Robotics 204: Introduction to Human-Robot Systems
Winter 2025

Lab Title: Non-verbal communication using head movement

Submission Type: Individual submission

Necessary Materials: Communication Lab Kit, Laptop

Learning Objectives:

1. Implement a posture-based non-verbal communication pipeline to control a robot.
2. Compare and contrast the effect of adding robot control states on the system usability.
3. Critique design decisions for a non-verbal communication strategy.
4. Identify alternate non-verbal communication design ideas to support human-robot communication usability.

Introduction:

In class, you have learned about non-verbal communication and the importance of gains when devising a multi-state system. There are many forms of non-verbal communication; for this lab, you will be focusing on gestures.

Gestures communicate many different types of information. In human-human communication, gestures convey emotion, reveal intent, signal attention, and indicate direction. Minute differences in human body language convey much information despite the smallness of their movements. In this lab, a robotic embodiment will communicate with lab members by imitating the sweep of an arm to direct a person's attention to specific symbols.

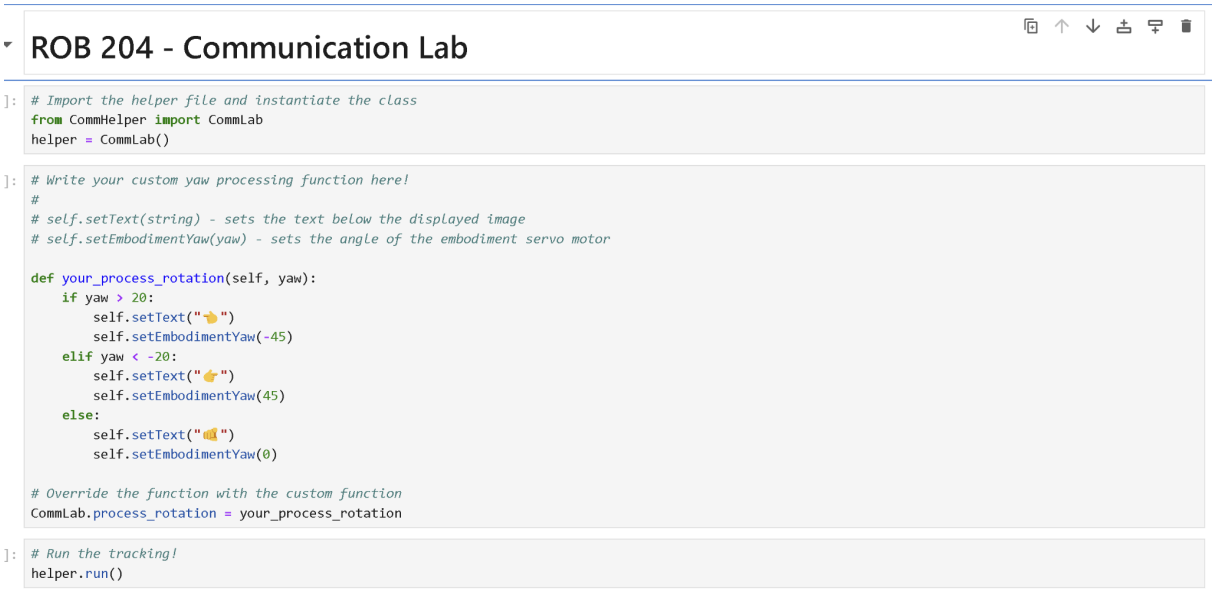
This lab also focuses on the concept of gains, or how a system determines the magnitude of its response to an input. For example, if you increase the sensitivity of a computer mouse, this is equivalent to increasing its gain — a small input now generates an even larger response. In this lab, we will explore how to alter and tune gains within a system to communicate many possible states. The robotic embodiment uses the head movement of a person as an input and the rotation of a servo motor as an output. Scaling these two variables so the response is easily understood by participants is part of the challenge.

In Week 1 of this lab you will attempt to communicate using only head movements and Morse code (Part 1 through 3). In Week 2 of this lab, you will implement at least two extra states to the software and attempt communication using a custom method (Part 4 through 6).

Week 1

Part 1: Setup

1. This lab functions similarly to the method employed by Mark Watney in the movie Martian. For inspiration, feel free to watch this video: https://www.youtube.com/watch?v=NttUBB98zg4
2. For this lab, you will use a Jupyter notebook to interface with the lab hardware and software. Go to this spreadsheet—Link—and select the link for the labeled kit that your group has been given. Note, it may take a few minutes for the link to update.
3. See Appendix B for a primer on Jupyter and Python. The software you have been provided includes three states: Point Left, Neutral, Point Right (see figure below). Test the gaze response by selecting 'Start Tracking'. As you move your head to these three locations, you should observe the embodiment behaving as expected on the three state setup.



