HYBRID HUMANOID ROBOT (HHR)

Prepared for

Tomonari Furukawa

Professor and Zinn Faculty Fellow
Department of Mechanical and Aerospace Engineering
University of Virginia

Written By

Karthik Eswarapragada, Austin Davis, Lucas Murphy, Orlando Sadek, Sagar Sapkota, and Yushek Sitoula

Undergraduate Students
Department of Mechanical and Aerospace Engineering
University of Virginia

May 10, 2024

Introduction

Humanoid robots are currently being designed to replace the need for humans to be present in dangerous and hazardous environments. As the name implies, the robot intends to provide human-like functionality and dexterity which can serve as a one-to-one replacement for existing workers. Sponsored the U.S. Navy, this project consists of the development of a hybrid humanoid robot (HHR) for its use in various naval ship settings.

Artificial intelligence has brought Naval engineers' new ideas. Fully automated ships were an idea that was sought after until monetary costs were considered. However, most of the Navy's existing fleet consists of human-operated vessels and it would cost many billions of dollars more to implement these vessels with autonomous functionality (Verma, 2022). Additionally, according to the 2022 United States GAO report on uncrewed maritime systems, the Navy's planned average annual budget for the next 30 years was $34.1 billion where 96% of that budget would be allocated for crewed vessels (United States Government Accountability Office, 2022). Additionally, several existing naval ship contracts including the construction of the Virginia class submarines and Ford class aircraft carriers consist of manned vessels (HII, 2024). The military has a sought-after goal of building an "armada of drone ships that use artificial intelligence instead of sailors to fight at sea" (Verma, 2022). Since it is not economically viable or realistic for the Navy to abandon their current fleet and invest billions of dollars on new ships that incorporate fully autonomous capabilities, they are seeking solutions that are more cost-effective and will begin to integrate robotics and autonomous technologies into their existing systems.

Also, the existing market for off-the-shelf robots is extremely limited. Many of these products are designed to be operated in very controlled environments through design choices including fully wheeled functionality, small profiles, and minimal load carrying capacities. HHRs are increasingly more limited, with their primary problem being that they are traditionally extremely difficult to operate and require significant training to become efficient when controlling each arm. Due to many of these factors, there is currently no HHR designed for optimal performance in a naval ship setting. Our goal is to create this design with the intent that it will be used and designed for effective operation in a naval ship setting and has a relatively simple ease of operation.

In order to simplify the design, this project aims to incorporate a compliant foot/wheel mechanism to facilitate a dual-mode mobility system, enabling the robot to intuitively switch between wheeled movement using four points of contact and bipedal walking as required by its environment. By creating a compliant mechanism for this feature, we hope to make the robot much easier and simpler to operate as opposed to having to manually control each single component on the device.

Many naval ships contain crucial components that either contain extreme temperatures or have exposed chemicals and materials that could be dangerous to humans. By creating robots such as these, the lives of operators and naval seamen deployed in these ships can be protected from the risk of injury or death and the future for fully autonomous naval systems and ships will become more feasible.

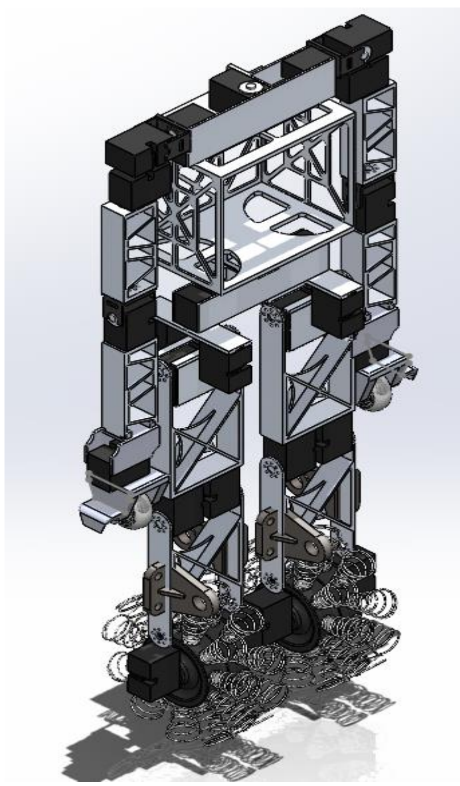


Figure 1: Conceptual Image of HHR

Bipedal robots, distinguished by having the ability to walk on two feet, offer unparalleled adaptability to human-centric environments. For this project's purposes, the bipedal robot's capacity to climb stairs and ladders is integral. The ability to seamlessly navigate spaces designed for humans positions bipedal robots as promising candidates for applications in homes, offices, and public spaces (Pratt et al., 2007). The humanoid appearance further fosters natural interactions and integration into human-centric environments. However, these benefits come with inherent challenges. Bipedal robots often exhibit slower movement, making them less suitable for scenarios where rapid movement is imperative. Additionally, challenges in achieving stable movements and the demand for high energy consumption present formidable obstacles, impacting the practicality of these robots in certain applications. In naval ship settings, it is also common for conditions to change from movement and propulsion of the ship and surface conditions from waves and weather conditions (Zhu, 2017). Due to these factors, bipedal robots can be subjected to poor obstacle navigation and stability in these given environments. Furthermore, the limited payload capacity poses restrictions on the types of tasks and functionalities that bipedal robots can effectively perform (Kajita et al., 2003).

In contrast, wheeled robots emerge as efficient and speedy alternatives, offering advantages such as energy efficiency, high-speed operation, and a greater payload capacity (Altagar, 2023). These attributes position wheeled robots as formidable contenders for applications requiring rapid and robust movement, particularly in structured environments like warehouses and manufacturing facilities (Yang & Spenko, 2013). The enhanced payload capacity expands the range of tasks these robots can undertake, making them suitable for scenarios demanding the transportation of heavier loads. Despite these strengths, wheeled robots face notable limitations. Their less-than-human-friendly interaction is a notable disadvantage as it makes social acceptance and integration into human-centric spaces quite challenging. Additionally, the limited adaptability to diverse terrains and the challenges in navigating obstacles constrain their applicability in environments where agility and adaptability are paramount, such as disaster response scenarios or outdoor exploration (Kim et al., 2019).

The primary focus of this initiative is to enhance the robot's adaptability to different terrains, a crucial aspect for naval operations where conditions vary significantly. The primary objectives for this robot are as follows. First, it must be able to be operated through teleoperation from a single device or controller for ease of operation. Second, it must incorporate a compliant foot/wheel mechanism to appropriately adapt its geometry to the given environment based on its use case, switching between a ‘flat foot’ and a wheel when necessary and without activation from the operator. Finally, the HHR must be able to navigate different terrains and obstacles, specifically, through passing a watertight door (Figure 2) frequently seen on naval ships and submarines, as well as climbing a 63-degree ladder (Figure 3).

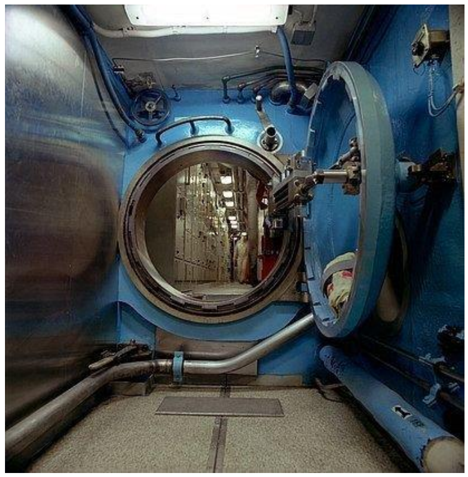


Figure 2: Image of Submarine Watertight Door.

Image courtesy from https://qph.cf2.quoracdn.net/main-qimg-f321ceafaabefb3a38c58924bfaa4a24-lq

Figure 3: Image of Submarine Watertight Door.

Image courtesy from https://qph.cf2.quoracdn.net/main-qimg-fbe99b57812ce9c39a74fe7c194fd82a

This paper will be outlined as follows. First, essential knowledge will be provided to bring context to our team's work within the scope of this general project. This will consist of progress made by other past teams and the knowledge necessary for future work including the fundamental systems and ways in which this HHR operates. Next, the design process will be outlined, describing the fundamental basis for this project and the intended goals set forth by the customer. This section will include listing customer needs, target specifications of the robot, and the proposed solutions to these challenges through concept generation and appropriate selection based on the criteria outlined earlier. The final design section will describe in greater detail the physical designs, including primarily the compliant foot/wheel mechanism, hand mechanism, and controller choice for teleoperation. Next, the latest progress section will display the developed physical system as of April 4, 2024, including specific details about the hardware and software incorporated into its existing design. Validation will cover experimental tests for the physical robot that will ensure or check to see that it is in line with the customer needs outlined earlier. An operations manual will be provided for future teams or customers to understand how they can operate and function the HHR including the software and setup required for operation. Finally, the conclusion will summarize this information and list suggestions for future work, including tasks and design considerations that have still yet to be implemented.

Essential Knowledge

This project is currently in its first year through its work with UVA. The general framework for the project, including roughly 80 percent of the physical design and the motor selection used, was inherited from an unknown external source outside of UVA. The HHR consists of a CNC machined aluminum chassis containing 30 Dynamixel XM540-W270-R servo motors. The general structure of the robot was passed down from its previous team/owner while missing its hand, foot/wheel, and camera/navigation components. The robot in its initial state was not equipped with any electronics outside of the physical components and motors previously installed. In order to accomplish the objectives outlined in the introduction, a central control and computing system needed to be specified and installed in the robot. This includes the need for a central control that can control all 30 motors as well as a wiring system designed to allow each part to be connected and able to move freely and unobstructed. The intended long-term objective for this project is to create a fully wireless and autonomous robot functioning in a naval ship environment according to the guidance set forth by the customers using it.

This project aims to strike a balance between advanced technology and functional utility. It does not seek to revolutionize robotics but to provide a tangible improvement in the operational capacity of robots in specific contexts. The potential applications of this technology in naval settings are vast, ranging from routine surveillance and maintenance tasks to more complex operational roles. The successful implementation of this technology could lead to enhanced efficiency and versatility in unmanned or robot-assisted naval operations.

Design Process

Customer Needs and Target Specifications

In the project's scope, the customer has delineated specific operational requirements for the hybrid humanoid robot. Firstly, it is imperative that the robot be equipped with a compliant wheel/foot mechanism, ensuring adaptability and resilience in its locomotion apparatus. Additionally, the robot must possess the capability to ascend a 63-degree ladder, reflecting a demanding criterion for its climbing mechanics. To accommodate a variety of environmental conditions, the robot's design must prioritize efficiency on both flat and uneven terrains. Operational control of the robot is to be executed via teleoperation, indicating a need for robust remote handling systems. Lastly, the robot is expected to traverse watertight doors, necessitating precise control and balance in its mobility functions. These needs drive the robot's design and functionality, ensuring that the final product is both versatile and reliable in meeting the specified requirements.

The design parameters for the hybrid humanoid robot have been specified, with a focus on some critical components. The foot mechanism and clearance target a range between 12 to 16 inches, optimizing the robot for varied terrain interaction. Movement speed specifications are set to range from 3 to 7 feet per second, ensuring prompt responsiveness. Energy efficiency is marked with a power consumption target between 700 to 900 watts, balancing operational capacity with sustainability. Climbing speed has been set to aim for 0.2 to 0.5 steps per second, which is critical for navigating ladder ascents. Lastly, the robot is engineered to handle a significant load, with a capacity ranging from 50 to 150 pounds, enabling it to perform a variety of tasks.

Concept Selection and Concept Generation

To define the HHR’s design target specification, a ranking system was used to assess customer needs across technical, importance, implementation difficulty, and priority. Each requirement—compliant wheel/foot mechanisms, ladder climbing, teleoperation, terrain navigation, and watertight door traversal—was ranked with specifications in mind. The specifications include power consumption, total mass, foot clearance, max load capacity, walking speed, climbing speed, and processing Power and RAM which are necessary to achieve customer needs. An example of the specification table used in concept selection is shown in Figure 5. After done with the ranking we were able to conclude that power and walking speed were some specifications that ranked high on technical priority while foot clearance ranked high on technical difficulty. This method guided focus towards integrating critical features efficiently, ensuring a robot that meets key needs while optimizing design and functionality.

