improving our technical competence in this field of computer vision and machine learning,

6.2 CONSIDERATION PUBLIC HEALTH, SAFETY, AND WELFARE

This project requires the use of multiple lithium battery power supplies. With this we must make sure to properly store and maintain the power supplies while working on the NAO Robots. If a power supply becomes damaged, it can explode and burst into flames. To prevent this we will use the standard charger provided to us which outputs 25.2 Vdc / 2A. Also, the NAO Robots are designed to function between 50-95 degrees Fahrenheit. Therefore, the batteries should not be exposed to temperatures above 113 degrees Fahrenheit and should only function between 10-90 percent relative humidity. Consequently, we will only use the robots indoors in climate controlled rooms.

This project also involves many motors and joints which are located in the NAO robot's head, arms, hands and legs. It is important that no loose clothing, body parts, and jewelry get stuck in these motors or joints. To prevent this, it is important not to touch or stand close to the NAO robot while it is moving. Doing so could cause the gears in the motors to grab onto a object which could damage both the object and the NAO robot. To summarize, we will make sure to operate the robot appropriately with precautions being undertaken at all times to prevent damage to the robot and people.

6.3 GLOBAL, CULTURAL, AND SOCIAL, FACTORS IN THE ENGINEERING DESIGN PROCESS

6.3.1 GLOBAL

While the NAO robot does have the capability to understand many different languages, and this is one possible global impact. We are not currently considering other languages in the programming of the NAO robots. Consequently, these robots will only work and perform in English. However, these robots can then have their internal English text translated to another language if they are operated in another country where a different language is used.

6.3.2 CULTURAL

While programming the NAO robots, they are made to perform tasks that are inherently seen through many of the group's own cultural biases. For example, we program robot B to console robot A based on what we deem as appropriate comforting behavior. However, What qualifies as comforting behavior varies based on culture. An individual from Asia may have a different sense of consolation compared to a native of the United States. This is an important factor to consider if the developed program is ever ported to a foreign country.

6.3.3 SOCIAL

The face recognition feature of the NAO robots allows them to detect faces and follow them. This feature does not consistently occur but rather activates randomly when a person passes the sensors of the robots. This means that the robots could arbitrarily pick up on the actions of a person, which in some cases, people may deem as an invasion of privacy. Considering this, the robots were placed directly in front of each other such that the only thing in their views were themselves. Each robot was also programmed to identify specific actions performed only by the opposing robot.

6.4 ENVIRONMENTAL FACTORS IN THE ENGINEERING DESIGN PROCESS

Both NAO Robots carry a lithium battery power supply. They are both properly stored in ODU's Vision Lab to prevent damage. If the lithium battery power supplies do become damaged, we will follow the correct protocol to dispose of them, dropping them off at a battery recycling drop off point. Also, if any part of the NAO robot becomes damaged, we will also dispose of it accordingly.

6.5 ECONOMIC FACTORS IN THE ENGINEERING DESIGN PROCESS

The price of the NAO V6 robot is $10,500.00 right now [2]; thus, the initial investment for someone purchasing the robot is relatively high and not realistic for most people. Luckily, the Vision Lab has two NAO robots, cables, and laptops with the necessary software installed. The software is also free.

6.6 LIFELONG LEARNING

Jacob - From my experience of working with many different Python libraries prior to this project, I am able to apply that knowledge by learning an entirely new library and software in NAOqi and Choregraphe. Taking CS 150, 250, and 361 has helped with hands-on programming, and taking ECE 241 and 341 has helped with creating flow charts that are similar to finite-state machines. I hope to continue learning new techniques and software to aid in my future career/studies.

Zeph - Upon choosing this project, I had two goals that I wished to accomplish. The first being to understand how the programming of robots could assist the lives of people. Secondly, I wanted to learn how to be more proficient in working in groups. Prior to being an engineer, I never thought I would be working with robots, however, being a part of this project is an experience that will assist in any future endeavour in robotics. I have never worked on a project of this scale before but it has been a great experience for me. It has taught me that with good collaboration, the hardest of assignments become simplified. Working on this project will definitely assist me in my career and the job market.