ROB 204 - Communication Lab

```
# Import the helper file and instantiate the class
from CommHelper import CommLab
helper = CommLab()
```

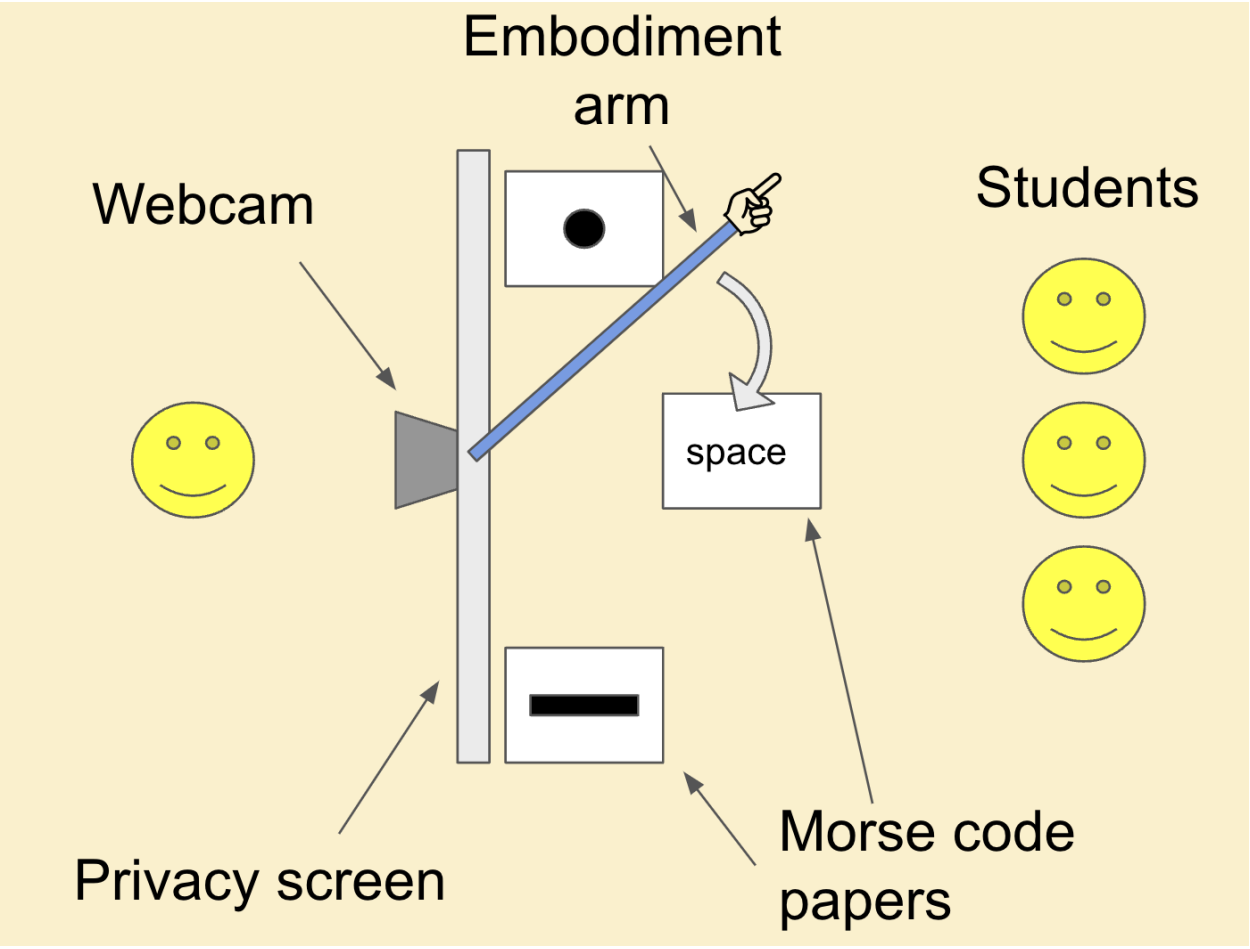
```
# Write your custom yaw processing function here!
#
# self.setText(string) - sets the text below the displayed image
# self.setEmbodimentYaw(yaw) - sets the angle of the embodiment servo motor

def your_process_rotation(self, yaw):
    if yaw > 20:
        self.setText("👈")
        self.setEmbodimentYaw(-45)
    elif yaw < -20:
        self.setText("👉")
        self.setEmbodimentYaw(45)
    else:
        self.setText("👊")
        self.setEmbodimentYaw(0)

# Override the function with the custom function
CommLab.process_rotation = your_process_rotation
```

```
# Run the tracking!
helper.run()
```

Deliverable: None

Part 2: Three State Exploration

1. For this section of the lab, you will attempt to communicate with a group member using only the head movement tracking software, the embodiment, and Morse code. Set up the components of the lab as in the diagram below.