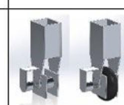
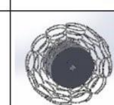
	Design ->		Inflatable Foot/Wheel		Inflatable Foot/Wheel with Vertical Supports		Spring Supported Design	
								
#	**Selection Criteria**	**Weight**	**Rating**	**Weight**	**Rating**	**Weight**	**Rating**	**Weight**
1	Compliant Mechanism	70%	2	1.4	**2**	1.4	5	3.5
2	Ease of Manufacturing	5%	2	0.1	3	0.15	**4**	0.2
3	Cost	2%	3	0.06	3	0.06	**5**	0.1
4	Speed to Manufacture	3%	2	0.06	2	0.06	4	0.12
5	Rolling Stability	10%	**5**	0.5	4	0.4	4	0.4
6	Climbing Stability (Flat Foot)	10%	1	0.1	5	0.5	3	0.3
		100%		2.22		2.57		4.62

Figure 4: Concept Selection Table for Compliant Foot/Wheel Design

An essential component to the operation of the HHR is the teleoperation and power system that will be responsible for operating the robot remotely. Using the approach outlined above, a remote transmitter was selected to operate the robot according to the use situations in a naval ship described earlier. As a result, the Logitech F310 Remote Controller, shown in Figure 6, was selected to operate the HHR. The Logitech controller is plugged into a USB port in the computer that runs the motion code, then there is a python function used, gamepad.py, that allows the controller to send inputs into the computer. Using all the available buttons and switches on the Logitech controller, the code will decide which components (motors) will move at one instance, such as the legs or the arms. This will permit individual movements of the arms and legs as well as default movement settings when the switches are oriented according to the operator's intent. The joystick controls will adjust accordingly to the goal of the engineering team at the time and can be programmed to have the motors individually achieve specific locations.

Figure 5: Logitech F310 Remote Control Device

Final Design

In order to navigate the obstacles of a naval ship, such as a watertight door and a 63-degree ladder, while remaining efficient on flat surfaces, a compliant foot/wheel mechanism has been requested by the customer. To be compliant, the foot mechanism must be able to adapt and deform appropriately based on its loading condition and intended use case without activation by the operator. Many current hybrid robot designs incorporate a foot/wheel mechanism that must be activated to switch between a rolling and bipedal loading configuration. To avoid unauthorized use of protected designs of similar mechanisms such as that used in a tank's wheel design, our customer has requested that the design of this mechanism is original and has not been used in preexisting robot designs. Consequently, this HHR will feature a "flat tire" foot/wheel design where an axially rolling wheel will be supported by conical compression springs along the face the rim as shown in Figure 6.

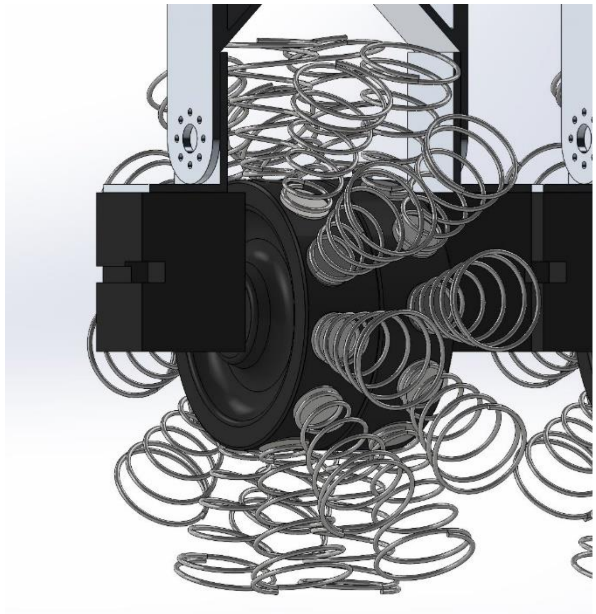


Figure 6: Computerized Model of Compliant Foot/Wheel Mechanism

Because of the request for this mechanism design to be original, the design includes the integration and combination of several different off-the-shelf parts. The materials to construct this specific design were ordered through the McMaster-Carr online catalog. Each wheel mechanism consists of two 4” polyurethane wheels and 22 conical compression springs (McMaster-Carr Part 1692K44). These springs were selected according to the desired displacement necessary to create a sufficient flat base in point-loading situations such as when navigating a 63-degree ladder. Additionally, the maximum height of the wheels was found experimentally to be 8.2” so that they did not interfere with the remainder of the robot housing. As a result, we decided to choose springs with 2” lengths such that the total height of the wheel profile was exactly 8”. This choice of spring length permits enough deflection such that other springs adjacent to the ground can be activated without requiring the entirety of the spring to flatten. This would also more easily allow the deflection in rolling situations to be relatively minimal, allowing the wheel to retain its circular shape.

According to the CAD model created in SOLIDWORKS, we found that the overall weight of the robot (with both the hand and foot/wheel mechanisms installed) was 22.78 lbs. Given that this robot is currently in its first iterations of design and will likely incorporate other components, such as an onboard computing unit with an ample number of batteries to support a long charge life, we approximated the load on the springs in a point-load situation to be 32 lbs.

For our intended deflection, we sought for the springs to deflect roughly half their length in point load situations to allow adjacent springs to be activated – providing a greater footprint and stability on the ground. For this proposed design as shown in Figure 6, we searched for appropriate conical compression springs through McMaster Carr. The selected springs were based on the parameters outlined above including the 2” desired length and were selected primarily based on the spring rate options provided. Due to the target deflection of the springs, a conical spring with a spring rate of 7.77 lbs./in. It was important to ensure that the springs selected were long enough such that they did not bottom out and approach its solid length. Due to the diameter of the wheels, 2” springs were selected to find this medium between being soft enough to deflect in point-load situations while also being rigid enough to remain in a circular shape during wheeled situations. Since this value for the spring rate was slightly higher than our calculated target spring rate of 6.51 lbs./in, we decided that ballast weight and other methods such as the incorporation of heavier components including batteries and power supplies could be used in order to increase the amount of deflection of the springs.

The hand/wheel design was made to support the robot as it moves around the ship. The wheel helps the robot move in 4-point contact where it would be able to roll and navigate around the ship more easily. The hand is used to guide the robot and change direction as needed by rotating the motor which is attached to the hand. The wheel is an Easy-Turn Soft Polyurethane Wheel which makes it easier to make trunks around tight corners. The hand also has a hook-like attachment which is used to support the humanoid robot while climbing up the ladder. The hook would wrap around the railing of the ladder so it can slide up as the robot climbs up.

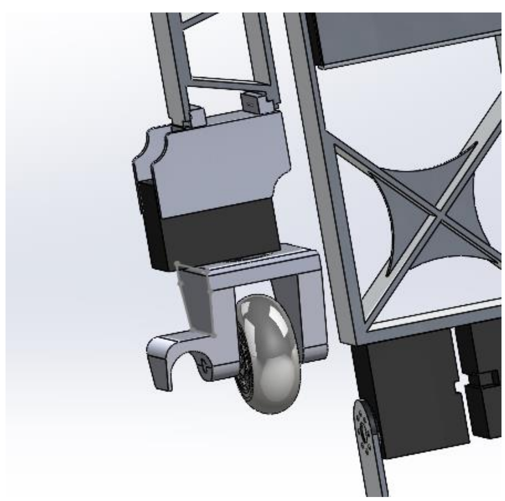


Figure 7: Compliant Hand/Wheel Final Design

The final design for controller mapping is listed below in Figure 8. This is how the team would like to get the motors to operate when the robot is in the final design stage

Figure 8: Controller Mapping Design

Mechatronics progress started in reverse engineering the robot to understand what motors are used and how they are run. The first step was to find wiring diagrams for the Dynamixel motors, and it was found that they are 4-pin motors. USB connection with a Dynamixel motor is mostly recommended through a Dynamixel Starter kit, which we have implemented into our system. We use a U2D2 Power hub board to send commands from the computer to individual Dynamixel motors. Pictured below is the Dynamixel motor wiring diagram.

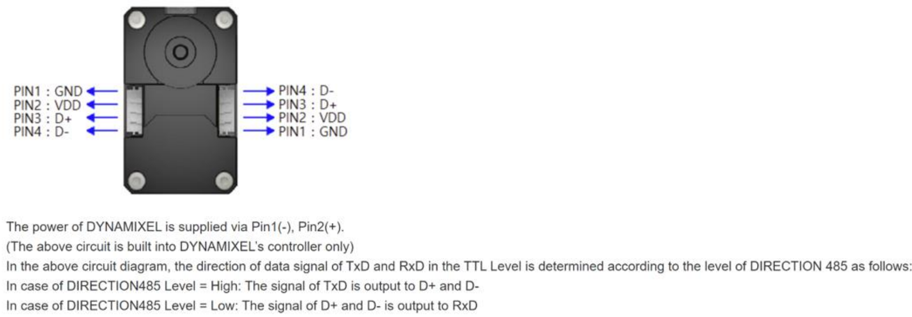


Figure 9: Dynamixel Motor Wiring Diagram

The wiring of the robot has many physical constraints with where the wires can be placed and the fact that all the wires must be connected to one U2D2 board. There are also 4 pin connector hubs in the robot that serve as a place for wires to be connected in series. This is how we connect each robot piece to a central process system. Each motor goes to a 4 pin connector hub and each 4 pin connector hub goes to a U2D2 power hub which then receives inputs from the python code in the computer which is connected to the Logitech controller. There is a limitation in the amount of 4 pin connector hubs available so the wiring must be designed in a way that allows the team to prioritize specific motors for specific actions. The final design for our mechatronic system is shown as it should be in its final stage, pictured as a wiring diagram for the arm and control center and a wiring diagram for the hip-leg connection.

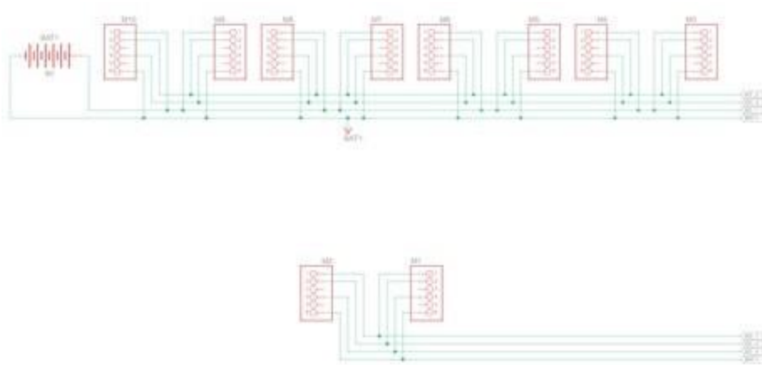


Figure 10: Hip-Leg Connection Wiring Diagram

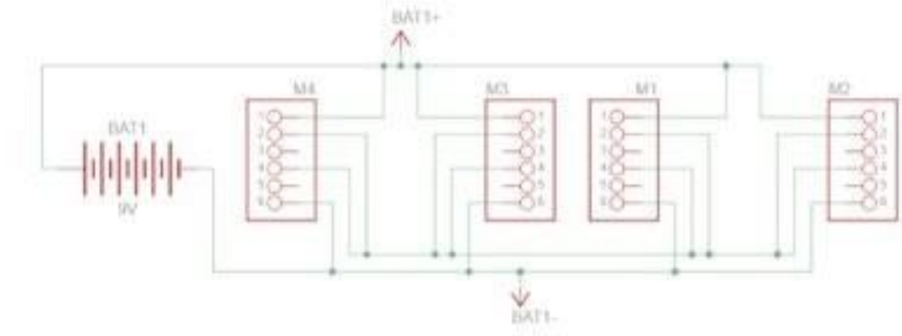


Figure 11: Arm/Control Center Wiring Diagram

Latest Progress

At its current state, the HHR can control each motor that is currently installed on the chassis. Although each motor can be called individually, there is still work to be done in order to intuitively map each of the general movement functions to the gamepad controller. Figure 12 shows the current physical layout of the robot. The compliant foot/wheel mechanism's construction has also been completed and is awaiting installation on the robot. An image of the constructed foot/wheel mechanism is shown in Figure 13.