Seth - With taking many Engineering classes at ODU, it has made me a better critical thinker and problem solver when it comes to new material such as working with the NAO robots. I plan on using what I learn from this project to further my career with robots, artificial intelligence and autonomous vehicles as they are some of my favorite things to work on. It will also benefit me as I have to communicate with my group daily and use teamwork to complete tasks. This is a very common in the engineering field. You are almost always in some type of small group working together to achieve a common goal.

Stephen- In my major, I have been focusing on machine learning and computer vision. This project will help broaden my knowledge of what is possible in these domains. I have also been able to use lots of material from my courses in this project. My experience in ECE 341 and ECE 241 with state machines has helped the team and I develop an effective plan for the reinforcement modules. In addition, my knowledge of computer vision from ECE 545 and machine learning from ECE 550 and 607 helps me understand exactly what is happening with the object recognition and voice recognition modules.

6.7 KNOWLEDGE OF CONTEMPORARY ISSUES

One of the biggest contemporary issues this project faces is the ethics behind the robots and the artificial intelligence behind them. AI that can learn complex tasks on its own with minimal training data is critical. With this, we have to make sure AI does not erode human freedom by replacing them in job fields [5]. Robots and AI grow increasingly every year as technology advances and trying to use them ethically is a challenge engineers face daily.

6.7.1 IMPACT OF ENGINEERING SOLUTIONS

Global

One very important tool the NAO robot has is its language barrier. The NAO robot can recognize up to 20 languages and also communicate with dialogue with the correct programming [6]. This is an important feature and with this feature it can connect with almost anyone around the world. NAO robots are already being used to help children learn their primary language with a second language [7].

Societal

The use of NAO robots is popular in aiding children with Autism either in the classroom or at home. This will be made more popular through the use of mobile apps being developed to control the NAO robots [8]. Having an easily accessible software application for non-technical professionals, such as therapists and teachers, to use can increase the level of comfort in using the NAO robots, thus increasing human and robot relations and helping children with disabilities.

Environmental

The NAO robots' use of a lithium-ion battery greatly reduces the amount of waste produced. Lithium is not expected to bio-accumulate and its human and environmental toxicity are low [9]. To add to that, lithium in certain doses can stimulate certain plants [9]. Compared to other types of batteries, lithium is considered less toxic to the environment.

Economic

NAO robots can be used in therapy and social work jobs. This can help aid those who work these jobs, and may even open up more collaboration with the STEM fields and social sciences, creating more jobs and opportunities.

6.7.2 EXPECTED OVERALL EDUCATIONAL BENEFITS FROM THIS PROJECT

The overall study of the NAO robots greatly assists in the mastering of the Python programming language. While Python 2.7 is used and the most up to date version of Python is Python 3.10, the skills learned should still be somewhat transferable to newer versions of Python. Python in general is a very powerful scripting language due to its simple and readable syntax as well as the extremely large assortment of libraries available ranging from machine learning to Arduino programming. The ability to write code in this new language will help the students working with this project in their future work.

In addition to learning the Python programming language, learning how to program the NAO robots will be beneficial in the future if any group members decide to go into robotics in the future. This added experience will surely make learning how to program any other robot easier. This could be similar to learning how to program in another language given that the person already knows one.

Exposure to the built-in machine learning capabilities of the NAO robot via the built-in vision and speech recognition modules is another expected educational benefit. This will help the group members realize the practical applications of machine learning in the real world.

This project will also grant us the opportunity to increase our knowledge of various subdomains and see how they can also be used in the real world. Some examples of this include designing the state machines in the two different NAO robots. Our knowledge of programming will also be tested when programming the robots while also keeping their sensors in mind.

References

[1] Aditya Jain, Ramta Bansal, Avnish Kumar, et al. *A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students*. 2015. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4456887/.

[2] Cedric Vaudel. *NAO Power V6 Educator Pack*. URL: https://www.robotlab.com/store/nao-power-v6-educator-pack.

[3] *Python*. URL: https://www.python.org/.

[4] Cedric Vaudel. *Nao Power V6 Standard Edition*. URL: https://www.robotlab.com/store/nao-power-v6-standard-edition.

[5] Steve Crowe. *10 Biggest Challenges in Robotics*. Apr. 2018. URL: https://www.therobotreport.com/10-biggest-challenges-in-robotics/.

