

A SWARM OF COLLABORATIVE MODULAR ROBOTS TO OVERCOME OBSTACLES IN PIPES: PRE-LINKING CONTROL ALGORITHM

MECH5080M Team Project - Individual Report
***A Swarm of Collaborative Modular Robots
to Overcome Obstacles in Pipes: Pre-
Linking Control Algorithm***
Author: Fintan Lyons
Supervisor: Andy Barber
Industrial Mentor: N/A
Examiner: Todd Stewart
Date: 01/05/2025



MECH5080M TEAM PROJECT 45 credits

TITLE OF PROJECT

A SWARM OF COLLABORATIVE MODULAR ROBOTS TO OVERCOME OBSTACLES IN PIPES: PRE-LINKING CONTROL ALGORITHM

PRESENTED BY

Fintan Lyons

OBJECTIVES OF PROJECT

1. Conduct a literature review, investigating existing in-pipe robot localisation, navigation, linking mechanisms, control strategies and simulation practices
2. Develop a pre-linkage control algorithm concept and a suitable sensor fusion strategy
 - a. Outline a control algorithm to link a scalable number of modular robots.
 - b. Identify potential sensor configurations
 - c. Perform experimental testing on individual physical sensors to validate performance and select the most appropriate sensors
3. Develop and validate a pipe simulation environment to test the control algorithm
 - a. Create scalable modular robot models with verified sensors and actuator plugins in a pipe world simulation
 - b. Implement and test the control algorithm within the simulation to evaluate its functionality, scalability and robustness
4. Develop and validate a physical prototype system
 - a. Build physical modular robots with selected sensors and actuators, controllable over Wi-Fi using ROS2 communication
 - b. Test and validate the control algorithm in a physical pipe environment

IF THE PROJECT IS INDUSTRIALLY LINKED TICK THIS BOX AND PROVIDE DETAILS BELOW

☐

THIS PROJECT REPORT PRESENTS OUR OWN WORK AND DOES NOT CONTAIN ANY UNACKNOWLEDGED WORK FROM ANY OTHER SOURCES.

SIGNED



DATE 01/05/2025

Contents

Abstract.....	v
Chapter 1: Introduction	1
1.1. Introduction	1
1.2. Individual Project Aims.....	2
1.3. Individual Project Objectives	2
Chapter 2: Literature Review	3
2.1. Pipe Robot Locomotion and Obstacle Challenges	3
2.2. Robot Linking Strategies	3
2.3. Robot Navigation and Obstacle Detection Sensors	5
2.4. Autonomous Robot Control	6
2.5. Robot Simulation	7
Chapter 3: Control Algorithm	8
Chapter 4: Sensor Selection	9
4.1. Centring Robots	9
4.2. Detecting Robots	9
4.3. Aligning and Connecting Robots	10
4.4. Validation of Physical Sensors	11
4.4.1. Range and Alignment Sensor Tests	11
4.4.2. IMU Sensor Tests.....	13
4.5. Sensor Selection Conclusion.....	13
Chapter 5: Simulation Setup, Testing and Results	14
5.1. Simulation Setup.....	14
5.2. Verification of Simulation Actuator Functionality.....	14
5.3. Verification and Testing of Simulation Sensors	15
5.3.1 IMU and TOF	15
5.3.2 IR Obstacle Detector	15
5.4. Control Algorithm Testing	16
5.4.1 Centring Robots.....	16
5.4.2 Detecting, Aligning and Connecting Robots	17
5.5. Control Algorithm Simulation Results and Discussion.....	17
Chapter 6: Physical Setup, Testing and Results	18
6.1. Physical Setup.....	18
6.2. Verification of Physical Setup, Sensors and Actuators	18

6.3.	Physical Testing	19
6.4.	Physical Results and Discussion.....	19
Chapter 7: Conclusion and Future Work		20
7.1.	Conclusion	20
7.2.	Future Work.....	20

Abstract

This paper presents the development of a pre-linkage control algorithm to connect a swarm of modular robots within a pipe environment. A finite state machine (FSM) approach was designed to manage robot behaviours across ten discrete states, supported by a conceptual framework of sensor fusion for centring, detecting, aligning, and connecting robots. Physical testing of candidate sensors informed the selection of a suitable sensor fusion layout, balancing performance, cost, and system constraints. A scalable simulation environment was created to develop and validate the control algorithm, with successful linkage of up to five robots demonstrating both functionality and scalability. Three physical robots were then built and tested under ROS2 control via Wi-Fi. Although the control algorithm proved successful in simulation, physical trials highlighted critical real-world limitations, particularly response delays and traction inconsistencies, which prevented successful linkage. The project establishes a foundation for future improvements to bridge the gap between simulated and physical robot performance.

Chapter 1: Introduction

1.1. Introduction

Pipelines are critical infrastructure for transporting essential resources such as oil, gas and water. Unfortunately, as pipelines age, they suffer from damage, corrosion, deformation and leakage, reducing network efficiency and increasing maintenance needs. Historically, inspections have been carried out by human workers [1], but this approach is limited to larger diameter pipes, exposes personal to hazardous environments, and is time-consuming and costly. In the UK alone, there are over 625,000 km of sewage pipelines [2], with an average diameter of just 15 cm [3] – far too small for human access.

To overcome these challenges, tethered CCTV rovers are commonly used. While effective, cable management issues and the high costs of deployment and footage analysis limit their efficiency. Autonomous robot inspection offers a promising alternative, with the potential to improve speed, safety, and coverage [4]. Although various specialised pipe inspection robots have been developed in laboratory settings ([4], [5], [6], [7]), none have successfully transitioned to commercial use. Current commercial solutions remain largely tethered and manually operated.

Operating underground presents major challenges for robotic systems. GPS signals are unavailable, and communication links are unreliable, making external localisation and control infeasible. Robots must instead rely on onboard sensors to localise, navigate, and make autonomous decisions in highly constrained, unpredictable environments. Autonomy not only reduces the need for constant human supervision but also enables faster, more efficient inspections across complex pipe networks where human access is impossible.

A further key challenge is adapting to the highly variable conditions within real-world pipes, such as sudden changes in pipe diameter, debris, steps, and bends. In this context, modular robots offer significant advantages. Drawing on developments in the field of self-reconfigurable robotics, modular systems can physically link together to share traction, overcome obstacles, and adapt dynamically to different environments. Unlike single rigid robots, modular swarms can increase their collective resilience, enabling access to areas that would otherwise be unreachable.

In collaboration with a larger team, this project seeks to address these challenges by developing a swarm of modular, collaborative, physically linking robots capable of overcoming obstacles inside pipes. Specifically, this individual project focuses on designing and testing the pre-linkage control algorithm — the stage where initially disconnected modular robots autonomously detect, approach, and connect to each other in preparation for collective movement.

The structure of this report is as follows: Chapter 2 presents a literature review of existing approaches to pipe robot locomotion, linking, navigation, control, and simulation, Chapter 3 introduces the proposed control algorithm, Chapter 4 details the sensor selection process and physical sensor testing, Chapter 5 presents the development and testing of the simulation environment, Chapter 6 describes the physical system implementation and experimental results and Chapter 7 provides conclusions and recommendations for future work.

1.2. Individual Project Aims

The aim of this project is to develop, test and integrate a pre-linkage swarm control algorithm to connect a swarm of modular robots.

1.3. Individual Project Objectives

1. Conduct a literature review, investigating existing in-pipe robot localisation, navigation, linking mechanisms, control strategies and simulation practices
2. Develop a pre-linkage control algorithm concept and a suitable sensor fusion strategy
 - a. Outline a control algorithm to link a scalable number of modular robots.
 - b. Identify potential sensor configurations
 - c. Perform experimental testing on individual physical sensors to validate performance and select the most appropriate sensors
3. Develop and validate a pipe simulation environment to test the control algorithm
 - a. Create scalable modular robot models with verified sensors and actuator plugins in a pipe world simulation
 - b. Implement and test the control algorithm within the simulation to evaluate its functionality, scalability and robustness
4. Develop and validate a physical prototype system
 - a. Build physical modular robots with selected sensors and actuators, controllable over Wi-Fi using ROS2 communication
 - b. Test and validate the control algorithm in a physical pipe environment

Chapter 2: Literature Review

2.1. Pipe Robot Locomotion and Obstacle Challenges

The locomotion mechanisms employed by in-pipe robots vary widely, from wheeled and tracked designs to screw-drive and snake-inspired systems. Each approach is developed with the aim of traversing the confined and often irregular geometries of pipelines reliably. Wheeled robots are noted for their simplicity and speed; however, they often struggle with rough terrain, low traction causing wheel slippage and sudden changes in ground height [7] [6]. In contrast, tracked, screw-drive, and snake-inspired systems benefit from increased traction and stability but tend to require more complex mechanisms and are frequently tailored to a single, unchanging pipe diameter [8]. Although extensive simulation and controlled experimentation have optimised individual locomotion strategies, very few systems have addressed the cumulative impact of several naturally occurring obstacles within operational pipelines – such as sequences of obstacles such as steps and bridges.

The Pipebots project, was created to develop robotic platforms for autonomous inspection of underground pipes. The project envisioned the collaboration of multiple different size robot platforms for inspection of different sized sewer tributaries. Initially a small autonomous inspection robot named “Joey” was created, which successfully mapped and explored 100 mm diameter pipes within a laboratory environment. Despite, the project’s success, autonomously mapping unknown sections of pipes, Joey failed to address the challenge of overcoming obstacles within pipes, which limited its mapping capabilities [4].

Given the limitations of standalone in-pipe robots, some recent research has turned to modular permanently linked robotic systems which form adaptable, resilient units. These systems aim to share load and distribute traction forces. For example, modular wall-press robots such as those demonstrated by Zhang et al. (2019) [5] and Jang et al. (2022) [8] have shown that in a laboratory setting, interconnected robots can adapt to changes in pipe orientation and effectively transverse sections of pipes that single robots cannot.