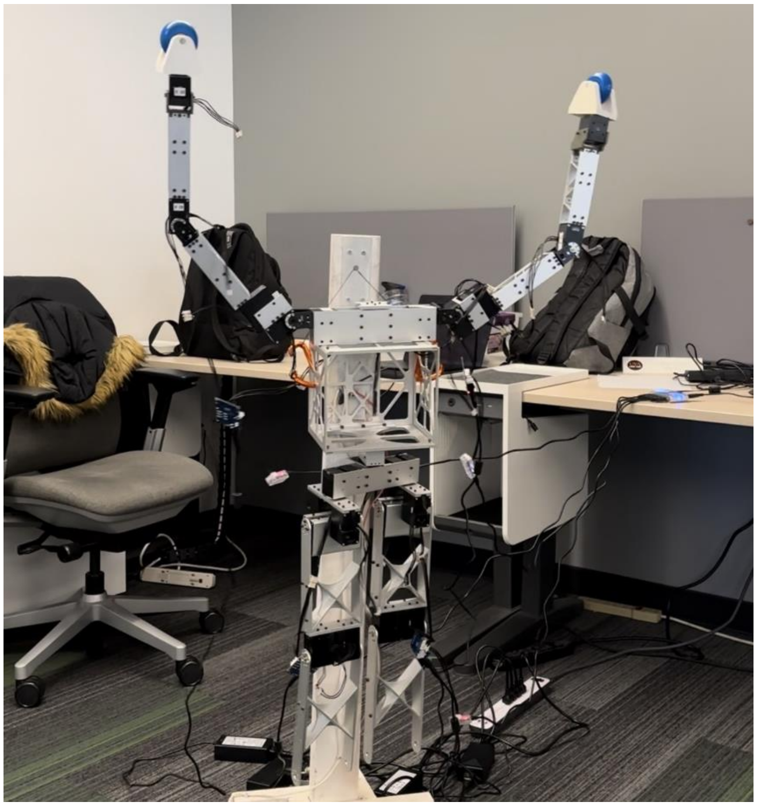


Figure 12: Image of HHR as of 4/1/2024

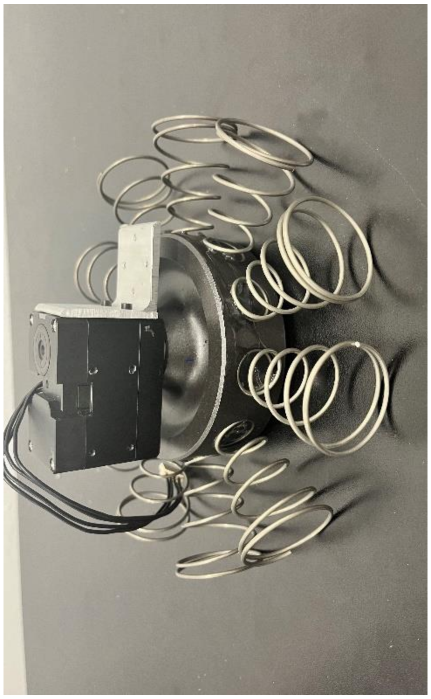


Figure 13: Image of the Completed Foot/Wheel Mechanism

The current development of the humanoid robot control script has successfully implemented the Dynamixel SDK to manage multiple motors across distinct communication ports. This arrangement allows for precise control over each limb, with each set of limb motors (right and left) connected through separate PortHandler instances. The script supports real-time interaction via a gamepad, with the integration of input handling compatible with both Linux and Windows systems.

Each motor is individually initialized with specific torque and position settings and identified uniquely within the script. This setup confirms motor readiness and connectivity, providing immediate feedback on the connection status through console outputs. Key to our implementation is the use of index toggles (**index_right** and **index_left**), which manage the movement range of each limb. These toggles respond to gamepad inputs to switch between two predefined positions (minimum and maximum), thus enabling the precise and independent control of limb movements. The control loop of the script actively reads gamepad inputs to adjust the motor positions. This is done using functions that write to and read from the motors, ensuring that the actual motor positions closely follow the commanded positions. Despite these advancements, the project faces challenges in maintaining synchronization between the motors for fluid limb movements and addressing occasional communication errors that cause motor response delays.

```python
while True:
    # Read the current position
    dxl_present_position, dxl_comm_result, dxl_error = packetHandler.read4ByteTxRx(portHandler, DXL_ID_RIGHT_ROTATOR, ADDR_PRO_PRESENT_POSITION)
    dxl2_present_position, dxl2_comm_result, dxl2_error = packetHandler.read4ByteTxRx(portHandler, DXL_ID_RIGHT_SHOULDER, ADDR_PRO_PRESENT_POSITION)
    dxl3_present_position, dxl3_comm_result, dxl3_error = packetHandler.read4ByteTxRx(portHandler, DXL_ID_RIGHT_TRI, ADDR_PRO_PRESENT_POSITION)
    dxl4_present_position, dxl4_comm_result, dxl4_error = packetHandler.read4ByteTxRx(portHandler, DXL_ID_RIGHT_FOREARM, ADDR_PRO_PRESENT_POSITION)
    dxl5_present_position, dxl5_comm_result, dxl5_error = packetHandler.read4ByteTxRx(portHandler2, DXL_ID_LEFT_ROTATOR, ADDR_PRO_PRESENT_POSITION)
    dxl6_present_position, dxl6_comm_result, dxl6_error = packetHandler.read4ByteTxRx(portHandler2, DXL_ID_LEFT_SHOULDER, ADDR_PRO_PRESENT_POSITION)
    dxl7_present_position, dxl7_comm_result, dxl7_error = packetHandler.read4ByteTxRx(portHandler2, DXL_ID_LEFT_TRI, ADDR_PRO_PRESENT_POSITION)
    dxl8_present_position, dxl8_comm_result, dxl8_error = packetHandler.read4ByteTxRx(portHandler2, DXL_ID_LEFT_FOREARM, ADDR_PRO_PRESENT_POSITION)


    if dxl_comm_result != COMM_SUCCESS:
        print("%s" % packetHandler.getTxRxResult(dxl_comm_result))
    elif dxl_error != 0:
        print("%s" % packetHandler.getRxPacketError(dxl_error))
        print("ID number 3")
    elif dxl2_comm_result != COMM_SUCCESS:
        print("%s" % packetHandler.getTxRxResult(dxl2_comm_result))
        print("ID number 3")
    elif dxl2_error != 0:
        print("%s" % packetHandler.getRxPacketError(dxl2_error))
        print("ID number 3")
    elif dxl3_comm_result != COMM_SUCCESS:
        print("%s" % packetHandler.getTxRxResult(dxl3_comm_result))
        print("ID number 3")
    elif dxl3_error != 0:
        print("%s" % packetHandler.getRxPacketError(dxl3_error))
        print("ID number 3")

    elif dxl4_comm_result != COMM_SUCCESS:
        print("%s" % packetHandler.getTxRxResult(dxl4_comm_result))
    elif dxl4_error != 0:
        print("%s" % packetHandler.getRxPacketError(dxl4_error))
    elif dxl5_comm_result != COMM_SUCCESS:
        print("%s" % packetHandler.getTxRxResult(dxl5_comm_result))
```

Figure 14: Image of Python Code Script

Looking forward, the project aims to enhance error handling to manage communication failures more effectively. Additionally, there is a plan to develop algorithms that will coordinate movements between limbs more naturally. Considering user interaction, the development of a graphical user interface (GUI) is also on the agenda, which would simplify configuration and control processes, minimizing reliance on direct console interaction. Overall, the script has made significant progress in establishing a robust and responsive control system for the humanoid robot, with clear paths identified for future improvements to increase reliability and usability.

Validation

There were tests performed with the hybrid humanoid robot to validate the project's success. The first test was a flat surface rolling motion test: a test to determine the capability of the robot rolling on the floor with its compliant foot/wheel design. The design of the test is one where all 23 motors were coded to move in a way so that the robot could lock into a position and roll on 4 points of contact. The key metrics for success are the ability for the joints to lock in to a stable 4 leg position and the robot being able to roll 6 meters. During the test, the robot was held up by a hoist to decrease the chance of the robot falling over and suffering damage. This hoist was lowered so that the robot could support its own weight while contacting the floor during the experiment.

Figure 15: Flat Rolling Motion Test

This validation test was deemed successful for the act of the motors being able to lock into position and the robot balanced on 4 points of contact. However, since the robot was partially supported by the hoist that was designed for this experiment, this key performance metric was only judged to be 80% successful, since that is the approximation for body weight that was being self-supported by the robot. The hoist also came with a limitation on motion. The designed code for the test was successful in getting the robot to move, but it only reached a distance of 4.6 meters which means performance in this metric was 76.7% successful.

Another validation test performed was the upper body ladder climbing test. The upper body ladder climbing test was devised to assess the hybrid humanoid robot's (HHR) capability to ascend a ladder using its arms. This capability test aimed to mimic scenarios where vertical movement is essential, leveraging the robot's upper body strength and coordination. The test involved predesigned coding that actuated the robot's motors to execute specific movements. The programming was intended to enable the HHR to grasp the ladder rungs securely with its hands and employ its upper body strength to push against these points, thereby lifting itself upward. During the test, the robot was programmed to attempt climbing over six steps of a ladder set up in the testing facility. The HHR's performance was focused on two main goals: securing all four limbs on the ladder's steps and effectively pushing down to lift its body upwards across the designated steps.

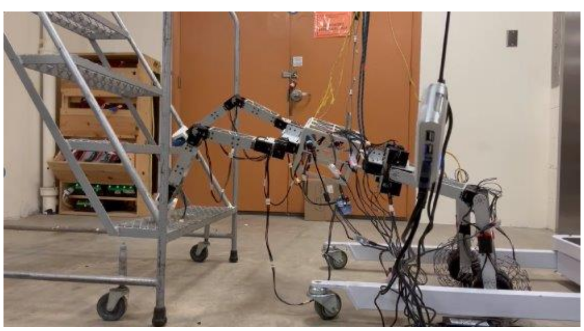


Figure 16: Upper Body Climbing Test

The results were mixed, highlighting areas of both achievement and needed improvement. The HHR managed to secure its hands on the stairs with 50% success. This partial success indicates that while the robot could occasionally grasp the rungs, the consistency of its grip needs enhancement. In terms of climbing, the robot successfully ascended over two of the six steps, equating to 33% success. This result suggests that while the robot can initiate an upward movement, sustaining this action effectively across multiple steps remains a challenge. A significant limitation noted during this test was the absence of advanced balancing logic within the robot’s programming. This shortfall likely contributed to the robot's underperformance in consistently climbing more steps. The lack of a sophisticated balance management system means that while the robot can execute basic climbing motions, it struggles to maintain stability throughout the activity.

Another test was performed on the design aspect of the robot. The experimental foot/wheel deflection test aimed to validate the performance of a compliant foot/wheel design for the hybrid humanoid robot. This design is critical for ensuring the robot can adapt its footing to flat surfaces effectively, resembling more closely the function of a human foot under weight. The design incorporated springs intended to deflect under load, simulating the behavior of the foot/wheel when the robot's weight is applied. The primary goal was to achieve a spring deflection of 1.2 inches, a specific measure chosen because it allows the wheel to flatten sufficiently, mirroring a foot's surface area under similar conditions. During the test, the springs were subjected to controlled loads to measure their deflection and assess the spring constant, as well as the maximum load each spring could withstand.