[6] *NAO the humanoid and programmable robot: SoftBank Robotics*. URL: http://www.softbankrobotics.com/emea/en/nao.

[7] Hal Hodson. *Robot language tutors to get kids up to speed before school*. Nov. 2015. URL: http://www.newscientist.com/article/mg22830492-500-robot-language-tutors-to-get-kids-up-to-speed-before-school/.

[8] Mohd Azfar Miskam, Syamimi Shamsuddin, Mohd Ridzuan Abdul Samat, et al. "Humanoid robot NAO as a teaching tool of emotion recognition for children with autism using the Android app". In: *2014 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*. 2014, pp. 1–5. DOI: 10.1109/MHS.2014.7006084.

[9] Hal Aral and Angelica Vecchio-Sadus. "Toxicity of lithium to humans and the environment—A literature review". In: *Ecotoxicology and Environmental Safety* 70.3 (2008), pp. 349–356. ISSN: 0147-6513. DOI: https://doi.org/10.1016/j.ecoenv.2008.02.026. URL: https://www.sciencedirect.com/science/article/pii/S0147651308000742.

Executive Summary

Stephen Lamczyk (CpE), Jacob Strother (EE), Zephaniah Amonoo-Harrison (CpE), and Seth Cummings (EE) are the group members for this project. Megan Witherow is the PhD student advisor and Dr. Khan Iftekharuddin is the faculty advisor. The purpose of this project is to use humanoid NAO robots to simulate a problem setting where one robot, Robot A, will model multiple behavioral states, and another robot,Robot B, will communicate and perform behaviors to move Robot A toward a target state.This project has implications in improving human/robot relations and interactions. The Vision Lab has two NAO robots, which will both be used in this project, each equipped with two five-megapixel cameras, four omnidirectional microphones, temperature sensor,gyroscope, accelerometer, and multiple position, tactile, force-resistive, and ultrasonic sensors. In addition, there are several LEDS placed in key locations on the robot such as the eyes and ears. The eye LEDs can have their color changed using a simple inputted hex code. The NAO robots are fully programmable using Python 2.7 and the NAOqi API. There is also a GUI application called Choregraphe that can be used to make block diagrams with programmable blocks. Test controllers for the behaviors of each robot will be designed and implemented. Each controller will require a behavior module that manages behaviors and communications performed by the robot, a recognition module to classify the behaviors of the other robot, and a reinforcement module to adjust behaviors based on feedback from the other robot. The reinforcement module can be thought of as essentially a state machine that facilitates communication between the behavior and recognition module.

The behavior module handles the communication between robots. The following will be used to communicate behavior: eye color and voice modulation. There are three possible behavior states per robot: a negative state, a neutral behavior, and a positive behavior. Each behavior state controls two possible robot attributes: eye color and communication. For robot A it's eye colors will correspond to its current mood, while for robot B, it's eye colors will correspond to its current mood towards Robot A. With regard to the voice modulation, robot A will be responsive to robot B's prompts only at certain moods. While robot B will always be trying to communicate with robot A in some manner, either by actively trying to change robot A's state, or by trying to maintain its state. Details can be found in tables 1 and 2 for both robots' mood.

The recognition module is supposed to recognize the behaviors from the other robot. For example, robot A's recognition module will monitor robot B for its behavior and output this behavior to the reinforcement module. The reinforcement module will then in turn modify it's behavior to suit this accommodation from robot B. Similarly, robot B will monitor robot A's mood. As robot A's mood worsens, the recognition module will detect this and output robot A's mood to the reinforcement module to output a behavior to improve robot A's mood.