2.2. Robot Linking Strategies

Outside of pipes, several self-reconfigurable robotic systems have been created with different alignment strategies, some of these are listed in Table 1:

Table 1: Self-reconfigurable Robotic Docking Strategies

System	DOF	Author	Year	Robot linking Strategy	Docking Strategy
Omni-Pi-tent	3	Peck, Timmis, Tyrell (University of York) [9]	2019	Docking Hook	Physical and IR Sensors
Symbion	3	EU Projects Symbion and Replicator [10]	2013	Cone Bolt	Physical and IR Sensors
SMORES	4	Davey, Kwok, Yim (UNSW, Upenn) [11]	2023	EP Magnets	EP Magnets
M-TRAN III	3	Kurokawa et al. (AIST) [12]	2008	Docking Hook	Physical
Sambot	3	Wei et al., (Beihang University) [13]	2010	Docking Hook	IR Sensors
N/A	2	Delrobaei et al. (University of Western Ontario) [14]	2011	Universal Joint	Camera and LED

A review of several self-reconfigurable robots found that magnets, whilst enabling large alignment tolerances, are energy intensive and require significant cooling periods. A more widely used docking strategy was involved mechanical docking using physical linkages, such as docking hooks. These systems enabled faster and more energy efficient docking but required more precise alignment. For instance, the M-TRAN (III), switched from a magnetic connection to a mechanical connection, reducing the connection time from over a minute to just 5 seconds [15].

To overcome the challenge of tighter mechanical tolerances, various sensors fusion techniques have been employed to aid alignment. These include combinations of cameras with LED markers, to infrared (IR) sensors and inertial measurement units (IMUs) [13], [11], [14]. For example, Sambot used IR sensors to monitor the relative position of two robots during docking, providing real-time guidance to the actuators. Its mechanical hook-and-groove design allowed for a small degree of misalignment [13]. Delrobaei [14] used a vision-based docking system, whereby an upward-facing camera and coloured LED reflections on a ceiling-mounted mirror to estimate the pose of the target robot.

The literature demonstrates a wide range of sensor fusion approaches to support alignment. IR sensors combined with mechanically guided alignment features are among the most common, offering a good balance between accuracy and low computational demand. Ultimately, there is no universally optimal alignment method; the choice depends on the specific requirements and constraints of the robot's design and intended use.

2.3. Robot Navigation and Obstacle Detection Sensors

Autonomous operation is essential for pipeline robots, as real-time communication from outside buried pipe networks is often unfeasible. To achieve navigation and inspection in these environments, a range of sensor systems have been integrated into different robotic platforms, to enable autonomous navigation and inspection of pipe networks. Table 2 summarises how sensors employed in recent research and their associated use cases:

Table 2: Pipe Robot's Sensor Utilisation

Sensor	Author	Use case
IR Sensor (TOF)	Nguyen (2022) [4]	Obstacle and Corner Detection
Ultrasonic Sensors	Zhao (2018) [16]	Obstacle and Corner Detection
LiDAR Sensor	Zeng (2019) [17]	Obstacle Detection
	Masuta (2013) [18]	Pipe Shape and Corner Detection
Encoders	Brown (2018) [19]	Corner Detection
IMU	Nguyen (2022) [4]	Robot Centring
	Murtra (2013) [20]	Localisation (with cable encoder)
	Song (2016) [21]	Localisation (with wheel encoder)
Sonde	YSI xylem [22]	Localisation
Camera	Edwards (2023) [23]	Localisation by feature recognition
	Oyama (2019) [24]	Localisation by feature recognition

Across the literature, sensor fusion was a common strategy to increase robustness in navigation and localisation. Nguyen's robot "Joey" exemplifies this by integrating Time of Flight (TOF) sensors, an IMU, encoders, and a camera. In this design, the IMU facilitated robot centring and orientation tracking, while the TOF sensors and encoders were used for obstacle detection and localisation. The camera served primarily for visual inspection and offline analysis.

Obstacle and corner detection is typically approached using distance-based sensing. Both Nguyen [4] and Zhao [16] used low-cost IR and ultrasonic sensors, which are effective in close-range environments but susceptible to noise and surface reflections. In contrast, Zeng [17] and Masuta [18] employed LiDAR, which offers the ability to detect complex shapes, at the cost of higher power consumption and size constraints.

For localisation, methods vary significantly depending on the robot's scale and mission profile. Cameras have been used effectively for feature-based localisation [23], [24], though they rely heavily clear visual features, which are not always available inside pipes. IMU-encoder fusion, as seen in Murtra [20] and Song [21], provided a more reliable alternative in feature-poor environments. YSI xylem [22] has developed a

commercially available sonde for absolute positioning, which, while accurate, adds external infrastructure requirements.

Overall, the choice of sensors and fusion strategies is highly application specific. While IR and ultrasonic sensors are effective for short-range obstacle detection, LiDAR and vision systems provide richer spatial data. Localisation strategies vary even more widely, with no single method universally preferred. The optimal configuration depends on factors such as pipe size, expected obstructions, power constraints, and the need for real-time feedback.

2.4. Autonomous Robot Control

Autonomous pipe robots rely on embedded control algorithms to make real-time decisions in constrained and often unpredictable environments. A rule-based control algorithm, as implemented by Zhao [16], represents one of the simplest and most used strategies. In such systems, the robot responds directly to sensor thresholds—such as ultrasonic distances—without maintaining an internal state. This method is computationally lightweight and easy to implement, making it well-suited for environments with well-defined triggers and limited resources. However, it lacks adaptability and scalability, particularly in complex scenarios involving decision hierarchies or ambiguous stimuli.

A more structured alternative is the finite state machine (FSM) approach, used by Nguyen [4]. FSMs offer improved modularity by explicitly defining all robot states and transitions, enabling more predictable and traceable behaviour. This structure is particularly useful when a robot must respond differently depending on context—such as navigating pipe junctions or reacting to sensor failures. Despite its advantages, FSMs can become cumbersome as the number of possible states grows, reducing scalability and maintainability in more complex systems.

In contrast, reinforcement learning (RL), as employed by Zeng [17], allows robots to learn optimal behaviours through trial and error, using feedback from their environment. RL is especially promising in environments where uncertainties or nonlinear dynamics make rule-based or FSM control impractical. However, RL typically requires significant training data and computational resources, and its learned policies can be difficult to interpret or guarantee in safety-critical applications like gas pipelines.

Each of these control strategies offers distinct advantages depending on the application context. Rule-based methods excel in simplicity and are ideal for

predictable environments but fall short in flexibility. FSMs provide a balance between structure and adaptability, making them suitable for moderately complex systems with clear operational modes. Reinforcement learning, while powerful and adaptive, introduces challenges in training, interpretability, and computational demand.

2.5. Robot Simulation

Simulation is a powerful tool when developing robots, enabling a virtual representation of the real world, allowing for virtual testing and development, saving money and time developing real-world prototypes and diagnosing problems [25]. In the context of pipe inspection robots, simulation is particularly valuable due to the difficulty and expense of constructing test environments that accurately replicate underground, in service pipeline conditions.

Modern simulation environments such as Gazebo, Webots and Unity provide realistic physics engines and support for sensor emulation, enabling the testing of detailed navigation and control algorithms. For example, Zeng [17]. used reinforcement learning within the simulation environment Gazebo to train agents over thousands of episodes – something that would be physically infeasible due to logistical constraints.

Despite its benefits, robotic simulation does have its limitations. Simulated environments rarely capture the full complexity and unpredictability of real-world conditions, control algorithms that perform well within a simulation may not directly transfer to hardware – a phenomenon known as the “reality gap”. This gap has been highlighted as a key limitation in several robot development projects.

To bridge this gap, research often employs sim-to-real transfer techniques, such as hybrid simulation-hardware testing, to prepare a system for the unmodelled variations experienced in the real-world [26].

2.6. Synthesis and research gap

Currently, no commercial or research individual modular collaborative physically linking pipe robots exist. This project looks to address the lack of research in this area, by developing scalable modular collaborative pipe robots to overcome obstacles. Drawing inspiration from non-pipeline self-reconfiguring robots and pipeline inspection robots.

Chapter 3: Control Algorithm

A Finite State Machine (FSM) control algorithm was chosen due to the limited number of distinct states a pipeline robot encounters during pre-linkage and FSMs support modularity and scalability. The algorithm includes ten defined algorithm states and six robot states. The flow chart in Figure 1 illustrates the transitions between these states:

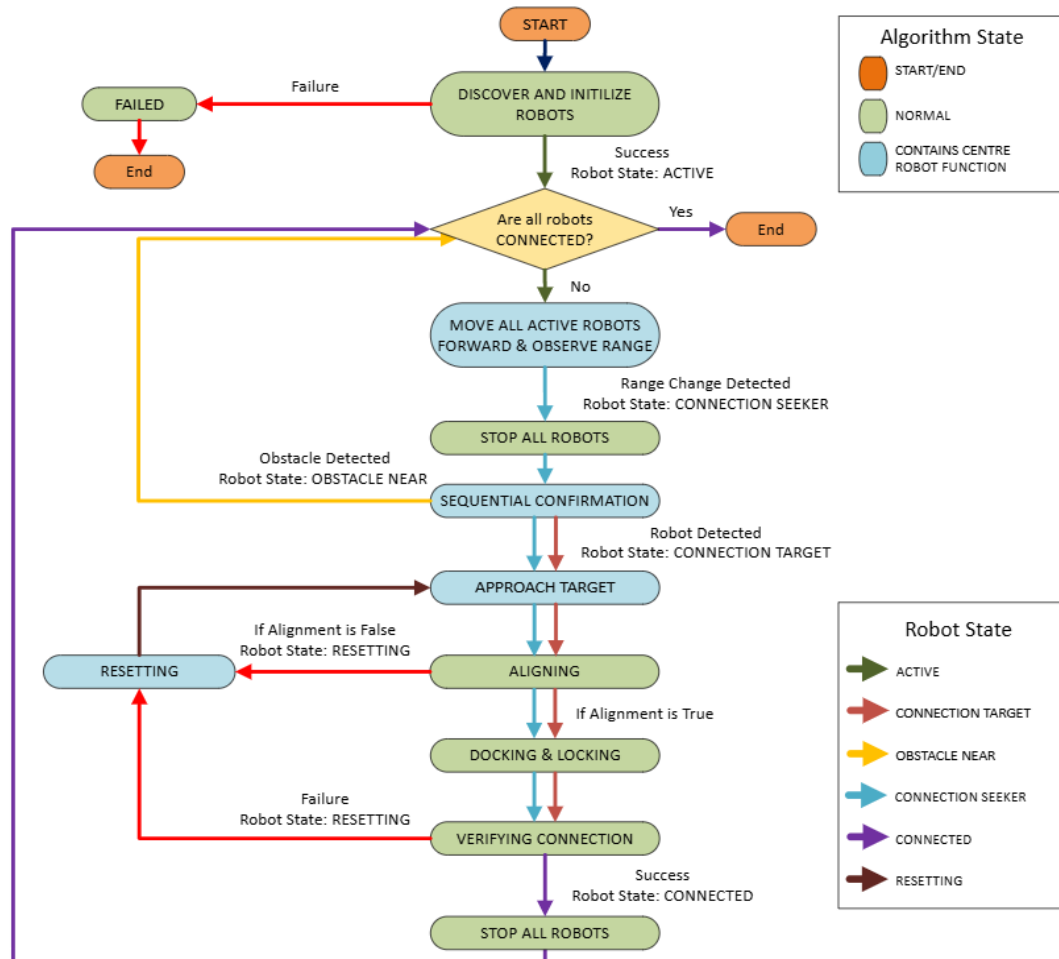


Figure 1: FSM Pre-Linking Control Algorithm

Initially, the system identified the number of robots. All ACTIVE robots then began moving forward, monitoring for changes in range. The robot that detected the range change was designated the CONNECTION SEEKER. Sequential movement was then used to determine whether the seeker has encountered another robot or an obstacle. If another robot was detected, it was assigned the state CONNECTION TARGET. If an obstacle was detected, the seeker robot entered the state OBSTACLE NEAR. If a connection target was identified, the seeker approaches, attempts to align, then proceeds to dock and lock. If the docking was verified as successful, both robots transition to the CONNECTED state. If the connection failed, they reset and tried again. Because the system operates in a closed loop, the robots will continue attempting connections until all robots are connected or a failure occurs.

Chapter 4: Sensor Selection

Four broad functions were needed to enable robot linkage: centring, detecting, aligning and connecting robots. Sensors were selected to achieve these four functions.

4.1. Centring Robots

Side mounted TOF, Ultrasonic and IMU sensors were proposed, to sense the robots offset from the centre of the pipe. Figure 2 shows the proposed arrangement and Figure 3, the logic of these sensors:

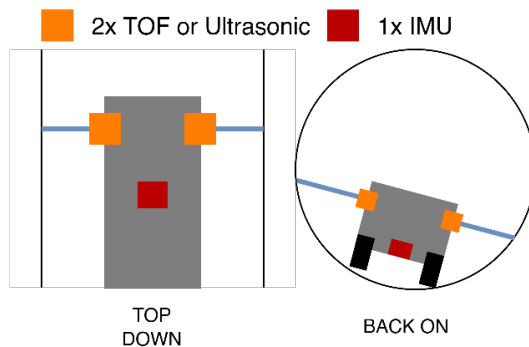


Figure 2: Centring Robot Sensor Options

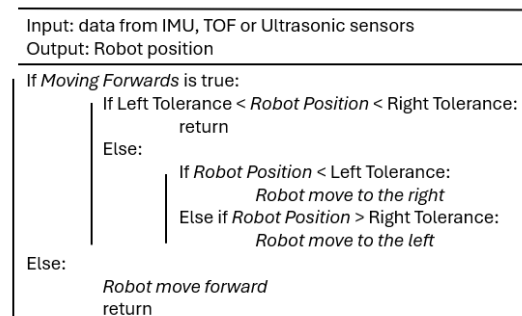


Figure 3: Centring Robots Pseudo Code

The TOF/Ultrasonic sensors can be used to measure the distance between the robot and the adjacent pipe walls, if one wall was found to be closer than another, the control algorithm would know the robot had moved off centre and take corrective action. Similarly, if the robot moved off centre, the IMU would measure a change in roll angle, feeding back to control algorithm. These two methods of centring will be tested in later chapters to decide on the sensor configuration.

4.2. Detecting Robots

To identify another robots and obstacles, two proposed methods were identified, a front mounted camera or a vertical 2D LiDAR. Both sensors would detect a change in the environment, resulting in the control algorithm classifying the obstacle type.

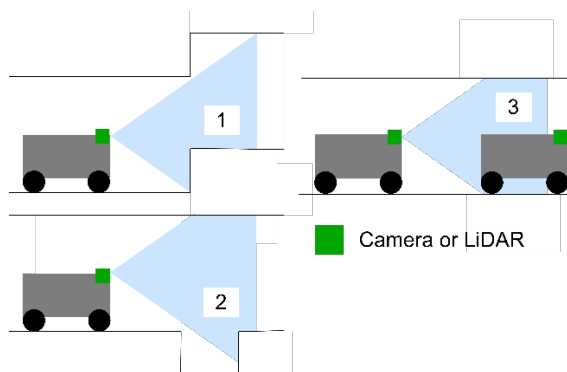


Figure 4: Detecting Robots Sensor Options

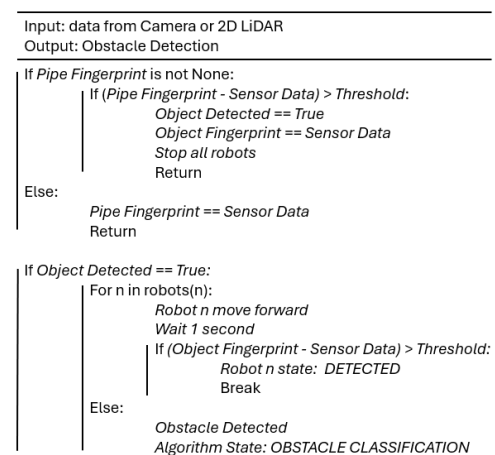


Figure 5: Detecting Robots Pseudo Code

Figure 4 illustrates the different objects the sensors would encounter, 1. Step, 2. Bridge, 3. Robot. Figure 5 shows pseudo code of how the control algorithm would differentiate between an obstacle and a robot.

4.3. Aligning and Connecting Robots

To detect whether robots were aligned, three different alignment detection methods were proposed as shown in Figure 6:

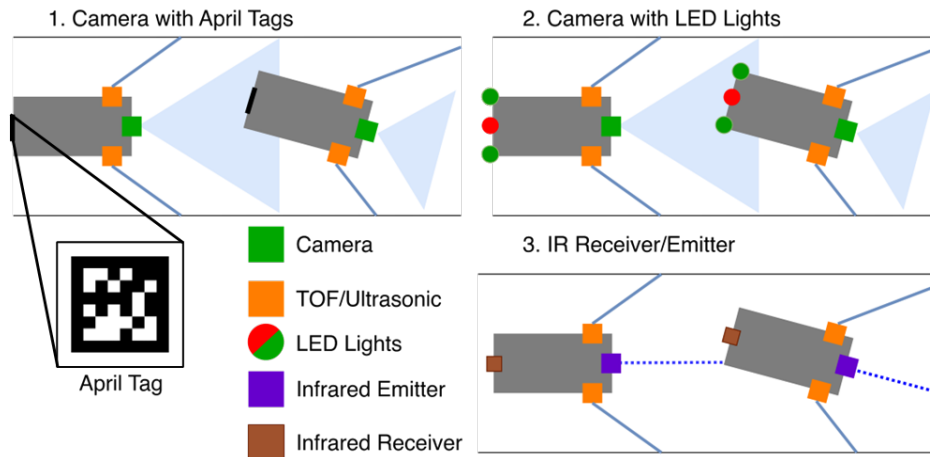


Figure 6: Aligning and Connecting Robots Sensor Options

A camera and April Tags provides the exact angle difference between two robots using an established image recognition algorithm. A camera and LED lights works similarly, by calculating the distance between each LED light, the angle difference between two robots could be determined. An IR Emitter/Receiver can only determine whether the robots are aligned. If misaligned, the pseudo code within Figure () describes how the robots would align. Pseudo code in Figure () outlines how the robots would connect, once aligned.

```

Input: Camera or IR Detector/Emitter
Output: Algorithm State

If Alignment Status is True:
    Stop all robots
    Algorithm State: LOCKING & DOCKING
Else:
    Twist CONNECTION TARGET to the left
    Wait 1 second
    Twist CONNECTION TARGET to the right
    Wait 2 second
    Twist CONNECTION TARGET to the left
    Wait 1 second

    Algorithm State: RESETING
  
```

Figure 7: Aligning Robots Pseudo Code

```

Input: Camera, IR Detector/Emitter or LiDAR
Output: Connection Status

If Alignment Status is True:
    Move CONNECTION SEEKER Forward
    Reverse CONNECTION TARGET
    If Distance < Threshold Distance
        Wait 5 seconds
        Lock CONNECTION SEEKER Male Linkage
Else:
    Algorithm State: ALIGNING

If CONNECTION SEEKER Male Locked is True:
    Stop all robots
    Locked Fingerprint == Sensor Data
    Reverse CONNECTION SEEKER

If Locked Fingerprint is not None:
    Wait 1 second
    If (Locked Fingerprint - Sensor Data) > Threshold
        Algorithm State: RESETING
    Else:
        Robot State: CONNECTION SEEKER = CONNECTED
        CONNECTION TARGET = CONNECTED
        Algorithm State: STOP ALL ROBOTS
  
```

Figure 8: Connecting Robots Pseudo Code

4.4. Validation of Physical Sensors

Having identified suitable sensor fusion arrangements, following extensive research, considering size, cost and specifications, the sensors in Table 3 were purchased.