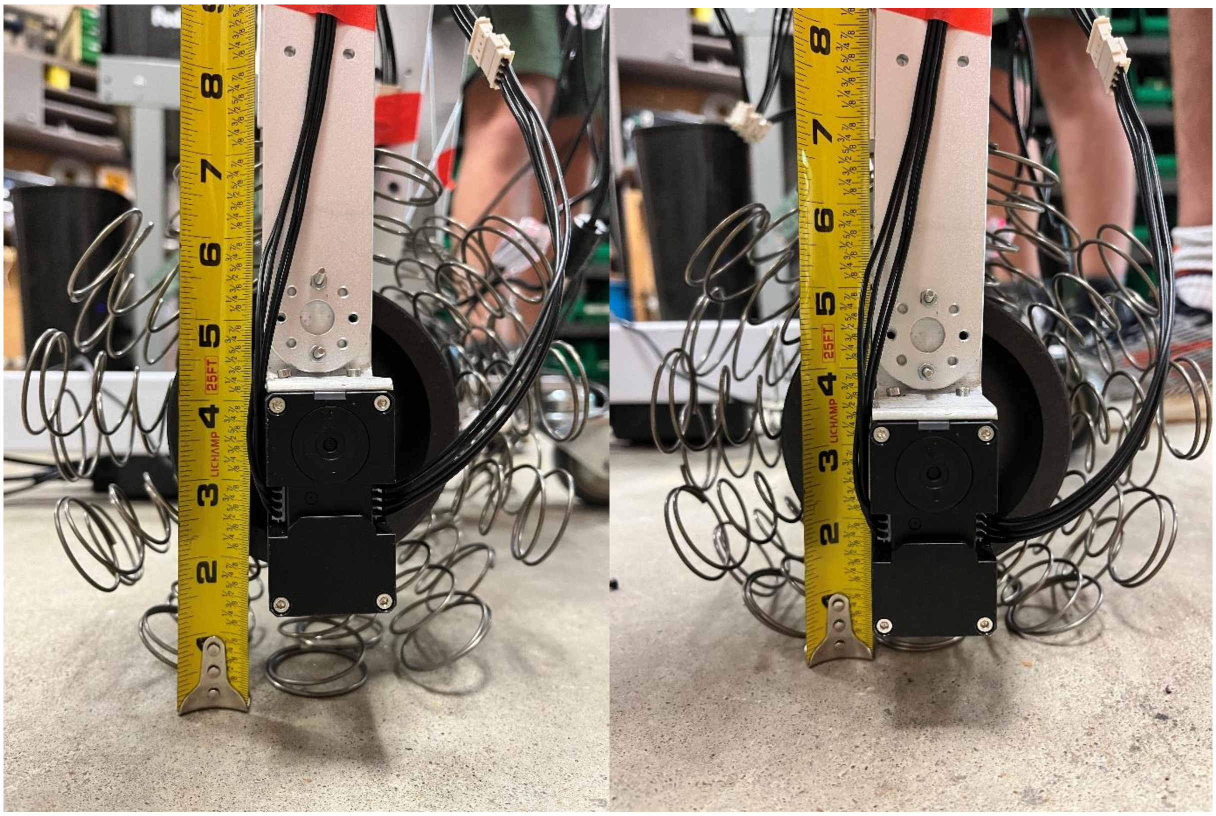


Figure 17: Spring Deflection Test

The springs achieved a deflection of 1.17 inches, representing a 97.5% success rate relative to the goal. This outcome nearly met the target deflection, indicating that the design's response under load is highly effective, though slightly short of the ideal. The measured spring constant was found to be 6.52 lbs/in, and the maximum load sustained by one spring was recorded at 8.7 lbs. These metrics provide additional data points for refining the foot/wheel design to better accommodate the robot’s weight distribution and mobility requirements. While the test was largely successful, the slight shortfall in achieving the exact deflection target suggests room for improvement in the spring design or the load application method. Future tests should consider adjusting these elements to optimize the foot/wheel structure for enhanced compliance and support. This could involve experimenting with different spring materials or configurations to fine-tune the deflection characteristics, ultimately improving the robot's stability and functionality when transitioning across varied terrains.

Operations Manual

This manual provides instructions for operating the hybrid humanoid robot equipped with 30 Dynamixel motors, only 25 are currently connected. It includes the list of components, motor identification, motor positioning, operational control using a Logitech Gamepad, and troubleshooting steps.

Safety Information

- Ensure that the robot is powered off before connecting or disconnecting hardware.
- Avoid physical contact with moving parts.

System Requirements

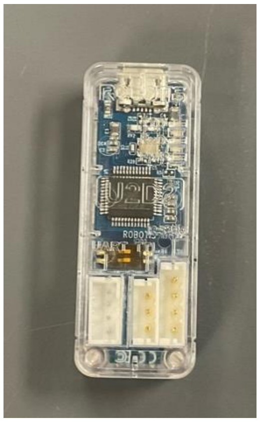
- Dynamixel Wizard software installed on a PC or Linux.
- Python installed on PC or Linux.
- Python IDE installed on PC or Linux.
- Dynamixel SDK library installed on the PC or Linux machine.
- Inputs library installed on the PC or Linux machine to recognize controller inputs.

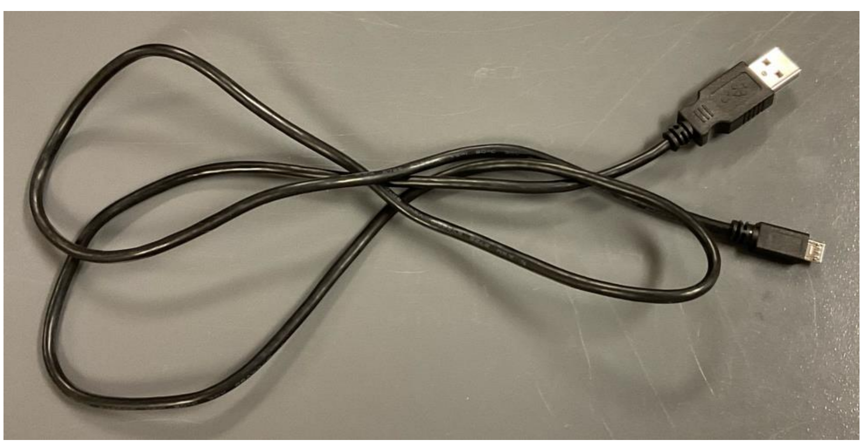
List of Components

- 30x Dynamixel XM540-W270-R servo motors
    - 


- 5x U2D2 Power Hub Boards
    - 

- 5x U2D2 USB Micro-B Boards

- 

- 5x AC Adapters
  - 
- 5x Micro-USB Cables
  - 
- Robot Cable-4P 180 mm

- 

- Robot Cable-X4P 180 mm
  - 
- Robot Cable-X4P (Convertible) 180 mm
  - 
- 4-Pin Connector
  -  
- 4-Pin Connector Hub

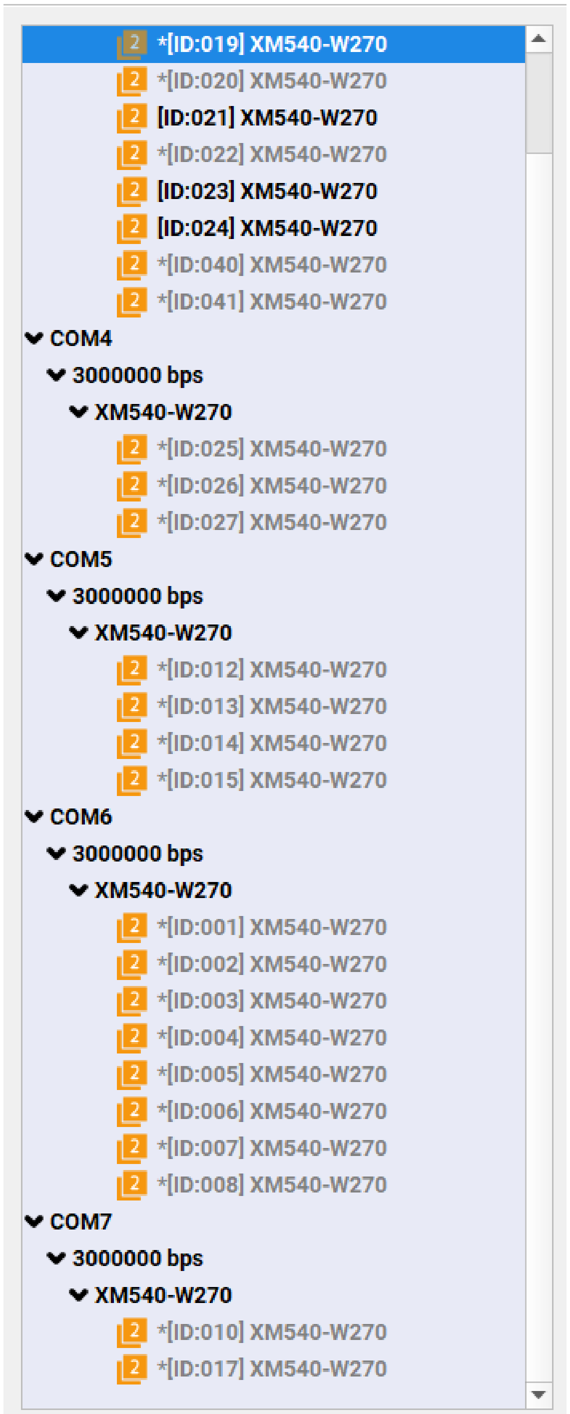
- 
- Logitech F310 Remote Controller
  - 

Setup and Initial Configuration

- Install the Dynamixel Wizard from the official Robotis website.
- Connect the robot to your PC using the U2D2 USB port.
- Ensure the Logitech pad is recognized by your PC.

Identifying Motor ID Using Dynamixel Wizard

- Open Dynamixel Wizard and connect to the robot.
- Use the "Scan" feature to detect all connected motors.
  - Protocol 2 must be selected along with a baudrate of 3,000,000.
- Review and note down the ID of each motor. Adjust if necessary to match the IDs as defined in the Python script (e.g., Right Rotator is ID 1, Right Shoulder is ID 2, etc.).

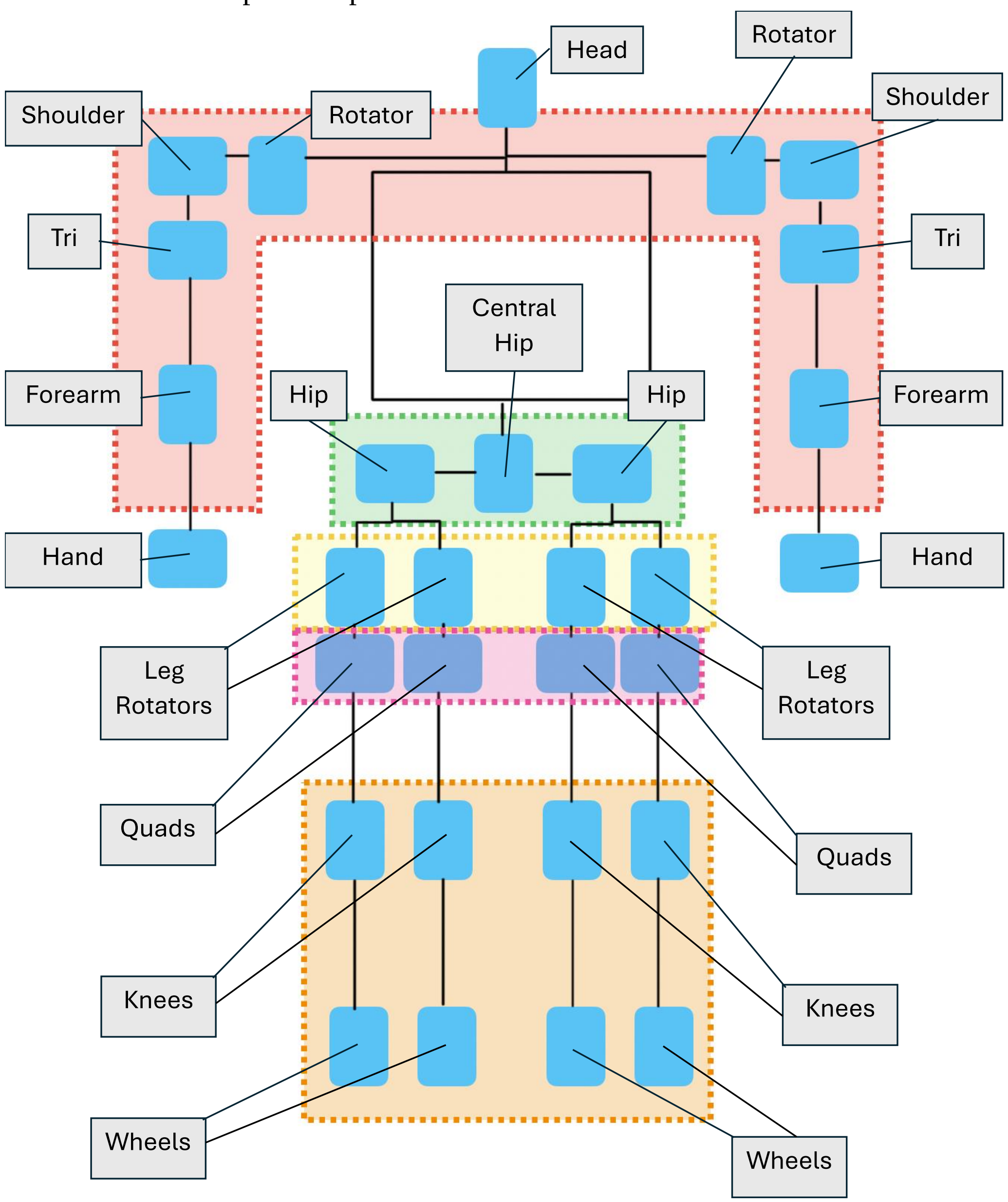


  - 25 motors should be recognized.