2. Appoint one group member to be behind the privacy screen. Experiment with how quickly the motor responds to head movements, and ensure that the group is in agreement for which direction (left or right) corresponds to each Morse code symbol.
3. Devise a system to communicate edge cases within the setup, like how to repeat a symbol, indicate a new letter, or communicate when the word is done. Remember, once a student is behind the privacy screen no verbal communication may happen until the other group members announce their translated word, **including starting and stopping**. You will only be communicating one word at a time, so no need to indicate a space between words.
4. Write a brief description of your specific method for using the head-movement based communication method. Specify if you have time delays or specific behaviors to influence the robot arm movement to support communication.

Deliverable: Brief description of your method

Part 3: Implementation

1. Once your non-verbal method is complete, have the person behind the screen generate five possible words using this link: https://wordcounter.net/random-word-generator. Pick **one** word from the list and use the head-tracking software to spell the word out in Morse code (refer to the table that came with your kit as well as in Appendix A)**. Please refrain from interacting with**

the rest of the group while attempting this communication, even if you hear them translating the word wrong.

2. Group members outside of the screen should watch the embodiment and keep track of the symbols to try and spell the word. This should be done as a group.
3. Once the group member has finished their word, indicate to the rest of the group their guessed word. Mark in the table on the submission template what the word was and if the group was successful in guessing it or not. **Each member should fill out their own submission.**
4. Repeat this process so each group member has a turn behind the privacy screen. You may iterate on the way you implement your communication method after each group member takes their turn. Keep record of the changes you make at each iteration.

Deliverable: Filled out table and writing section

Part 4: Three State Reflection

Discuss with your group the below questions. Write your own responses on the submission template.

1. What was easy and challenging about this method of non-verbal communication? Reflect on your group's capability to communicate and decipher the words. Did your system for addressing edge cases work or did you have to adjust your methods?
2. Beyond the movement of the robotic arm, were there other communication channels that your team used when receiving or providing the coded words? If so, how did those communication channels help? If not, what other communication channels would have made communication more efficient?
3. Create a hypothetical scenario where this Morse code embodiment would be useful. Provide two limitations you think could emerge from using head tracking as a communication input for your scenario. Explain how you could address these limitations in your scenario.
4. Provide one additional human sense and/or behavior that could be used to non-verbally communicate information when audio is not available. Describe how you could use this additional source of information in your scenario.

Deliverable: Answer the questions on the submission template

Week 2

Part 5: Setup

1. Same setup as Week 1.

Deliverable: None

Part 6: Five State Implementation

1. For this portion of the lab, you will modify the code controlling the embodiment to implement your own language mapping for non-verbal communication. Your new method must add at least two states, for a total of five states. The goal of this exercise will be to communicate a sequence of letters **and** numbers.
2. Open the Jupyter notebook (link) and go to the **process_rotation** function. Note where the yaw thresholds are set to determine how much the servo of the embodiment rotates relative to head movement, as well as how many statements are present to delineate a three-state system.

Editable Function

```
]: def process_rotation(yaw):
       global tracking_output
       if yaw > 20:
           tracking_output.value = "👈"
           setEmbodimentYaw(-45)
       elif yaw < -20:
           tracking_output.value = "👉"
           setEmbodimentYaw(45)
       else:
           tracking_output.value = "👊"
           setEmbodimentYaw(0)
```

3. Setup the components of the lab similar to how they were in Part 2. Remove the Morse code symbols from the screen.
4. Test a few different methods for setting the thresholds on the motor to implement the two extra states. Think about the gain or scale factor that relates the head yaw to the embodiment yaw. IIteratively change the yaw thresholds and gains to finalize your new embodiment software and method for communicating. Use the provided paper to come up with the new symbols for your communication method.
    a. Note: The servo yaw limits are -90 degrees to 90 degrees. For the camera yaw limits, you will have to experiment to see what the practical limits are.
5. Once your group is satisfied with your setup, **take a picture of your setup.**

Deliverable: Take an image of your setup. Copy paste your new process rotation function and describe it.