Table 3: Sensor Specifications

Sensor		Ultrasonic	TOF	IR Emitter & Receiver	IR Obstacle	IMU
Manufacturer		Multicomp [27]	ST FlightSense [28]	UMT Media [29]	Youmile [30]	Inven Sense [31]
Part Number		HC-SR04	VL53L1X	EP0049	TS-YM-070	MPU 6050
Specification	Min Range (cm)	2	4	N/A	2	N/A
	Max Range (cm)	400	400	N/A	30	N/A
	Resolution (cm)	0.3	0.25	N/A	N/A	N/A
	Angle (°)	N/A	N/A	N/A	35	N/A
Cost per unit		£3.27	£16.50	£3.05	£1.14	£2.68

The camera was omitted due to memory limitations of the chosen microcontroller (MCU), the ESP-WROOM-32. The LiDAR was omitted due to delivery delays. An additional IR Obstacle Detection sensor was tested alongside the IR Emitter/Receiver, which works similarly but relies on one sensor detecting reflected IR light.

4.4.1. Range and Alignment Sensor Tests

To verify manufacturer sensor specifications, compare different sensor types, and validate sensor functionality, a series of physical tests were carried out. All range and alignment sensors were tested using a consistent procedure, distance and angle values were systematically varied, and the resulting data was recorded. Figure 9 illustrates the experimental setup used for these tests.

4.4.1.1 Diagram and Results

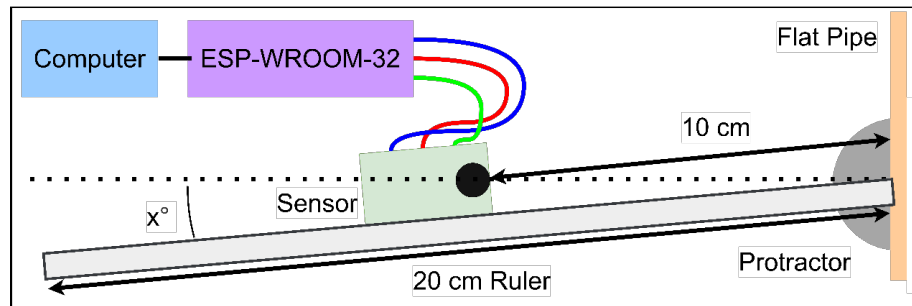


Figure 9: Range and Alignment Experimental Setup

Ultrasonic and TOF distance and angle data are compared in Figures 10 and 11. The IR Emitter/Receiver and the Obstacle Detection sensors do not give distance values, solely giving true or negative values. The IR Obstacle detector features a potentiometer to adjust the sensor sensitivity. Figures 12 and 13 show the results of distance and angle tests for both sensors, including different setting for the IR Obstacle Detector.

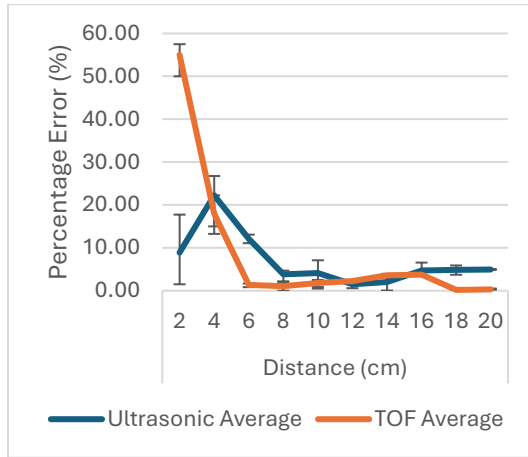


Figure 10: TOF and Ultrasonic Distance Error

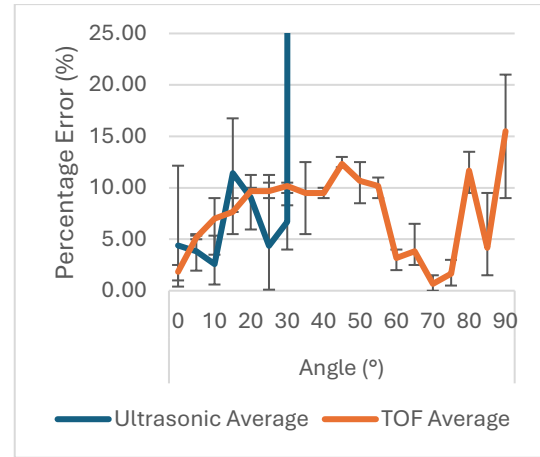


Figure 11: TOF and Ultrasonic Angle Error

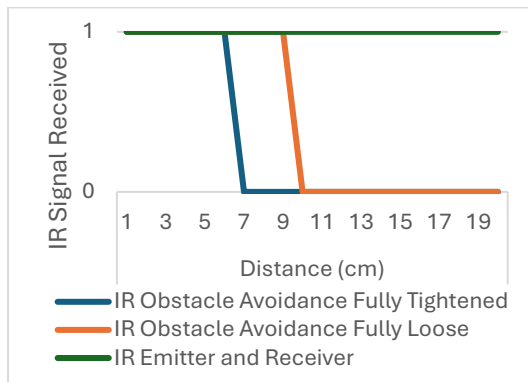


Figure 12: IR Sensors Distance Results

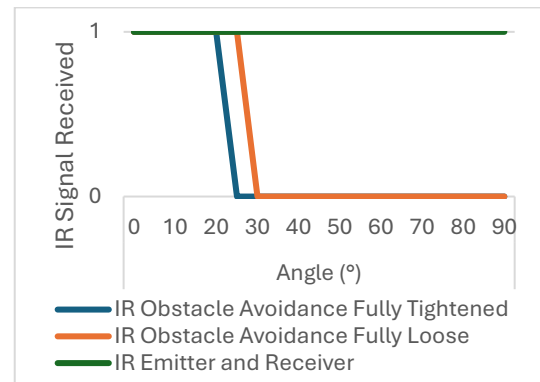


Figure 13: IR Sensors Angle Results

4.4.1.2 Discussion

The TOF sensor similarly to the manufacturer's specifications, exhibiting an average distance error of 0.253 cm – very close to the claimed 0.25 cm. Accuracy deteriorated significantly at close range; at 2 cm, the sensor had a 55% error, supporting the manufacturer's claim that the sensor is only reliable at distances of 4 cm and greater. Varying the angle of incidence increased the error, although the sensor still maintained an error margin under 16%.

In contrast, the ultrasonic sensor outperformed the TOF sensor at close range. It had an average overall error of 0.561 cm, compared to a specified accuracy of 0.3 cm. Excluding the 2 cm data point, the ultrasonic sensor was 54% less accurate than the TOF sensor across the tested range. Additionally, it failed to detect objects distance reliably beyond an incidence angle of 25°.

The IR emitter/receiver pair registered a positive signal at all tested distances and angles, indicating a lack off distance discrimination, rendering it unsuitable for robot detection and alignment. In contrast, the IR Obstacle avoidance sensors output varied with both distance and angle. When the onboard potentiometer was fully tightened, reflections were only recorded under more constrained conditions.

4.4.2. IMU Sensor Tests

The IMU MPU 6050 by InvenSense provides three acceleration readings (a_x, a_y, a_z) and three gyroscopic readings (g_x, g_y, g_z). Due to the tendency of gyroscopic readings to drift over time, when integrated, only the accelerometer data was used to calculate the roll and pitch. Yaw cannot be determined, as rotation about the z-axis does not change the direction of the gravity vector. Roll and pitch were calculated using Equations 1 and 2, derived from the projection of the gravity vector onto local axes.

$$pitch = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad roll = \arctan\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right) \quad (1\&2)$$

InvenSense specified the maximum Roll and Pitch error to be 2.86° and 4.57° , respectively. Whilst it was difficult to validate the accuracy of the sensor without dedicated testing equipment, a rudimentary test was formulated as shown in Figure 14. The angle of the IMU was physically varied and the IMU data was recorded.

4.4.2.1 Diagram and Results

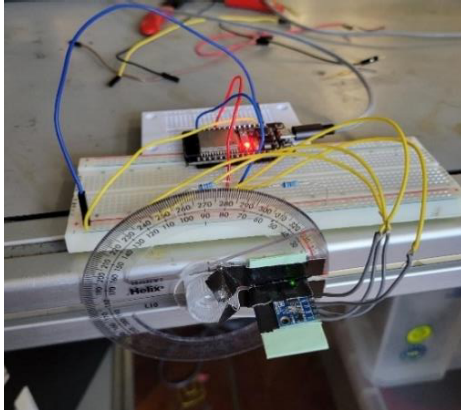


Figure 14: Experimental IMU Sensor Setup

Attempt	Rig		IMU	
	Pitch	Roll	Pitch	Roll
1	0	0	-2.00	-2.00
2	0	0	-0.74	-2.10
3	0	0	0.11	-1.27
1	-90	0	-88.8	-3.44
2	-90	0	-84.6	-3.87
3	-90	0	-85.4	-1.20
1	0	90	-3.50	88.2
2	0	90	-2.80	87.6
3	0	90	-0.30	87.8

Table 4: Rig and IMU Experimental Data

4.4.2.1 Discussion

On average, there was a 2.13° difference between the experimental readings and sensor readings, less than the claimed maximum error, validating the IMU's accuracy.

4.5. Sensor Selection Conclusion

Physical testing identified the TOF sensor as the most accurate for obstacle detection, though unsuitable below 4 cm. The ultrasonic sensor was more effective at close range but suffered from angular sensitivity and reduced overall accuracy. The IR obstacle avoidance sensor outperformed the basic IR emitter/receiver, which lacked distance discrimination. The IMU produced reliable pitch and roll measurements. Despite some measurement error due to low-resolution tools (protractor and ruler) and human experimental error, the physical tests validated a robust sensor suite for use on a robot.

Chapter 5: Simulation Setup, Testing and Results

The control algorithm was initially developed and tested in the simulation environment Gazebo, to establish functional code.