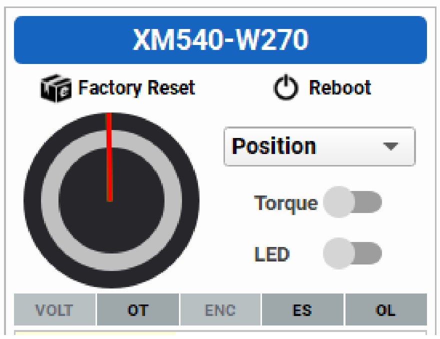
- Use the following Motor ID Chart and Diagram for troubleshooting any connection issues.

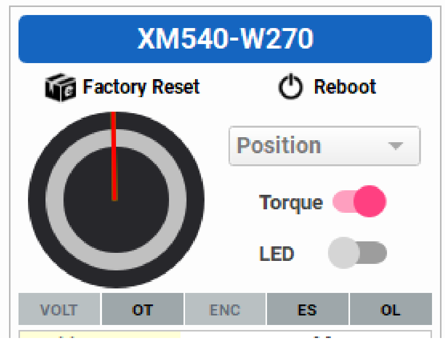
	Rotator	Shoulder	Tri	Forearm	Hip	Central Hip	Leg Rotators	Quads	Knees	Wheels
Right	1	2	3	4	25	26	13, 15	17, *	19, 20	21, 22
Left	5	6	7	8	27		12, 14	10, *	40, 41	23, 24

*Daisy-chained motors not picked up due to internal connection issues.



Locking Motors using Dynamixel Wizard

- After scanning motors, it is recommended to lock the hips and leg rotators to ensure stability when operating HHR.
- Locking the motors can be done by following the steps below:
    - Select the motor to be locked.
    - Toggle the Torque on
        - 

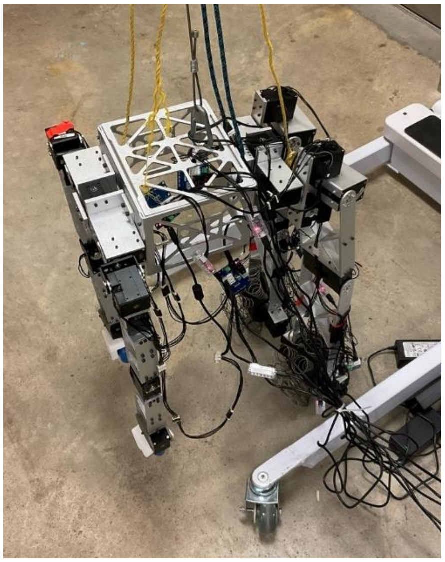
à


Using Python Code for Rolling Motion

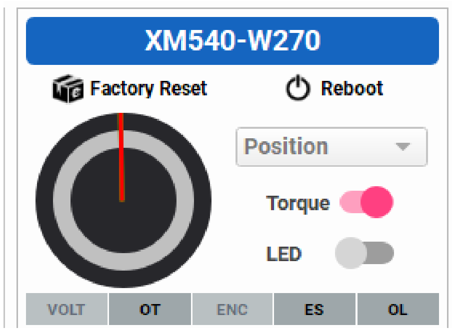
- Open the Python script "climb_stairs_with_legs" provided to initialize positions for all the motors, after locking the hips and leg rotators. After running the code, if there are no connection issues, the terminal should output this:

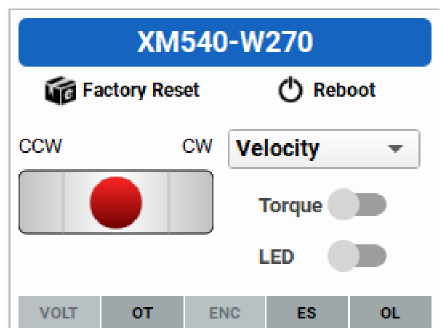
```
Succeeded to open the port
Succeeded to open the port
Succeeded to open the port
Succeeded to change the baudrate
Succeeded to change the baudrate
Succeeded to change the baudrate
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
```

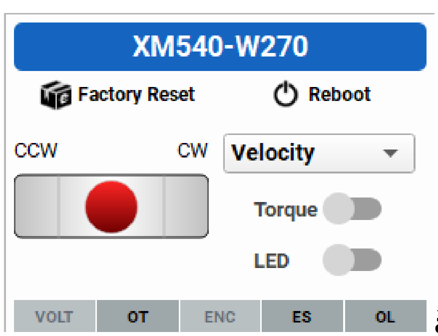
    - Once connected press the Y button on the controller to initialize HHR's climbing position.
        - HHR should be in this position now:

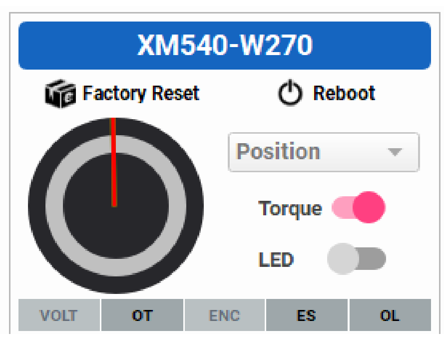


à

- Once HHR is in this position, kill the current terminal and run the Python script "wheel_roll" to move HHR forward and back using the left joystick.
  - If no motion occurs when moving the joystick up and down, ensure the wheels are in "velocity" mode and not locked in the Dynamixel Wizard.
    - 

à


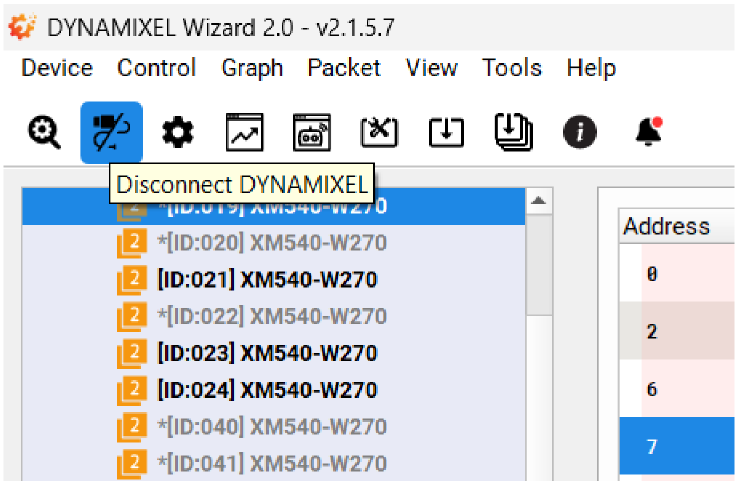
- When done with the rolling motion, go back to the Dynamixel Wizard and switch back to position mode and toggle the Torque on.
    - 

à


- If an error occurs and the following is displayed on the terminal then the motors are still connected to the Dynamixel Wizard and must be disconnected first for the python code to work properly.

```
Succeeded to open the port
Traceback (most recent call last):
  File "c:\Users\orlan\Box\CLASSES\Fourth Year\Spring 2024\MAE 4620\SoftwarePackages\Final Code - HHR\climb_stairs_with_legs.py", line 150, in <module>
    if portHandler3.openPort():
       ^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\dynamixel_sdk-3.7.51-py3.12.egg\dynamixel_sdk\port_handler.py", line 44, in openPort
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\dynamixel_sdk-3.7.51-py3.12.egg\dynamixel_sdk\port_handler.py", line 68, in setBaudRate
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\dynamixel_sdk-3.7.51-py3.12.egg\dynamixel_sdk\port_handler.py", line 114, in setupPort
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\pyserial-3.5-py3.12.egg\serial\serialwin32.py", line 33, in __init__
    super(Serial, self).__init__(*args, **kwargs)
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\pyserial-3.5-py3.12.egg\serial\serialutil.py", line 244, in __init__
    self.open()
  File "C:\Users\orlan\AppData\Local\Programs\Python\Python312\Lib\site-packages\pyserial-3.5-py3.12.egg\serial\serialwin32.py", line 64, in open
    raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM3': PermissionError(13, 'Access is denied.', None, 5)
PS C:\Users\orlan>
```

- Go back to the Dynamixel Wizard and click on the disconnect button.

