RoboCup Detail Design Report

ENMT301

Group 25

Henry Hall - 27711878 Alister Giles - 76347357 William Thorpe - 36133777

Executive Summary

From an initial concept, a design has been developed with multiple different subsystems. Each of these subsystems has significant functionality in its current state. These different subsystems include the chassis, pickup mechanism, navigation, movement, storage, and software. The chassis was simplified significantly by using preexisting side and top plates for the robot with open aluminium profiles to hold it together. The pickup mechanism has gone through several iterations during development. In its current design, the weights are knocked over by a funnel. Once inside the robot, the weights are dragged along the ground against a back plate. The weights can then be scooped a ramp into the robot using a large servo motor. The navigation algorithm uses a reactive method with wall-following. A finite-state machine (FSM) is used to control what the robot does at a given time. An array of time-of-flight sensors are used to provide information about the arena to the robot. The movement system uses tank tracks around a series of bearings and 3D printed rollers. This improves the torque and ground clearance of the robot. The robot needs to be capable of storing up to three weights. This was done by embedding aluminium rods into the top of the ramp from the pickup mechanism. A servo at the end of the rods stops the weights from falling out the end until desired. The software for the robot is written on the Arduino IDE. This uses a timer scheduler to control the frequency of four tasks that are essential to the robot's function.

Multiple tests were carried out to determine the performance of the different subsystems. The first tests determined the effectiveness of the pickup mechanism. This showed that the robot could pick up weights 92.5% of the time and would take an average of 8.5 seconds. The next test done was to evaluate the effectiveness of the navigation algorithm. This was done by laying out three different arenas that were divided into sectors. The robot was observed navigating through these arenas while the number of sectors entered was recorded to get a percentage of area covered by the robot. This showed that the robot covered about 60% of the arena on average and could reliably get itself unstuck from collisions. The next test was used to determine the speed of the robot moving forward and backward. This showed an average moving speed of approximately 0.35 m/s. The CPU load was calculated using the desired frequency of each task and the length of time the tasks took to complete.

Each of the subsystems of the robot was evaluated against the specification of requirements. Any possible improvements were also outlined. It was identified that the pickup mechanism had a potential weakness where it would jam if too many weights were collected consecutively. The navigation algorithm did not cover the entire arena during all the tests. A potential method to fix this would be to use localisation of the robot to change the heading to areas that have not been entered yet. The movement, storage, and software systems both worked adequately. Some small improvements to the strength of the storage could be made to improve reliability.

Introduction

This report describes the design and manufacture of a robot that will take part in the Robocup competition. Following completion of a conceptual design report, a design was chosen. The design used a scoop to collect weights before storing them within the chassis. The process of forming, improving, and implementing the chosen design has taken place over the last six weeks. Many improvements have been achieved through rapid prototyping and iterative design. The design choices made in relation to the six major subsystems: pickup, chassis, storage, navigation, movement, and software are described in detail. Each of these systems was tested and evaluated to see if the current design meets the functional requirements set out in an initial design report. A fault tree analysis was performed to determine potential points of failure the robot may experience during the competition. While the physical design is near completion the process of refinement and programming will continue until the competition. Any improvements noted in this report will be implemented during the remaining time.

Design Description

Overview

The design philosophy followed was mainly rapid prototyping. New subsystems and component designs were physically made and tested before being refined or phased out. This has caused the design of the robot to evolve over time. Its current state differs from the proposed design chosen in the initial design report. The current design's CAD model can be seen in Figure 1.