5.1. Simulation Setup

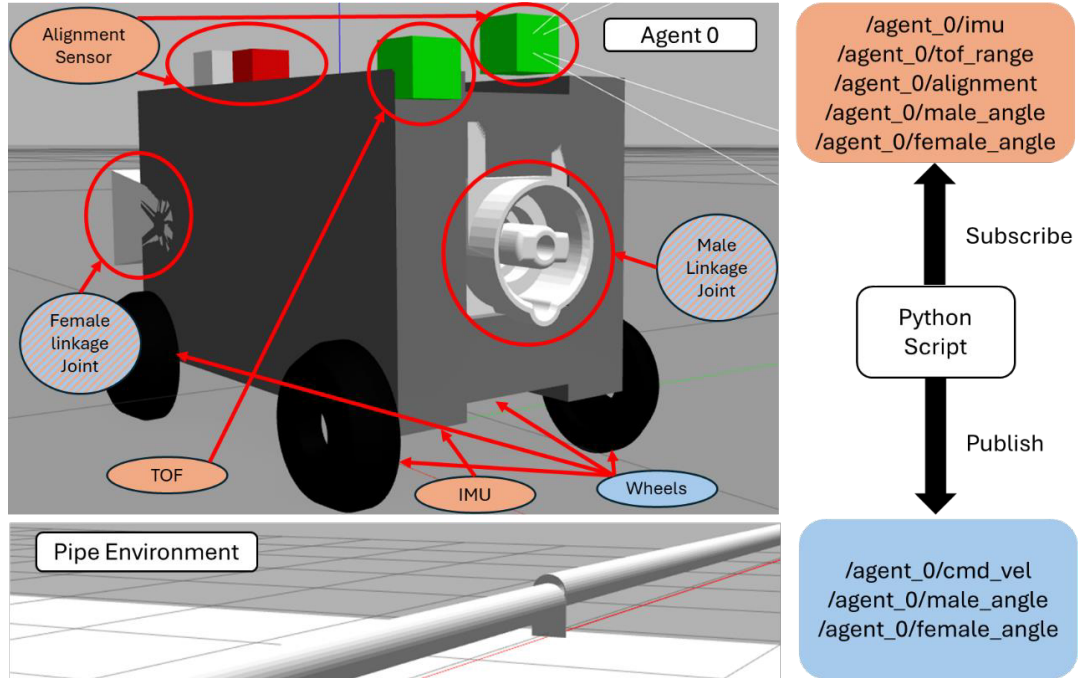


Figure 15: Simulation and Control Algorithm Setup

Figure 15 shows how the simulation was setup. A virtual pipe robot - equipped with all required sensors and actuators - was developed in Gazebo, alongside two virtual pipe environments containing the predefined obstacles. The simulation operates through ROS2, where sensors (orange) and actuators (blue) publish and subscribe to dedicated topics. A python script interacts with the simulation by subscribing to sensor topics, processing the data, and publishing commands to actuator topics to control the robot's behaviour. The simulation was designed to be scalable, enabling the creation and simultaneously operation of multiple robots.

5.2. Verification of Simulation Actuator Functionality

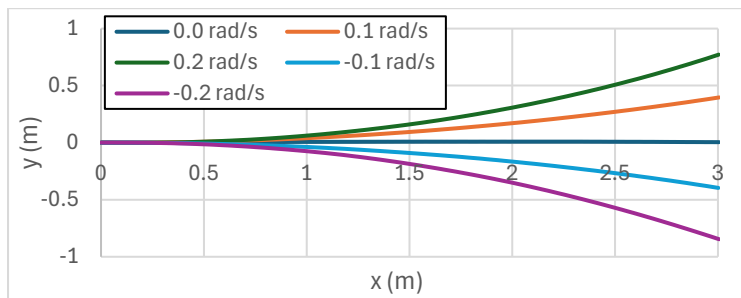


Figure 16: Verification of Wheel Functionality

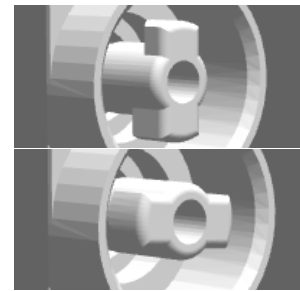


Figure 17: Verification of Male Linkage Functionality

To verify the functionality of the wheels, linear (0.5 m/s) and angular commands were given to the robot on a flat plane and the output was recorded in Figure 16. To test the functionality of the male linkage, it was locked and unlocked, as seen in Figure 17.

5.3. Verification and Testing of Simulation Sensors

5.3.1 IMU and TOF

To verify the functionality of the IMU and TOF sensors, a robot was placed inside a 15 cm diameter pipe at three yaw angles: 0°, +5° and -5°. As the robot traversed along the inner surface of the pipe, changes in roll angle and TOF range readings were recorded, verifying that both sensors work (Figure 19, Tables 5 and 6):

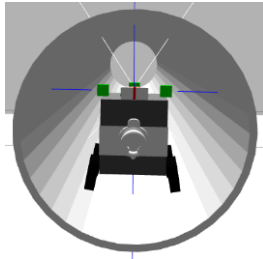


Figure 18: Robot conducting IMU and TOF Test

Table 5: Side TOF Range Readings

Start Angle (°)	0	+5	-5
Max Difference (cm)	0.0519	0.5609	0.4552
Max Difference (%)	1.04%	10.67%	9.58%

Table 6: IMU Angle Readings

Start Angle (°)	0		+5		-5	
Dimension	Roll	Pitch	Roll	Pitch	Roll	Pitch
Max Angle (°)	3.72	1.23	22.13	3.58	24.79	1.23

The IMU showed a maximum absolute roll difference of 10.73% between the +5° and -5° tests, indicating consistent and repeatable simulation performance. The TOF sensors registered a maximum left–right difference of 0.56 cm—above the quoted 0.25 cm resolution—however this small error range was insufficient for precise control. Given the physical IMU's quoted maximum error of 4.7° and observed roll angles of over 20°, the IMU was determined to be the more suitable sensor for detecting robot orientation.

5.3.2 IR Obstacle Detector

ROS2 includes plugins for many common sensors, allowing rapid simulation development. However, it does not directly have an IR Obstacle Detection sensor. To simulate this, a ROS2 plugin camera and a red detection object were created. A python script subscribed to the camera feed topic and processed the image data; if red pixels were detected, a custom ROS2 topic was published to indicate robot alignment. To ensure consistency with the physical sensor, a calibration test was conducted within the simulation. The virtual sensor setup was adjusted to match the physical sensor, as shown in Section 4.4.1.1 the real obstacle detection sensor could detect obstacles from 6 cm at a 20° offset. Figures 19 and 20 show the simulation IR Obstacle Detection calibration test. At a 20° offset, the seeker robot can just see the red detection object on the target robot, indicating the robots are aligned.

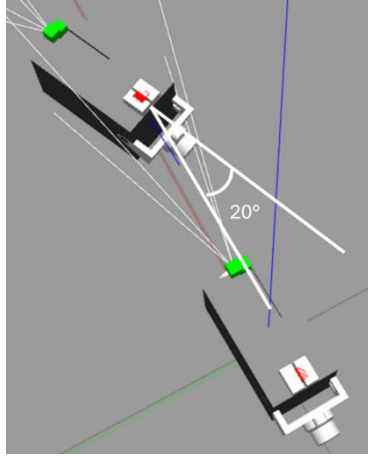


Figure 19: Top-Down View of the Simulation IR Obstacle Detector Calibration Test

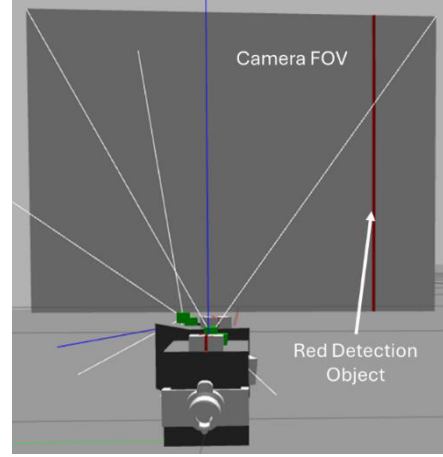


Figure 20: Back View of the Simulation IR Obstacle Detector Calibration Test

5.4. Control Algorithm Testing

The three functions outlined in Section 4 Sensor Selection were tested within the simulation before being combined into the larger FSM control algorithm.

5.4.1 Centring Robots

A test was conducted in a scaled-up pipe environment of diameter 30 mm to reduce the self-centring effect present in smaller pipes. Equation 3 describes how angular velocity ($\dot{\theta}$) was calculated:

$$\dot{\theta} = K_p * \theta \quad (3)$$

Where θ is the roll angle in radians. A larger roll angle results in a larger corrective angular velocity command. The value of proportional gain (K_p) was manually tuned for both simulated and physical environments. Figure 22 shows the tuning results:

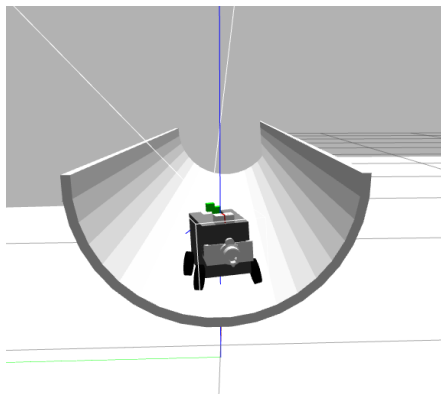


Figure 21: Simulation Centring Test

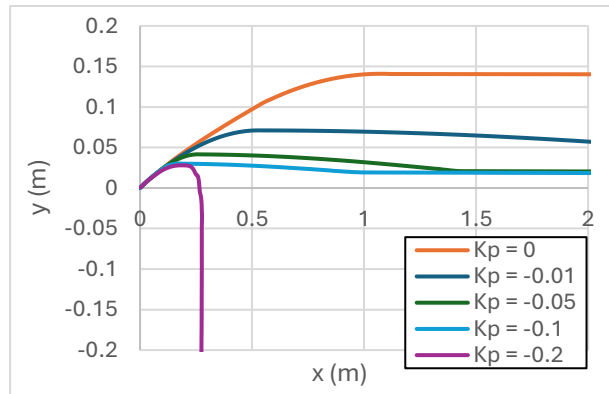


Figure 22: Kp Tuning Results

When $K_p = -0.2$, the robot overshoot its centre position, indicating overcorrection. An optimal K_p value of -0.1 was selected, as it enabled the robot to self-centred over the shortest distance. The negative sign ensures the corrective angular velocity acts in opposite direction of the roll angle.