Fig. 7: Example of Robot A and Robot B Interacting With One Another

Appendix

TABLE 3: Past Meeting Schedule

Meeting Date (Members)	**Topics**
9/1/21 (Seth, Stephen, Jacob and Zephaniah)	Introduction to project. Determine whether we want to program real or virtual NAO robot. Relevant literature given.
9/8/21 (Seth, Stephen, Jacob and Zephaniah)	Discussed whether to go with the virtual or physical robot based upon Stephen's findings. Decided to go with the real robots.
9/15/21 (Seth, Stephen, Jacob and Zephaniah)	Discussed overall project design. Made plans to meet day after.
9/16/2021 (Seth, Stephen, Jacob and Zephaniah)	Developed overall project design. Sent it to Megan for feedback.
9/22/21 (Seth, Stephen, Jacob and Zephaniah)	Split up project based upon everyone's strengths and comfort levels with various modules.
9/29/21 (Seth, Stephen, Jacob and Zephaniah)	Discussed possible method to implement the recognition module. Creating our own network from scratch or using built-in functionality.
9/30/21 (Jacob, Stephen, Zeph)	Experimented with the NAO robots and tried to get familiar with the API.
10/6/21 morning (Seth, Stephen, Jacob and Zephaniah)	Further discussed the implementation of recognition modules either from scratch or using built-in functionality.
10/6/21 afternoon (Jacob and Stephen)	Built flow charts for the different recognition modules.
10/12/2021 (Seth, Stephen, Jacob and Zephaniah)	Used the flow charts to implement one large master plan for the entire project. Proposed multiple alternative designs and decided to perform feasibility tests to determine which design is the best.
10/13/21 (Seth, Stephen, Jacob and Zephaniah)	Met with Dr. Khan. Discussed our progress with him. Decided to establish a series of future deadlines so we can manage and keep track of our pace.
10/14/21 (Jacob and Stephen)	Were able to get the NAO robots to recognize each others voice.
10/19/21 (Seth, Stephen, Jacob and Zephaniah)	Experimented with getting the NAO robots to recognize and classify each others eye color.
10/20/21 (Seth, Stephen, Jacob and Zephaniah)	Discussed environmental impacts of the NAO robots. Discussed making robot behaviors realistic.
10/21/21 (Jacob, Stephen and Seth)	Were able to get the robots to recognize each other's eye color. Decided on the following implementation for moods: Red for Mad, Green for Happy, and Blue for Neutral.
10/26/21 (Jacob, Stephen, and Zeph)	Discussed how to use LaTeXto write the report. Discussed the report overall
10/27/21 (Jacob, Stephen, Zeph, Seth)	Discussed the report and necessary changes.
11/3/21 (Jacob, Stephen, Zeph)	Discussed the presentation.
11/10/21 (Jacob, Stephen, Zeph,Seth)	Discussed possible blocks of times to reserve the NAO robots from the museum. Discussed the use of robot pose as an additional behavioral variable.
11/10/21 (Jacob, Stephen, Zeph,Seth)	Discussed possible blocks of times to reserve the NAO robots from the museum. Discussed the use of robot pose as an additional behavioral variable.
11/17/21 (Jacob, Stephen, Zeph,Seth)	Discussed the use of robot pose as an additional behavioral variable and further discussed possible ways to determine if a improvement was present over using no poses.
11/18/21 (Jacob, Stephen, Seth)	Experimented with the NAO robots and various poses. Decided that extra poses would result in excess complexity to the behavioral module.
12/3/21 (Jacob, Stephen, Zeph, Seth)	Discussed the final proposal. Discussed necessary changes to the midterm proposal.

TABLE 4: NAO 6 Robot Specifications

Category	**Specifications**
Height	574 millimeters (22.6 in)
Depth	311 millimeters (12.2 in)
Width	275 millimeters (10.8 in)
Weight	5.48 kilograms (12.1 lb)
Power supply	Lithium battery providing 62.5 Wh at 21.6V
Autonomy	90 minutes (active use)
Degrees of freedom	25 degrees
CPU	Intel Atom E3845 Quad Core @ 1.91 GHz
RAM	4 GB DDR3
Storage	32 GB SSD
Built-in OS	NAOqi 2.8 (openembedded-based)
Compatible OS	Windows, Mac OS, Linux
Programming languages	C++, Python, Java, MATLAB, Urbi, C, .Net
Simulation environment	Webots
Cameras	Two HD OV5640 67.4°DFOV cameras
Sensors	36 x Magnetic Rotary Encoders using Hall-effect sensor technology 3-axis gyrometer 3-axis accelerometer 8 x Force Sensitive Resistors 2 x bumpers located at the tip of each foot Sonar: 2 emitters, 2 receivers. Frequency: 40kHz 2 x I/R. Wavelength = 940 nm. 4 omnidirectional Microphones Capactive Sensor
Connectivity	Ethernet, Wi-Fi IEEE 802.11 a/b/g/n