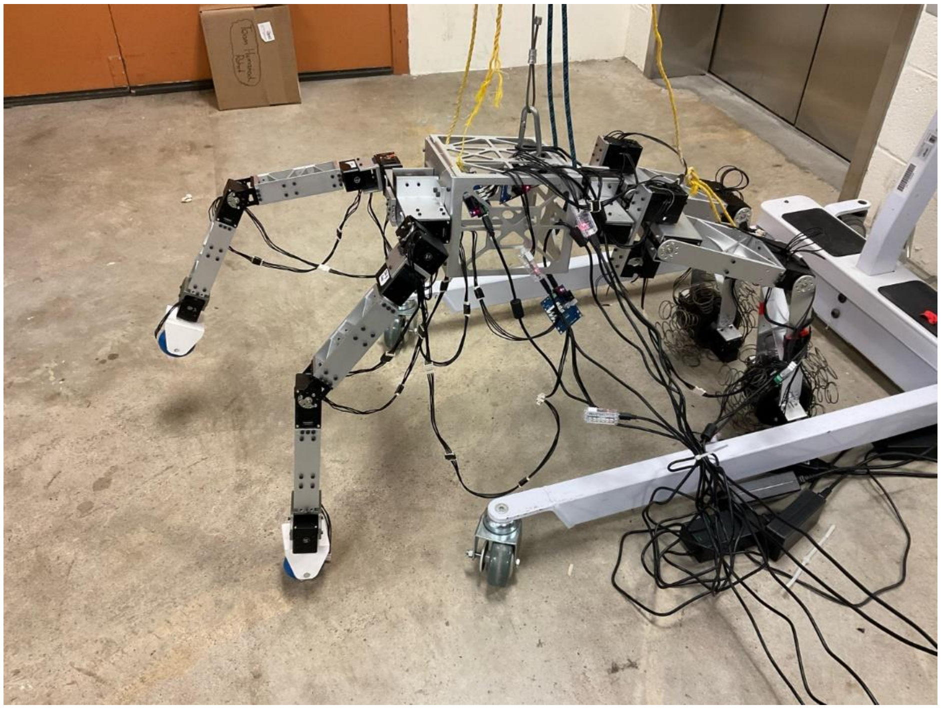


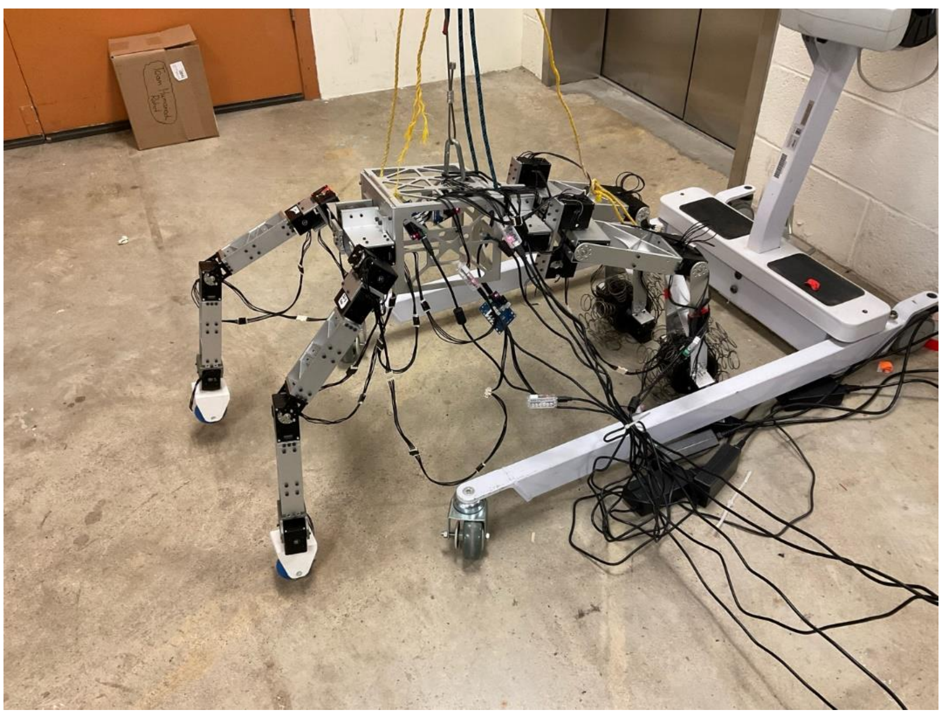
Using Python Code for Climbing Motion

- Open the Python script "climb_stairs_no_legs" provided to initialize positions for the arms, after locking the hips and leg rotators. After running the code, if there are no connection issues, the terminal should output this:

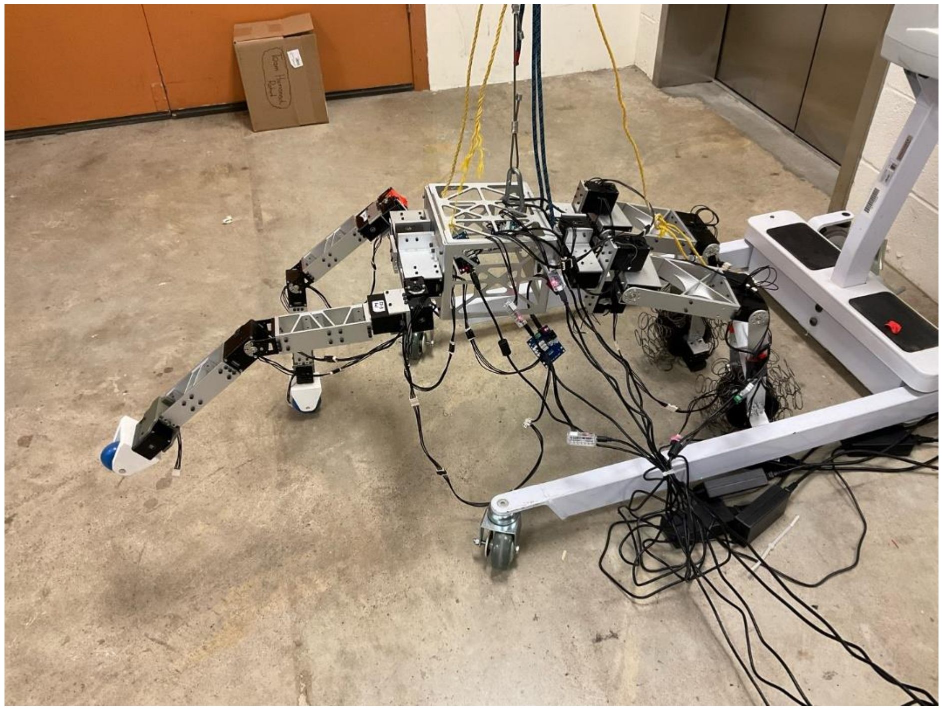
```
Succeeded to open the port
Succeeded to change the baudrate
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
```

    - If any error occurs consult the previous troubleshooting steps.
- Use the following controls to operate HHR to climb the ladder.
    - **Y Button**

- 
  - Right shoulder up

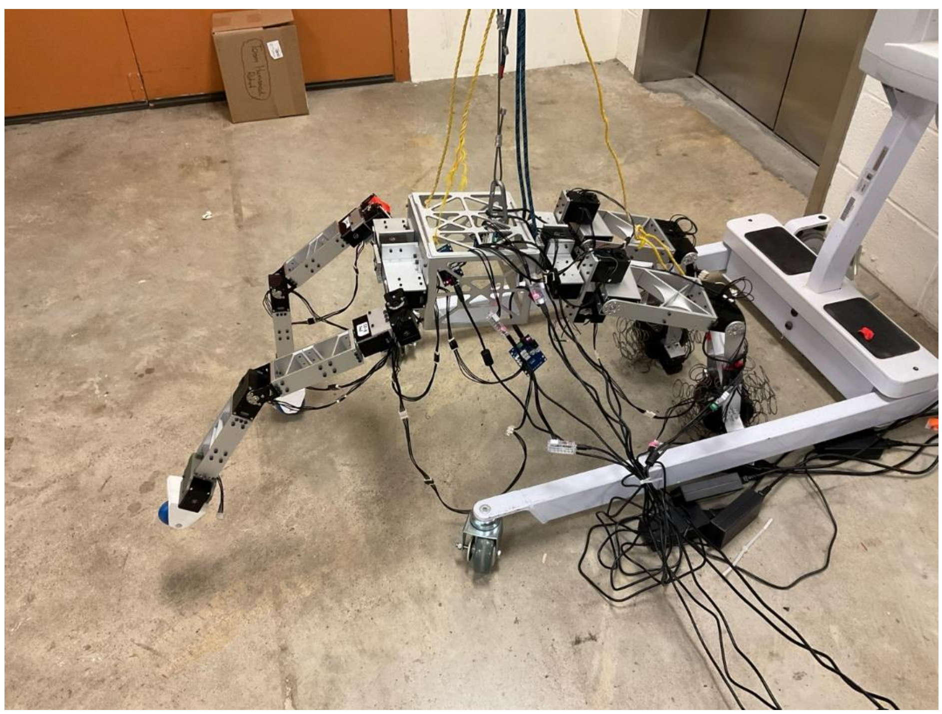
- **A Button**
  - 
    - Right Shoulder Down

- **X Button**



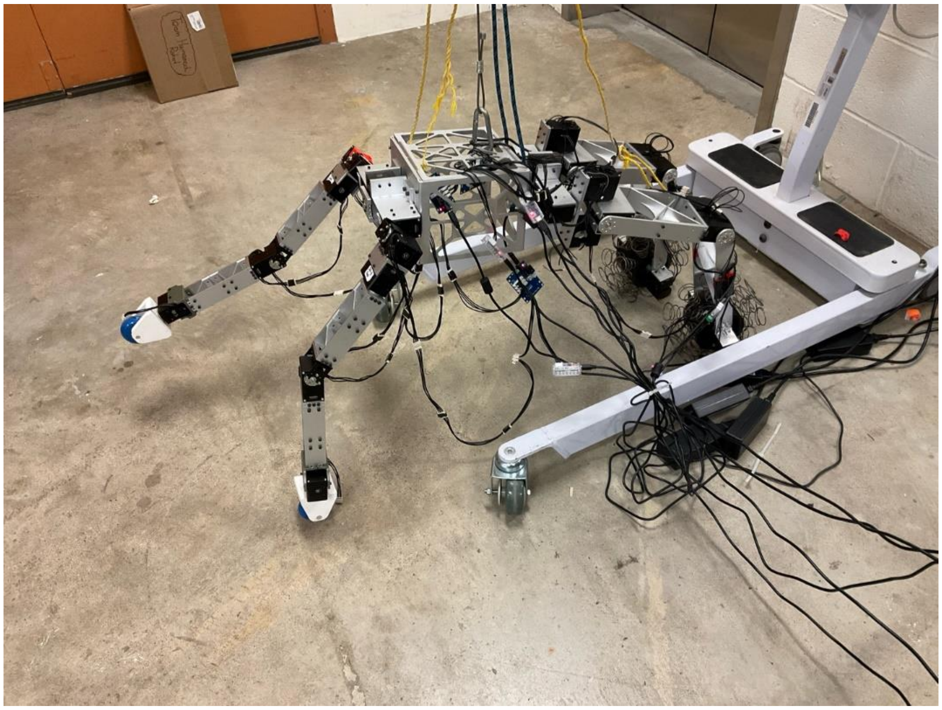
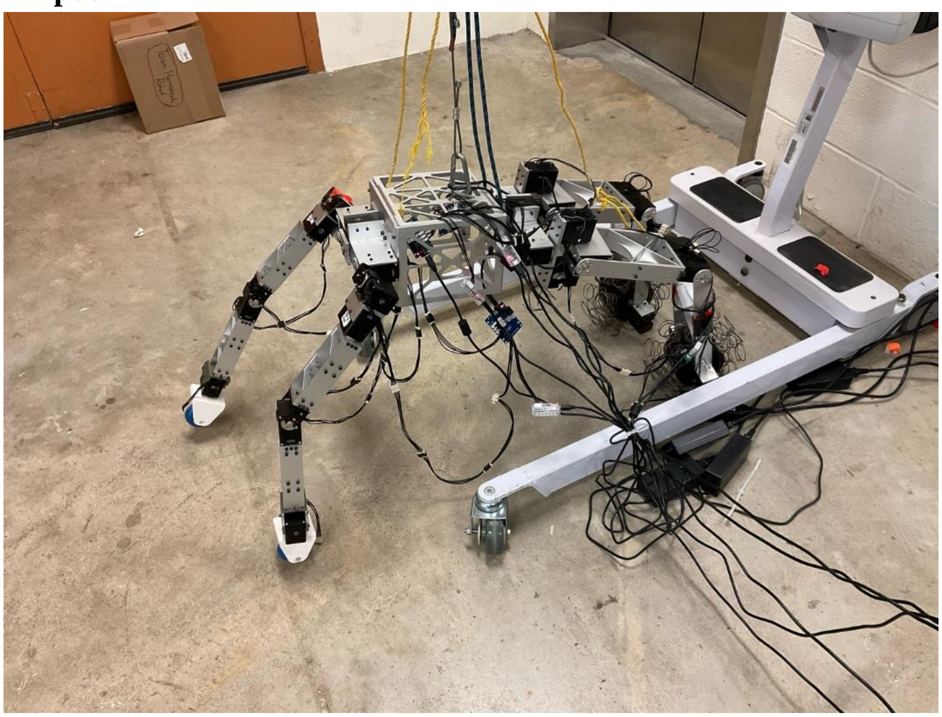
- Left Shoulder Up