5.4.2 Detecting, Aligning and Connecting Robots

To evaluate the performance of the robot detection code described in Section 4.2, three obstacle types - a step obstacle, a bridge obstacle and a robot obstacle – were each tested three times. The results are recorded in Table 7, with the algorithm correctly identifying all obstacles in all test attempts, achieving a 100% success rate.

Table 7: Detecting Robots Results

Obstacle Type		Correctly Identified Obstacle		
		Bridge	Step	Robot
Attempt	1	TRUE	TRUE	TRUE
	2	TRUE	TRUE	TRUE
	3	TRUE	TRUE	TRUE
Success Rate (%)		100	100	100

Table 8: Aligning Robots Results

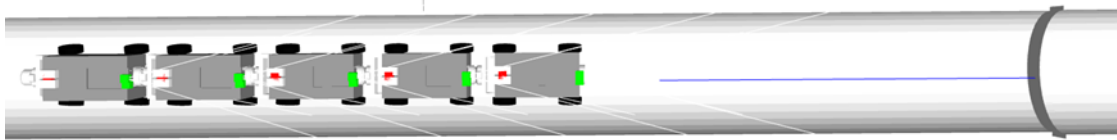
Pipe Size (cm)		15	20.32	30
Successfully Connected	1	TRUE	TRUE	TRUE
	2	TRUE	TRUE	TRUE
	3	TRUE	TRUE	TRUE
Success Rate (%)		100	100	100

To assess the effectiveness of the robot alignment and connection functions tests were conducted in three different pipe sizes (15 cm, 20.32 cm and 30 cm in diameter). As the pipe diameter increased, the self-centring geometric effect diminished, making alignment more challenging. Table 8 summarises the outcomes, showing successful robot connections in all tests.

5.5. Control Algorithm Simulation Results and Discussion

Following verification of all simulation components – actuators, sensors and control functions – the full control algorithm was evaluated in Gazebo by varying the number of robots and testing whether the control algorithm could successfully link all robots.

The control algorithm successfully navigated each robot within the simulated pipe environment, using sensors to detect robots, self-centring within the pipe, aligning robots and initiating the linking sequence. A maximum of five robots were tested and confirmed as successfully forming a continuous chain, validating the modular design and algorithm logic, robustness and scalability. The algorithm could support distributed coordination, each robots acted independently yet contributed to the shared task of forming a physically linking chain.



Having demonstrated the control algorithms functionality within a simulation environment, physical testing (Section 6) was required to validate its performance under real world conditions.

Chapter 6: Physical Setup, Testing and Results

6.1. Physical Setup

Figure 23 illustrates the physical control system used to control the modular robots. Each robot's equipped microcontroller ran Arduino code (with a unique IP address). The microcontroller handled direct communication with onboard sensors and actuators, while also broadcasting this data over Wi-Fi to a laptop. On the laptop, a python script acted as a ROS2 bridge, subscribing to and publishing messages that mimicked standard ROS2 topics. Enabling the main control algorithm - implemented in Python - to interface seamlessly with the physical robots without modifications.

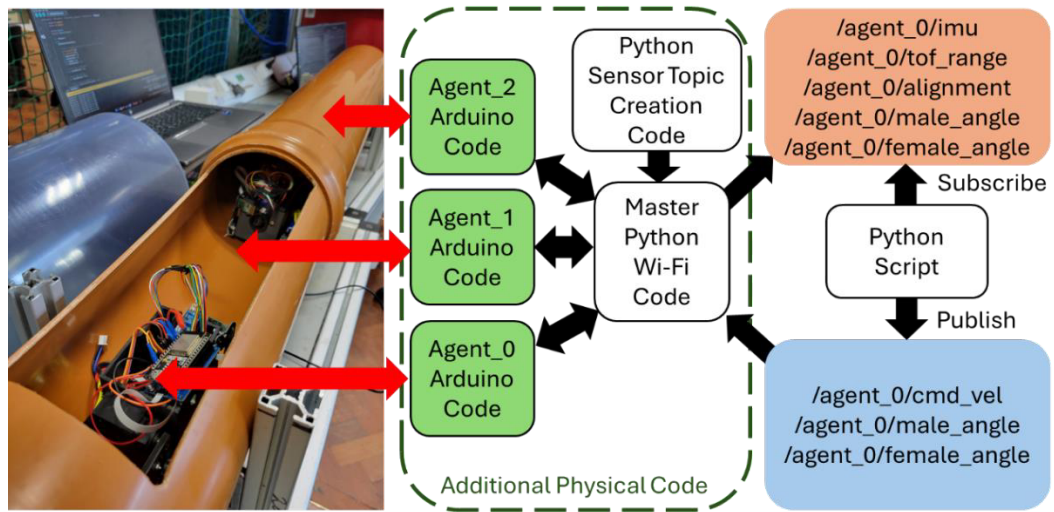


Figure 23: Physical Setup and Control Algorithm Communication

6.2. Verification of Physical Setup, Sensors and Actuators

After all robots were assembled, the ROS2 Wi-Fi control setup, along with each sensor and actuator were tested. A robot was commanded drive in a straight line along a flat surface, whilst collecting sensor data. The recorded data is shown in Figure 24:

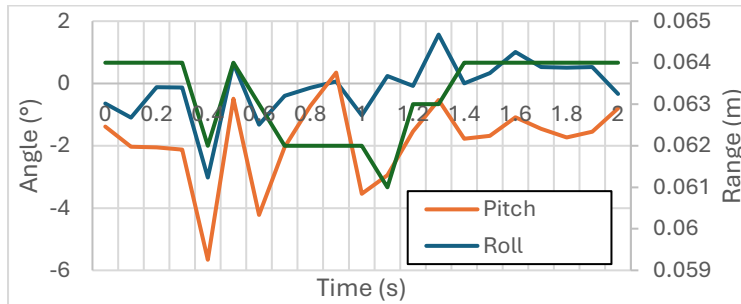


Figure 24: Sensor Noise

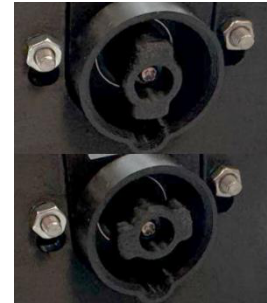


Figure 25: Male Linkage

The robot moved along the ground, veering slightly to the left, verifying the operation of the wheels and motors, but indicating there was a motor speed mismatch. While the robot travelled on level ground, the IMU roll angle readings fluctuated between -3.02° and 1.57° . To accommodate this noise, a tolerance of $\pm 5^\circ$ was used in the centring

logic. The TOF sensors reported a variation of 0.003 m, therefore a threshold of 0.005 m was set to filter noise in range-based decisions. Figure 25, shows the male linkage in its locked and unlocked position, confirming its mechanical and control functionality.

6.3. Physical Testing

The rotational speed of each wheel was measured independently while varying motor speed commands. The linear speed of the robot was recorded to evaluate drive performance. Both tests are depicted in Figure 26. Response delay time was determined by sending a motor command via the control algorithm and recording two key timestamps: when the message was received by the Wi-Fi python interface, and when the wheels physically responded. The results are shown in Table 9:

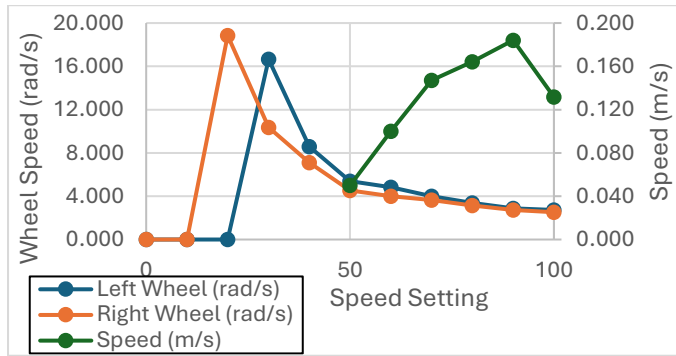


Figure 26: Speed Setting vs Linear and Wheel Speed

Attempt	Time (s)	
	Message Received	Wheels Moved
1	0.12	1.30
2	0.14	1.40
3	0.09	1.10
Avg	0.12	1.27

Table 9: Response Delay Time

6.4. Physical Results and Discussion

Figures 27 and 28, show the final physical test setup. Multiple tests were conducted to tune control tolerances, movement speeds and timing parameters to suit real-world conditions. Despite this, the control algorithm failed to physically link any robots.

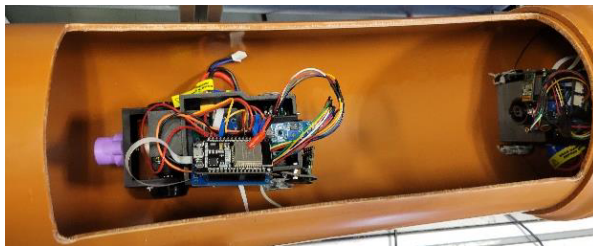


Figure 27: Top-Down View of Robots Enacting Control Algorithm



Figure 28: Back View of Robot

The main reasons for failure were:

- Delayed response times: The average delay between sending a command and the robot's physical response was 1.27 s. This lag prevented timely corrections and often led to collisions and misalignment.
- Inconsistent wheel speeds: Figure 26 shows a 19.1% difference in left and right wheel rotation speeds. This asymmetry significantly impaired the robot's ability to maintain a straight path, undermining the centring and alignment functions.