Part 7: Iterate and Evaluate the Communication Task

1. For this exercise, your group will only do **one** iteration of the communication task. Choose one person to sit behind the privacy screen. Have them go to this website, https://www.gigacalculator.com/randomizers/random-alphanumeric-generator.php, and generate a sequence of numbers and letters. Use the alphabet and 0-9 as possible characters. Your generated sequence must be at least 4 characters long, but you are welcome to challenge yourselves and choose a longer set to translate.
2. **You may do trial runs of your method, but once you are ready to evaluate, you will have one chance, so be sure you are ready!** Keep a record of the iterations you made prior to the evaluation.
3. Using your custom method of communication, the group member behind the screen will attempt to spell the alphanumeric sequence. The rest of the group may work together to parse the symbols into letters and numbers. **Please refrain from interacting with the rest of the group while attempting this communication, even if you hear them translating the word wrong or getting confused.**

Deliverable: Fill out the corresponding section on the submission template.

Part 8: Five State Reflection

You may discuss with your group the questions below but each member should write their own answers

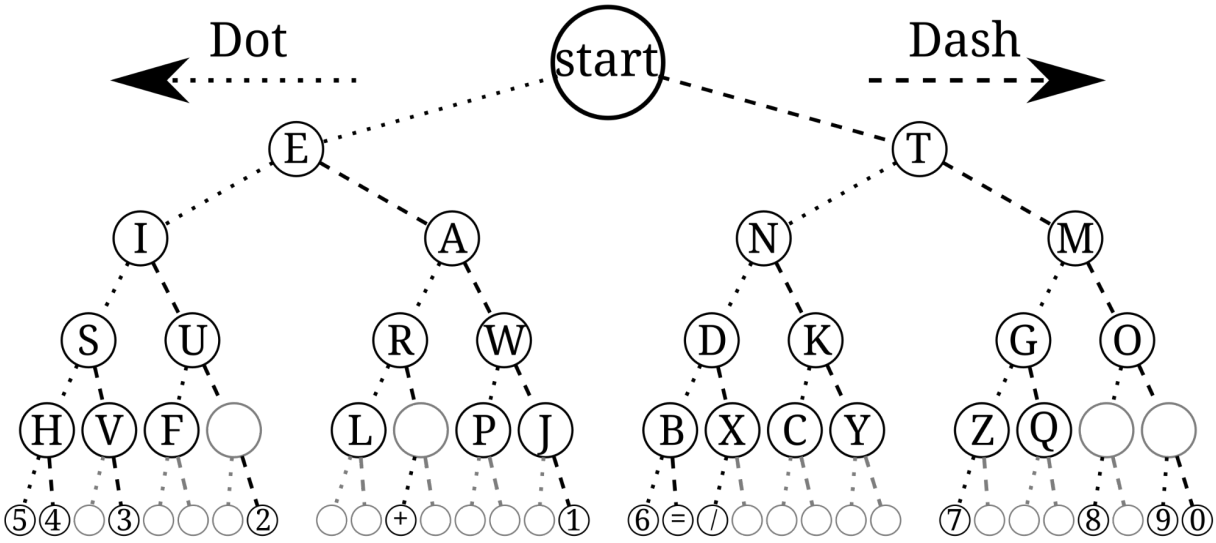
1. If your final iteration worked, explain why it was successful. If there were still errors, explain what challenges your group faced.
2. If you were able to extend the algorithm to permit more degrees of freedom of robot motion (e.g., moving up and down), how would you modify your communication mapping design to improve ease of use? Describe why you would select these modifications.
3. If you were able to add display modalities for the person receiving information (e.g., LED displays) and/or decoding automation, how would you incorporate these capabilities in your design? Provide the pros and cons of these design changes. Describe two potential system errors or edge cases that could influence your new design.

Deliverable: Answer these questions on the submission template.

Appendix A

Morse Code Table

Morse Code Tree - Alternative way of reading morse code



Appendix B: Jupyter and Python Primer

Running a Jupyter Notebook:
To run the Jupyter Notebook, start by running the first cell. This can be done by clicking on the run button the the top bar of the screen, or by pressing shift + enter when being the in code block you wish to run. Once the code block finishes running, your cursor will move to the next block and you can continue the process of running the cells.

If you need a reference for Python syntax, use this link. Please also feel free to ask the IAs or GSI for any Python help!