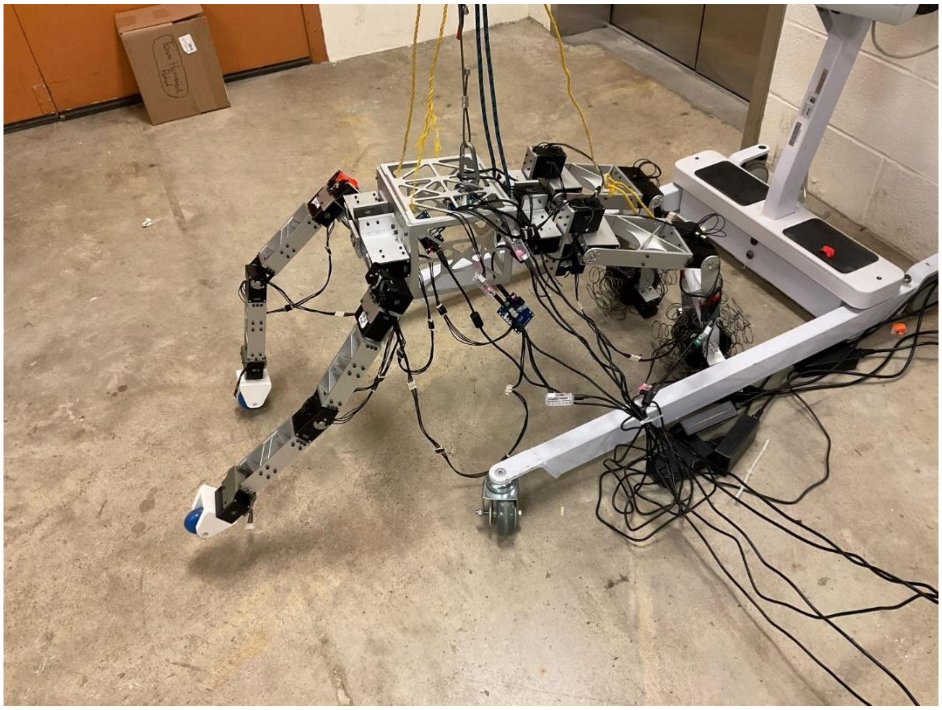
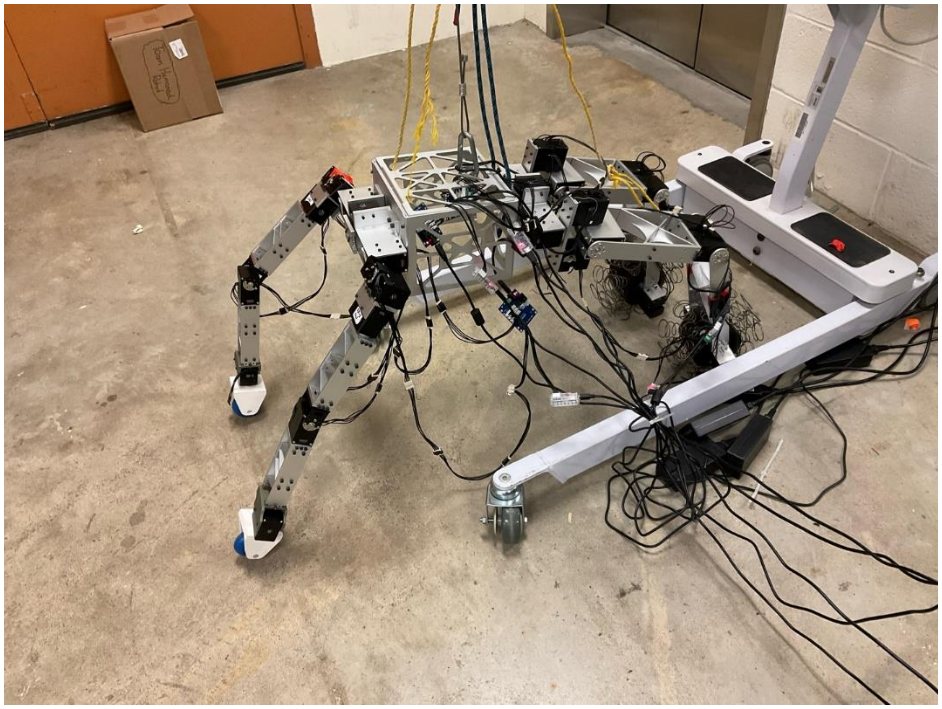
- **B Button**



- Left Shoulder Down

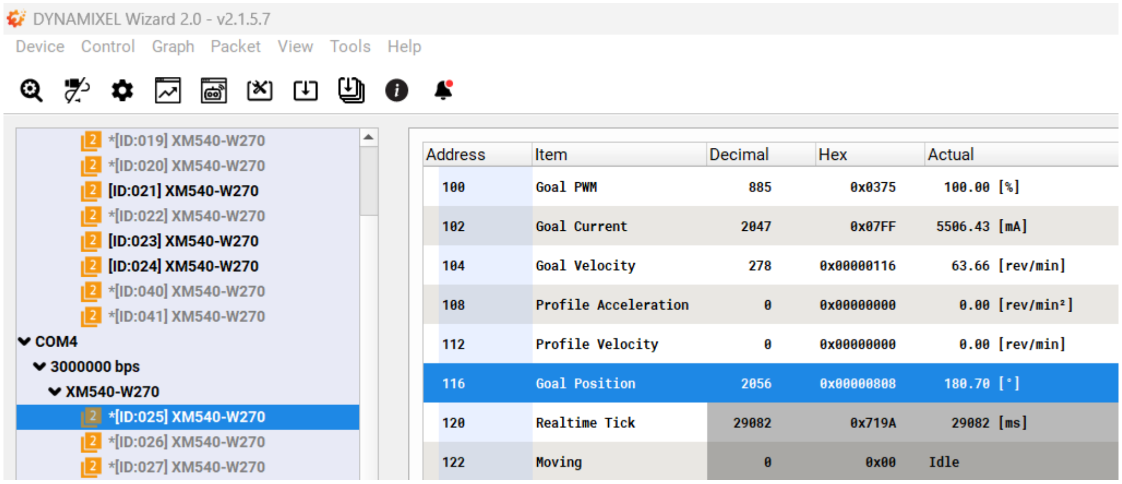
- **Up – D-pad**

- 
  - Right Elbow Flexes Out
- **Down – D-pad**
  - 
    - Right Elbow Flexes Back
- **Left – D-pad**

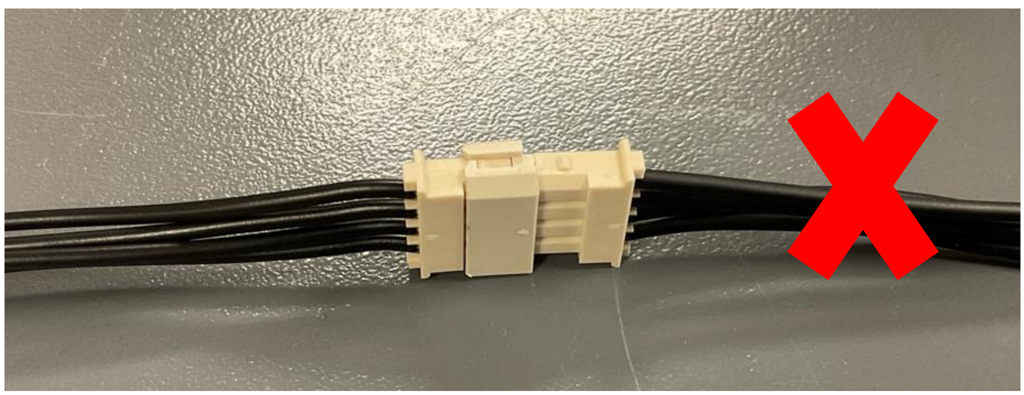
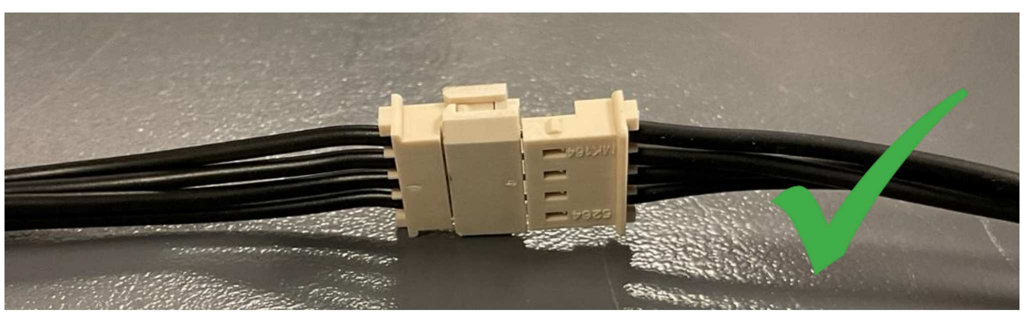
- 
  - Left Elbow Flexes Out
- **Right – D-pad**
  - 
    - Left Elbow Flexes Back

Using Dynamixel Wizard to Determine Current Motor Positions

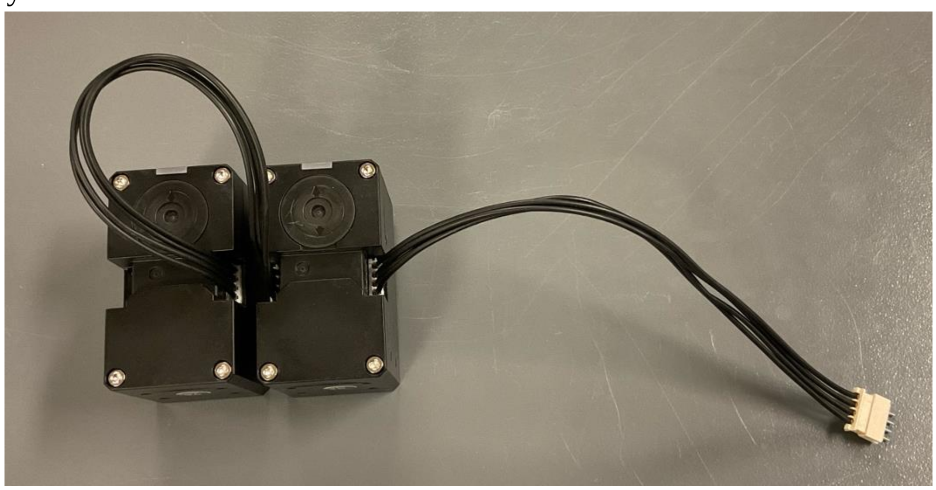
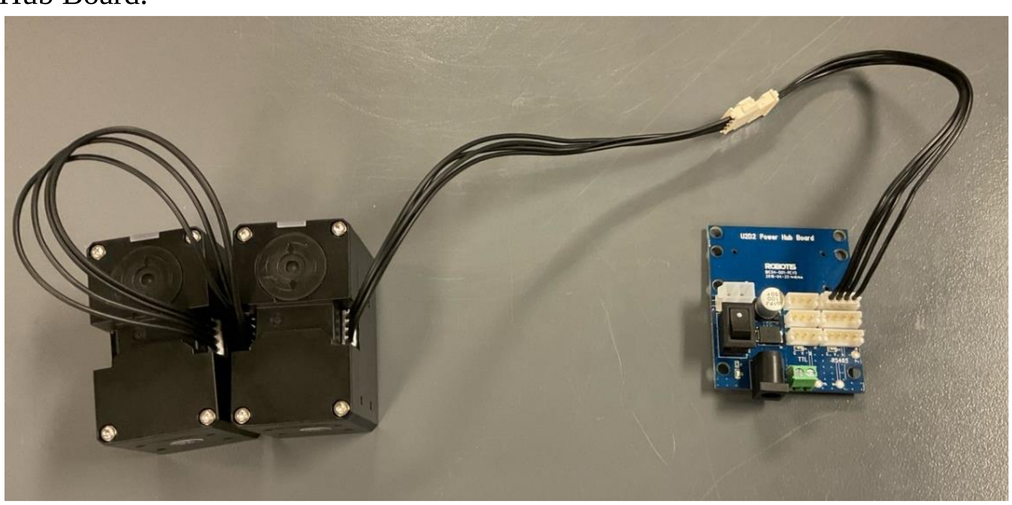
- In Dynamixel Wizard, select a motor and find the 'Goal Position' value in the control table.