Chapter 7: Conclusion and Future Work

7.1. Conclusion

A ten-state FSM control algorithm was successfully developed to govern the pre-linking behaviour of the modular pipe robots. Key functions – centring, detecting, aligning, and connecting – were achieved through carefully designed sensor fusion layouts, selected based on physical testing and practical constraints.

A scalable simulation environment was created, verified for functionality and used to test and develop the control algorithm. Within the simulation, the algorithm successfully connected up to five modular robots, demonstrating both functionality and scalability.

Three physical robots were constructed, with all sensor and actuators verified via ROS2-based Wi-Fi control. Whilst the algorithm performed reliably in simulation, it failed to achieve successful linkage in a physical environment. The primary issues were significant delays in command response and mismatches in wheel traction, which could not be mitigated through parameter tuning. These factors ultimately limited the robot's ability to centre and align in real-time. In simulation, message passing was effectively instantaneous, masking latency-related issues. Although simulations introduce some noise, they often fail to replicate the random variability present in physical systems - such as minor wheel traction differences. This illustrates the well documented “reality gap” between simulated and real-world environments and highlights key challenges in translating a simulation-based control algorithm into a real-world physical environment.

Despite this, the project successfully delivered a proven robust and scalable control algorithm that is ready for deployment on an improved physical hardware platform.

7.2. Future Work

Future work should focus on addressing the limitations identified in the physical testing. To reduce response delays, a lower-latency communication method could be implemented, such as onboard processing or a faster wireless protocol. Improving wheel traction and motor control accuracy would also enhance the robots' ability to centre, align, and connect reliably. Hardware upgrades, such as including motor encoders, may help reduce variability in wheel speeds. Furthermore, the simulation environment could be refined to better account for physical imperfections, narrowing the reality gap. Finally, future development could extend the control algorithm to manage larger swarms, more complex pipe networks, and dynamic environmental.

References

1. FLYABILITY, The Reality of Sewer Inspections - Everything You Need to Know. (n.d.). <https://www.flyability.com/blog/sewer-inspection> (accessed April 28, 2025).
2. Yusuf Allinson, Sewage in the UK - Market Research Report. (2024). <https://www.ibisworld.com/united-kingdom/industry/sewerage/2315/> (accessed April 28, 2025).
3. South West Water, Your wastewater. (n.d.). <https://www.southwestwater.co.uk/your-wastewater> (accessed April 28, 2025).
4. T. L. Nguyen, A. Blight, A. Pickering, G. Jackson-Mills, A. R. Barber, J. H. Boyle, R. Richardson, M. Dogar, & N. Cohen, Autonomous control for miniaturized mobile robots in unknown pipe networks. *Frontiers in Robotics and AI*, **9** (2022). <https://doi.org/10.3389/frobt.2022.997415>.
5. Y. Zhang, H. Chen, L. Wang, Z. Fu, & S. Wang, Design of a Novel Modular Serial Pipeline Inspection Robot. *2023 IEEE International Conference on Mechatronics and Automation, ICMA 2023* (Institute of Electrical and Electronics Engineers Inc., 2023), pp. 1847–1852. <https://doi.org/10.1109/ICMA57826.2023.10216215>.
6. R. S. Elankavi, D. Dinakaran, R. M. K. Chetty, M. M. Ramya, & D. G. H. Samuel, A Review on Wheeled Type In-Pipe Inspection Robot. *International Journal of Mechanical Engineering and Robotics Research*, **11** (2022) 745–754. <https://doi.org/10.18178/ijmerr.11.10.745-754>.
7. S. Kazeminasab, N. Sadeghi, V. Janfaza, M. Razavi, S. Ziyadidegan, & M. K. Banks, Localization, Mapping, Navigation, and Inspection Methods in In-Pipe Robots: A Review. *IEEE Access*, **9** (2021) 162035–162058. <https://doi.org/10.1109/ACCESS.2021.3130233>.
8. H. Jang, T. Y. Kim, Y. C. Lee, Y. S. Kim, J. Kim, H. Y. Lee, & H. R. Choi, A Review: Technological Trends and Development Direction of Pipeline Robot Systems. *Journal of Intelligent and Robotic Systems: Theory and Applications*, **105** (2022). <https://doi.org/10.1007/s10846-022-01669-2>.
9. R. H. Peck, J. Timmis, & A. M. Tyrrell, Omni-pi-tent: an omnidirectional modular robot with genderless docking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **11650 LNAI** (2019) 307–318. https://doi.org/10.1007/978-3-030-25332-5_27/FIGURES/9.
10. S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, P. Corradi, & L. Ricotti, Symbiotic robot organisms: Replicator and Symbion projects. *Performance Metrics for Intelligent Systems (PerMIS) Workshop* (2008), pp. 62–69. <https://doi.org/10.1145/1774674.1774685>.

11. C. Liu, Q. Lin, H. Kim, & M. Yim, SMORES-EP, a Modular Robot with Parallel Self-assembly. (2021). <https://doi.org/10.1007/s10514-022-10078-1>.
12. H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, & S. Murata, Distributed self-reconfiguration of M-TRAN III modular robotic system. *International Journal of Robotics Research*, **27** (2008) 373–386. <https://doi.org/10.1177/0278364907085560>.
13. H. Wei, Y. Chen, J. Tan, & T. Wang, Sambot: A self-assembly modular robot system. *IEEE/ASME Transactions on Mechatronics*, **16** (2011) 745–757. <https://doi.org/10.1109/TMECH.2010.2085009>.
14. M. Delrobaei & K. A. McIsaac, Design and Steering Control of a Center-Articulated Mobile Robot Module. *Journal of Robotics*, **2011** (2011) 1–14. <https://doi.org/10.1155/2011/621879>.
15. H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, & S. Murata, Distributed self-reconfiguration of M-TRAN III modular robotic system. *International Journal of Robotics Research*, **27** (2008) 373–386. <https://doi.org/10.1177/0278364907085560>.
16. Wen Zhao, Mitsuhiro Kamezaki, Kento Yoshida, Minoru Konno, Akihiko Onuki, & Shigeki Sugano, *An Automatic Tracked Robot Chain System for Gas Pipeline Inspection and Maintenance Based on Wireless Relay Communication* (IEEE, 2018).
17. X. (Xiangshuai &) Zeng, *Reinforcement learning based approach for the navigation of a pipe-inspection robot at sharp pipe corners*.
18. Hiroyuki Masuta, Hisato Watanabe, Kaname Sato, & Hun-ok Lim, *Recognition of Branch Pipe for Pipe Inspection Robot using Fiber Grating Vision Sensor* (IEEE, 2013).
19. L. Brown, J. Carrasco, S. Watson, & B. Lennox, Elbow Detection in Pipes for Autonomous Navigation of Inspection Robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*, **95** (2019) 527–541. <https://doi.org/10.1007/s10846-018-0904-7>.
20. S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, & S. Kokaji, M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, **7** (2002) 431–441. <https://doi.org/10.1109/TMECH.2002.806220>.
21. H. Song, K. Ge, D. Qu, H. Wu, & J. Yang, Design of in-pipe robot based on inertial positioning and visual detection. *Advances in Mechanical Engineering*, **8** (2016) 1–22. <https://doi.org/10.1177/1687814016667679>.
22. Xylem Eritrea, YSI YSI EXO3s Multiparameter Sonde. (n.d.). <https://www.xylem.com/en-er/products--services/analytical-instruments-and-equipment/monitoring-sampling-instruments-sensors->

equipment/multiparameter-sondes-sensors-accessories/exo3s-multiparameter-sonde/ (accessed April 28, 2025).

23. S. Edwards, R. Zhang, R. Worley, L. Mihaylova, J. Aitken, & S. R. Anderson, A robust method for approximate visual robot localization in feature-sparse sewer pipes. *Frontiers in Robotics and AI*, **10** (2023).
<https://doi.org/10.3389/frobt.2023.1150508>.
24. A. Oyama, H. Iida, Y. Ji, K. Umeda, Y. Mano, T. Yasui, & T. Nakamura, *Three-dimensional Mapping of Pipeline from Inside Images Using Earthworm Robot Equipped with Camera*.
25. Lentin Joseph, *Mastering ROS for Robotics Programming: Design, Build, and Simulate Complex Robots Using Robot Operating System and Master Its Out-of-the-box Functionalities* (Packt Pub Ltd, 2015).
26. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, & P. Abbeel, Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. (2017).
27. MultiComp, *Ultrasonic Ranging Module HC-SR04*.
28. ST FlightSense, *VL53L1X Datasheet*.
29. UMTMedia, Digital 38kHz IR Infrared Receiver And Transmitter Set Sensor Module Compatible With Arduino And Raspberry Pi - UMTMedia. (n.d.).
<https://www.amazon.co.uk/Infrared-Receiver-Transmitter-Compatible-Raspberry/dp/B0CY2XHRC7> (accessed May 1, 2025).
30. Youmile, IR Infrared Obstacle Avoidance Sensor. (n.d.).
<https://www.amazon.co.uk/Youmile-Infrared-Obstacle-Avoidance-Arduino/dp/B07PY3CVSV> (accessed May 1, 2025).
31. *MPU-6000 and MPU-6050 Product Specification Revision 3.4 MPU-6000/MPU-6050 Product Specification* (2013).