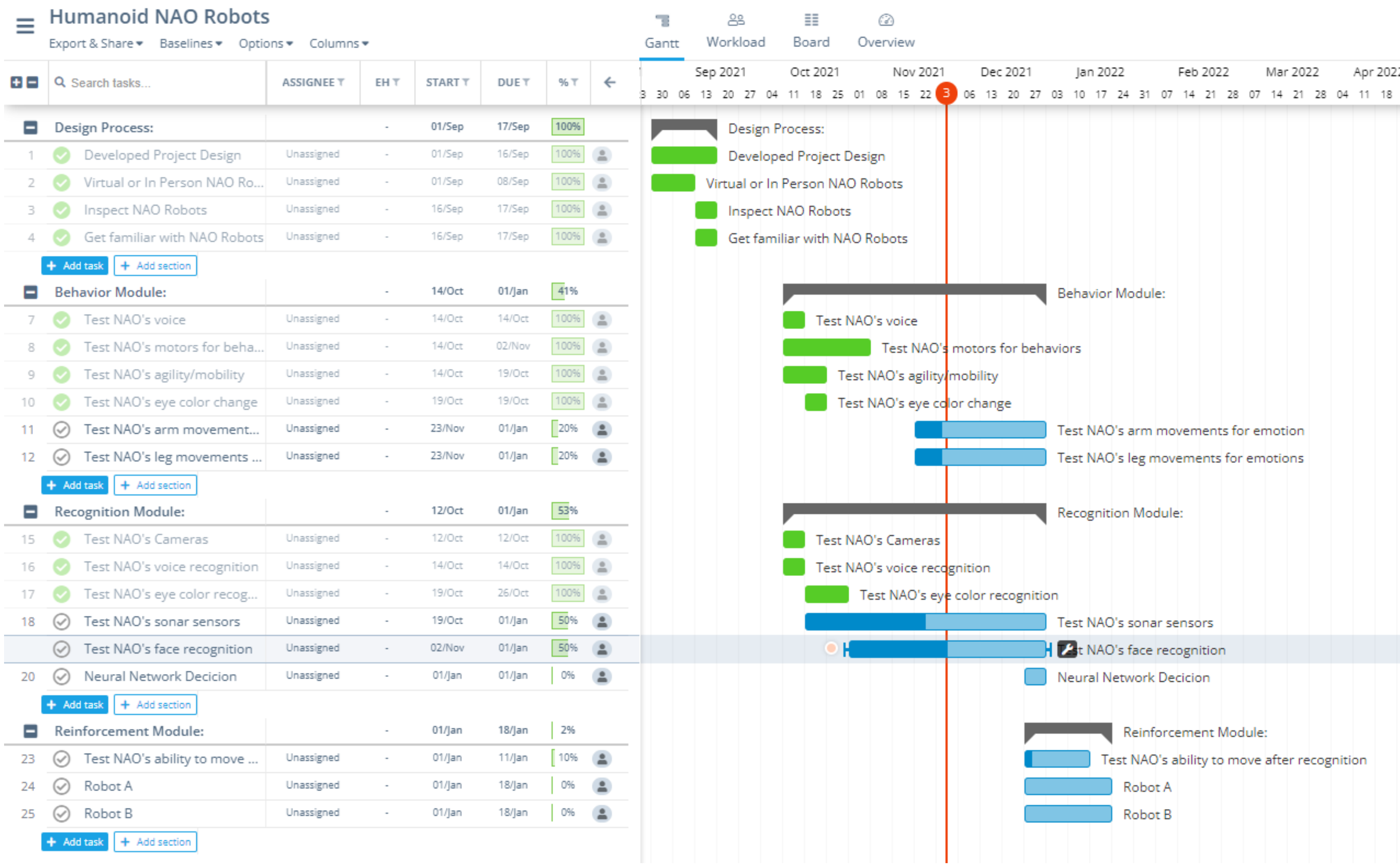

Fig. 8: Gantt Chart Representing our Milestones and Goals