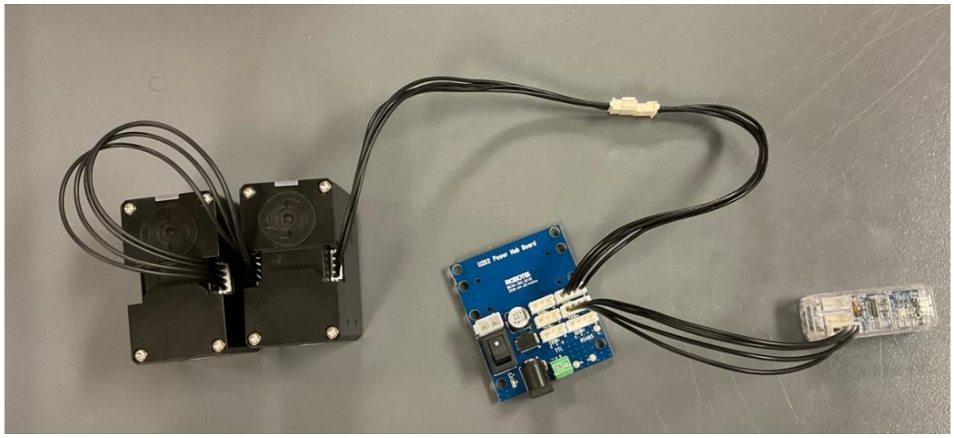
- Record this value for each motor as a reference for desired starting positions.
    - 

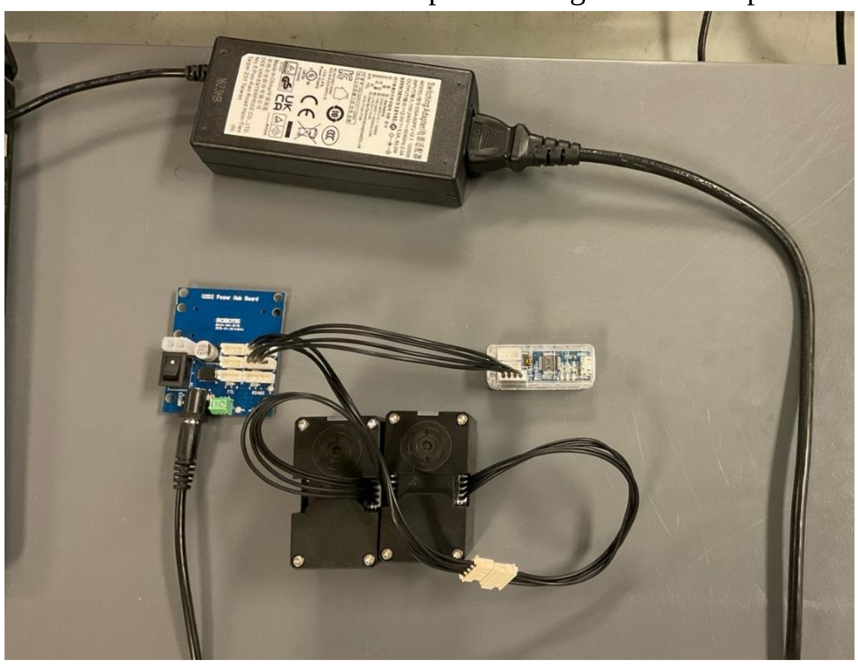
Maintenance

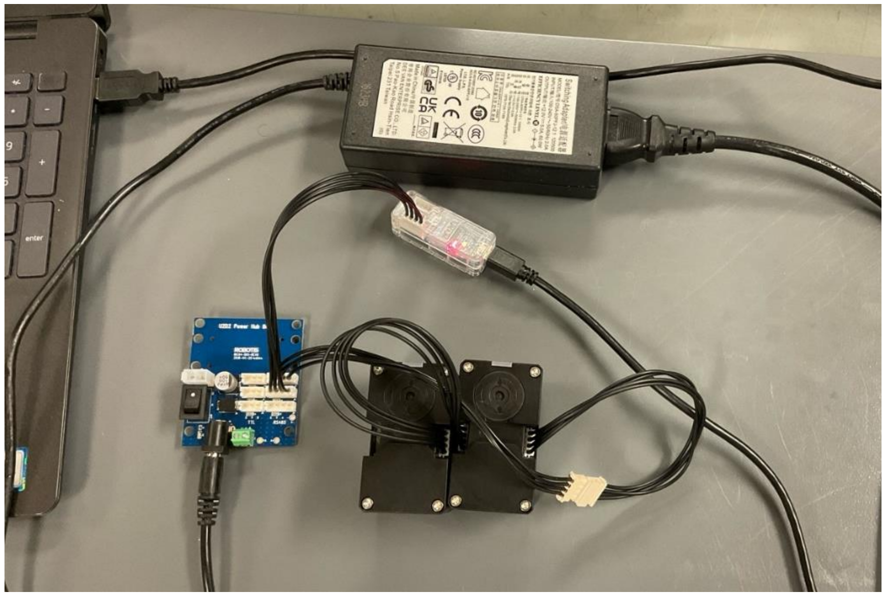
- Regularly check motor connections and cables for wear and tear. Periodically update the Dynamixel Wizard and Python scripts to accommodate any changes in hardware or software.
- Ensure the extension cables are in the proper orientation, if not that will cause motors to not be recognized by the U2D2 boards.
    - 
    - 

Assembly/Disassembly

- **Daisy-Chaining 2 Motors**
    - Use a Robot Cable-X4P 180 mm to connect 2 motors to each other.
        -  à 
- **Connecting to the U2D2**
    - Use a Robot Cable-X4P (Convertible) 180 mm and a 4-pin connector to connect the daisy-chained motors to a hub or extend to the U2D2 Power Hub board.
        - 
    - Use another Robot Cable-X4P (Convertible) 180 mm to connect to the U2D2 Power Hub Board.
        - 
    - Use a Robot Cable-X4P 180 mm to connect to the U2D2 USB Micro-B Board.

- 

- Connect the U2D2 Power Hub Board to power using the AC Adapter
    - 
- Connect the U2D2 Micro-B Board to the computer machine using a micro-USB cable.

- 

- **Disassembly**
  - Follow the Assembly steps in reverse to completely disassemble the system.

Conclusions and Future Work

Our project to develop the Hybrid Humanoid Robot (HHR) for the U.S. Navy has made noteworthy progress. We have focused on designing a robot that can switch between walking and wheeling modes. Those modes adapt to different environments such as ladders, watertight doors, or just a flat surface. Our design process has been through a lot of steps where we were able to determine the best solution for each part of the robot to best adhere to the robot's functionality. We have chosen components carefully while balancing efficiency and innovation. Currently, robots can move around on all fours, and we have made it so that the robot can step onto the ladder while grabbing onto the railing. However, there are still challenges to overcome, like improving limb coordination and addressing communication errors. If we can overcome these challenges, HHR will handle a wide variety of tasks by blending bipedal and wheeled mobility. There is still a long way to go before HHR can function in a complex environment, but if progress is made at this pace, we might achieve this goal in due course of time.

In our capstone research of humanoid robot, it is becoming clear that there are several key areas that still need further investigation to improve the stability, performance, and flexibility of the robots. One crucial focus for the future is integrating the robots center of mass (CoM) and multi body dynamics modeling into our control algorithms. Analysis of the CoM still have to be done. There needs to be Implementation of CoM equations within ROS packages to allow for dynamic adjustments in posture based on this information. This integration will enable adaptable control over the robot's movements enhancing stability and performance in tasks like walking and handling objects within Naval ships.

Aside from analyzing the CoM and modeling body dynamics, incorporating various sensors is vital to enhance the robots awareness of its surroundings, interaction capabilities and maneuvering abilities. Depth cameras like Intel RealSense or Microsoft Kinect will be utilized for accurate distance measurements, obstacle detection and navigation purposes. Inertial Measurement Units (IMUs) will also be integrated to provide data on the robots orientation, acceleration, and angular velocity to assist in balance control and planning movements.

Furthermore, we will incorporate force sensors, torque sensors, and tactile sensors as laser range finders to enhance object manipulation capabilities such as grasping objects securely and mapping out the environment effectively in Navy vessels. For the future we envision cameras to be implemented in HHR that will identify objects, track movement and navigate visually. By combining these sensors with body dynamics modeling and Center of Mass (CoM) analysis in ROS packages as well as creating sophisticated adaptive control algorithms our humanoid robot will be able to move more effectively engage better with their surroundings and handle complex tasks with precision and efficiency.

Hopefully, our upcoming work will concentrate on integrating CoM analysis multi body dynamics modeling and sensor data into ROS packages to enable accurate and adaptable control of humanoid robots. By developing flexible control algorithms and feedback mechanisms we aim to increase the robot's agility, interaction capabilities and performance in challenging scenarios in Navy vessels.

References

Aircraft carriers. HII. (2023, September 18). https://hii.com/what-we-do/capabilities/aircraft-carriers/

Altagar, G. (2023, March 12). *Robots - 2 legs or wheels. Which is better?* Robotics. https://www.unlimited-robotics.com/post/robots-2-legs-or-wheels-which-is-better#:~:text=Wheeled%20robots%20are%20designed%20to

Berkowitz, B. (2014). Sea power in the robotic age. *ISSUES in Science and Technology*, *30*(2), 33-40.

Coito, J. (2021). Maritime autonomous surface ships: New possibilities—and challenges—in ocean law and policy. *International Law Studies*, *97*(1), 19.

Emad, G. R., Enshaei, H., & Ghosh, S. (2022). Identifying seafarer training needs for operating future autonomous ships: a systematic literature review. *Australian Journal of Maritime & Ocean Affairs*, *14*(2), 114-135.

Hashemi-Petroodi, S. E., Thevenin, S., Kovalev, S., & Dolgui, A. (2020). Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. *Annual Reviews in Control*, *49*, 264-276.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1620-1626). IEEE.

Khurshid, J., & Bing-Rong, H. (2004, December). Military robots-a glimpse from today and tomorrow. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.* (Vol. 1, pp. 771-777). IEEE.

Kim, S., Cho, J., Kim, Y., & Oh, J. (2019). Development of a bipedal humanoid robot with wheel feet for stable and efficient locomotion. *Robotics and Autonomous Systems, 118,* 75-87.

Lehman, J. (2018). From ships to robots: The social relations of sensing the world ocean. *Social Studies of Science*, *48*(1), 57-79.

Love, L. J., Jansen, J. F., & Pin, F. G. (2004, April). On the modeling of robots operating on ships. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004* (Vol. 3, pp. 2436-2443). IEEE.

O'Heir, J. (2021, June 21). *How Robots Will Help the Navy Increase Workforce Diversity—ASME*. The American Society of Mechanical Engineers. https://www.asme.org/topics-resources/content/navy-set-sail-for-new-robotic-systems

Pratt, J., Chew, C.-M., Torres, L., Dilworth, P., & Pratt, G. (2007). Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research, 26(2),* 171-188.

Submarines. HII. (2023b, September 18). https://hii.com/what-we-do/capabilities/submarines/

Sparrow, R., & Lucas, G. (2016). When robots rule the waves?. *Naval War College Review, 69*(4), 49-78.

United States Government Accountability Office. (2022). *Uncrewed Maritime Systems: Navy Should Improve Its Approach to Maximize Early Investments*. GAO. https://www.gao.gov/assets/gao-22-104567.pdf

Verma, P. (2022, April 15). The military wants 'robot ships' to replace sailors in battle. *Washington Post*. https://www.washingtonpost.com/technology/2022/04/14/navy-robot-ships/

Wright, R. G. (2020). *Unmanned and autonomous ships: an overview of mass*. Routledge.

Yang, J., & Spenko, M. (2013). Design of a novel wheel-leg mobile robot for efficient and adaptive locomotion. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4495-4500). IEEE.

Zereik, E., Bibuli, M., Mišković, N., Ridao, P., & Pascoal, A. (2018). Challenges and future trends in marine robotics. *Annual Reviews in Control, 46,* 350–368. https://doi.org/10.1016/j.arcontrol.2018.10.002

ZHU, J., LIU, R., GE, Y., WANG, Z., & HUANG, K. (2017). Pure loss of stability calculation of naval ships in regular waves. *Chinese Journal of Ship Research, 12*(3).