Appendix

All code can be found at:

<https://dev.azure.com/islamtagi/Pipe%20Swarm%20Project>

Figures 29 to 32 show the pictures of the physical tests undertaken:

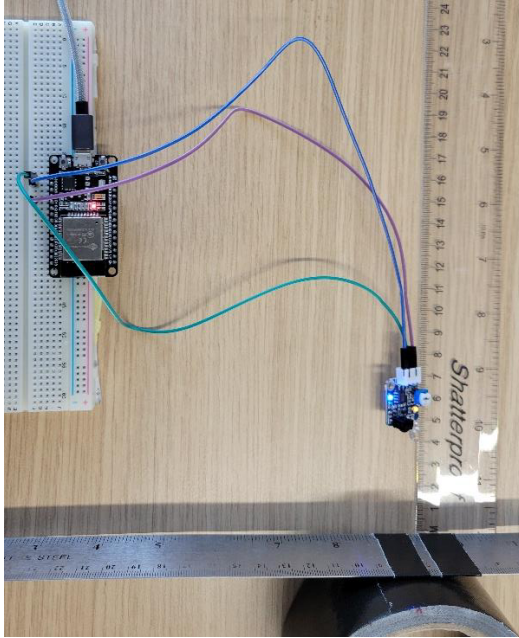


Figure 29: IR Obstacle Detection Sensor Testing:

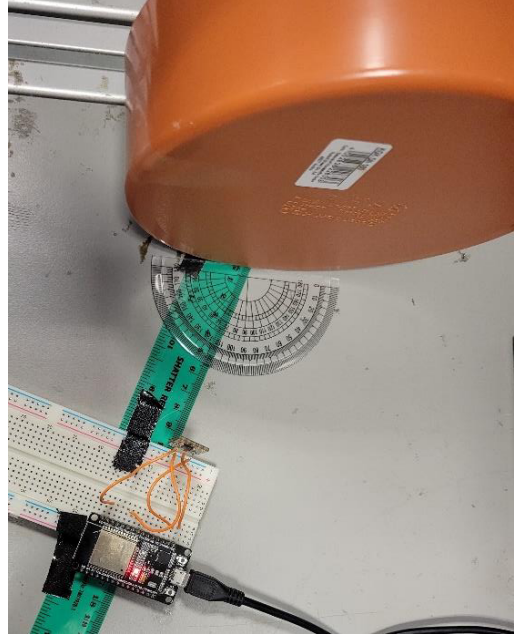


Figure 30: TOF Sensor Testing

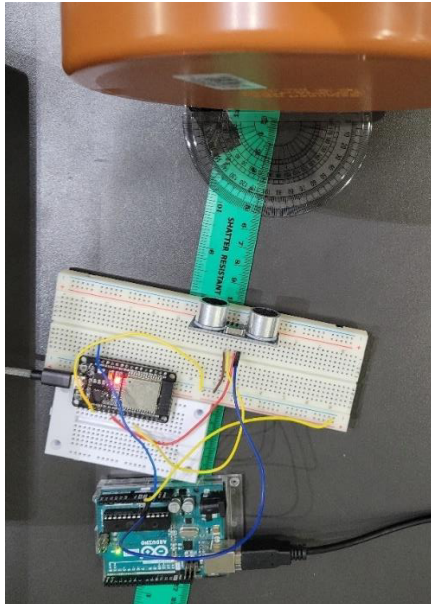


Figure 31: Ultrasonic Sensor Testing

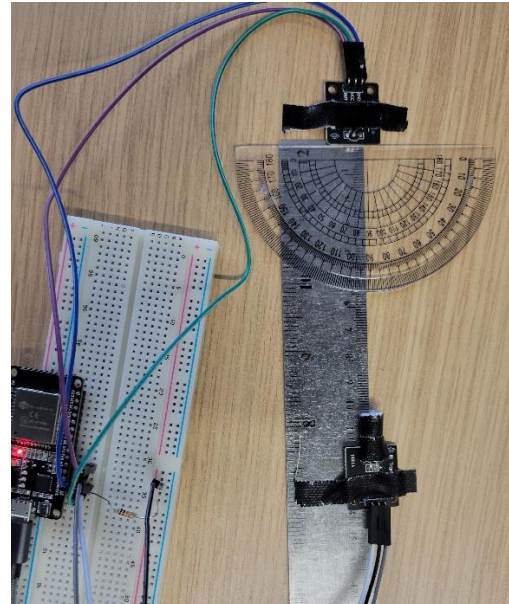


Figure 32: IR Receiver/Emitter Sensor Testing

Meeting Logs

Meeting 1

Date	01/11/2024
Progress Since Last Meeting	• Completed the initial draft of the Coursework Project Proposal (CPP). • Assigned project roles (Path Planning, Electronic/Drive Systems, Mechanical Design/FEA, Obstacle Detection). • Conducted general research on pipe exploration, obstacle detection, modular robots, and swarm control. • Discussed methodologies for robot design, control methods, and testing environments. • Gathered inspiration for potential locking mechanisms (e.g., latches, electromagnets, hook & latch, interlocking mechanisms, snap-fits, solenoids, bow release).
Planned Work Until Next Meeting	• Finalize the CPP (refine Introduction, Aims/Objectives, Deliverables, Project Tasks, Resources Declaration, Gantt Chart, Risk Management, Ethical Considerations). • Continue research in assigned work areas. • Consider power constraints for design concepts.
Supervisor Signature	ARB

Meeting 2

Date	15/11/2024
Progress Since Last Meeting	• Conducted initial component searches. • Discussed design mechanisms and component types. • Progressed with drafting the CPP.
Planned Work Until Next Meeting	• Finalize list of components (to borrow and to purchase). • Decide drivetrain configuration (e.g., 2-motor system). • Select sensors (ultrasonic considered). • Assemble basic prototype for Week 11 presentation. • Review GrabCAD models (targeting 300mm pipes). • Investigate servos and power requirements. • Research sensors, actuators, MCU, power solutions. • Start presentation draft and component purchases.
Supervisor Signature	ARB

Meeting 3

Date	22/11/2024
Progress Since Last Meeting	• Investigated power consumption: batteries and motors. • Created chassis blockout model. • Researched sensors (ToF, Camera, Ultrasonic, LiDAR, AprilTags). • Investigated motor options and connection mechanisms. • Researched MCUs and communication methods (Zigbee, Bluetooth). • Estimated BOM costs and power draw.
Planned Work Until Next Meeting	• Select the best camera option and check availability. • Investigate robot detection through communication/odometry. • Purchase initial test components (LiDAR, ToF, Ultrasonic, Camera). • Decide on drivetrain (wheels vs tracks, 2 vs 4 wheel drive). • Design and evaluate linkage mechanisms. • Define project aims (e.g., 150mm pipe target). • Prepare for the presentation.
Supervisor Signature	ARB

Meeting 4

Date	29/11/2024
Progress Since Last Meeting	• Selected battery type, motor (N20 with 50:1 gear ratio), and communication method. • Listed components for purchase. • Defined system overview. • Drafted circuit diagrams and initial design blockouts. • Estimated max current draw and operation times.
Planned Work Until Next Meeting	• Finalize full circuit diagram (fuses, diodes, buck converters, IMU). • Determine current draw and select appropriate converters. • Choose wheel configuration (3 vs 4 wheels). • Confirm final battery choice (LiPo 2S 7.4V 450mAh).
Supervisor Signature	ARB

Meeting 5

Date	14/02/2025
Progress Since Last Meeting	• Completed BOM. • Delivered first presentation. • Received several components. • Refined current draw estimates across voltage rails.
Planned Work Until Next Meeting	• Refine schematic and PCB design (correct ESP MCU, avoid breadboards, no sharp angles). • Implement safety features (polyfuses, kill switch, reverse polarity protection, low battery indicator). • Develop WebUI dashboard. • Begin circuit control tests with Arduino and WiFi.
Supervisor Signature	ARB

Meeting 6

Date	28/02/2025
Progress Since Last Meeting	• Printed and tested linkage prototype (4kg holding). • Completed individual component testing and initial coding. • Finalized schematic diagrams and initial PCB Gerber layout. • Created electronics 3D model and mass calculations. • Developed collision detection algorithm. • Carried out torque and step height calculations. • Advanced ROS2 simulation and MoveIt integration.
Planned Work Until Next Meeting	• Complete testing for DC Motors and Stepdown Converters. • Finalize power management, sensor connections on PCB. • Route PCB layout. • Finalize battery choice. • Decide between LiDAR or 3x ToF sensors. • Evaluate Infrared sensors. • Continue ROS2 development and finalize V1 robot design.
Supervisor Signature	ARB

Meeting 7

Date	07/03/2025
Progress Since Last Meeting	• Tested IR sensors, DC motors, motor drive. • Structured Arduino code. • Final PCB design based on selected sensors. • Selected 2S 450mAh battery. • Modeled V2 robot with linkage and compact features. • Printed TPU wheels. • Imported linkage into simulation and tested. • IMU data sent to ROS topic (with minor data loss).
Planned Work Until Next Meeting	• Test buck converters. • Fix DC motor code. • Integrate wireless communication. • Finalize full system testing. • Finalize TOF connector board and compact voltage regulators. • Manufacture V2 robot. • Complete and verify simulation functionalities.
Supervisor Signature	ARB

Meeting 8

Date	14/03/2025
Progress Since Last Meeting	• Completed linkage prototype and electronics model. • Completed initial PCB layout and BoM. • V1 robot design nearly finished.
Planned Work Until Next Meeting	• Design driven motor hub. • Update V2 robot design. • Plan detailed wheel tests (varying TPU infill, PLA toothed/smooth, incline/load/wet surface tests). • Design TOF sensor array.
Supervisor Signature	ARB

Meeting 9

Date	28/03/2025
Progress Since Last Meeting	• Manufacturing of V2 robot started. • TPU tyres printed. • Simulations for obstacles and IR sensors completed. • Final PCB and TOF array ordered. • WiFi communication set up. • Robot connected in MoveIt2. • Finalized ROS integration.
Planned Work Until Next Meeting	• Complete V2 manufacturing. • Test TPU tyres and assemble robots. • Verify simulations. • Integrate sensor data and obstacle detection. • Final system testing.
Supervisor Signature	ARB

Meeting 10

Date	10/04/2025
Progress Since Last Meeting	• Final PCBs received and soldered. • Robot assembly progressed (Robot 1 done, Robot 2 90%, Robot 3 parts printed). • Full Arduino/ROS integration tested. • Buck converters tuned. • Simulation matches physical robot behavior. • Complete modular robot simulation code written.
Planned Work Until Next Meeting	• Finalize servo control. • Complete circuit testing (wired and WiFi). • Finalize obstacle differentiation. • Complete robot assembly and full system physical testing.
Supervisor Signature	ARB