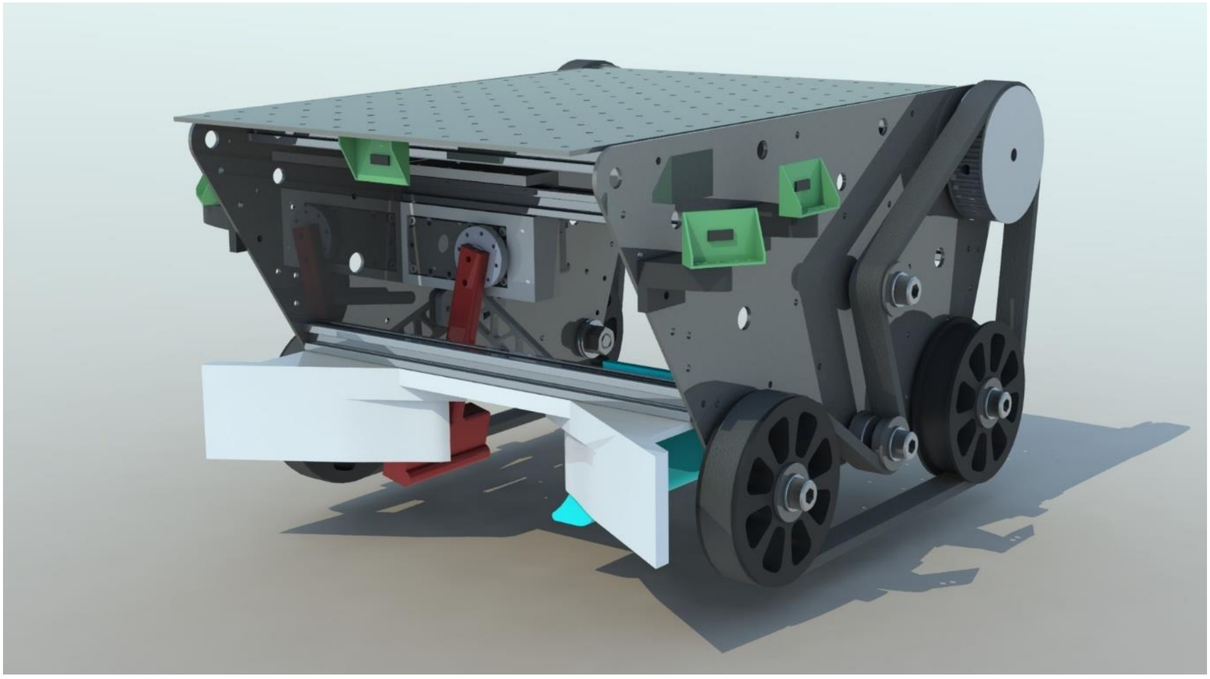


Figure 1: A rendered CAD model of the designed robot.

Figure 2 displays the functional architecture of the robot, showing different types of components used on the robot. The design can be split into six major subsystems: pickup, chassis, storage, navigation, movement, and software. Each of these subsystems was designed in tandem with each other, aiming to seamlessly integrate different modules on the final robot. Due to the requirement for total autonomy, the different functional groups are crucial to providing the robot with the information it needs. This allows it to make decisions and carry out actions.

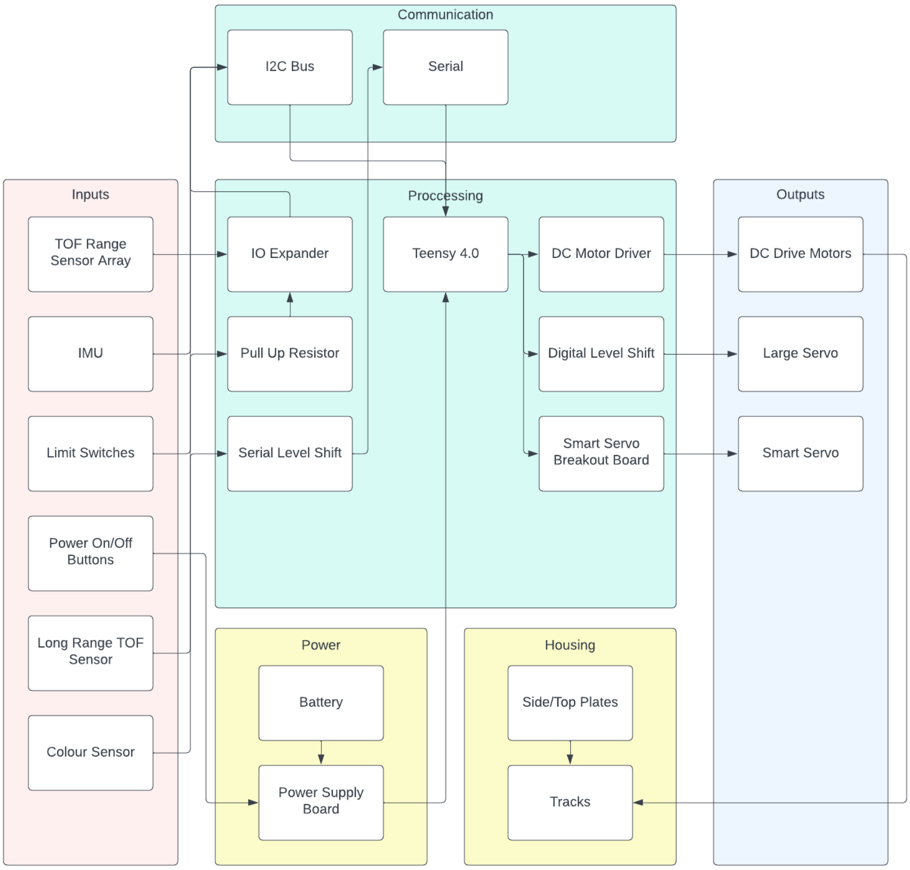


Figure 2: The functional architecture of the system.

Chassis

The chassis of the robot is comprised of the supplied side panels and cross bars. The focus of the chassis design was to maximise the internal space available, which contains the pickup and storage systems. The supporting cross bars are spread throughout the chassis providing stiffness and useful locations to mount sensors, servos, or other components. The current design also allows for the use of the supplied top plate, providing space for the attachment of power and control circuitry as well as storage for batteries.

Pickup Mechanism

The pickup mechanism is an essential part of the robot's functionality. The designed mechanism uses several stages to position a weight and lift it up into the robot where it can be securely stored. The flow diagram of the pickup mechanism can be seen in Figure 3. The initial design concept differentiated between plastic and metal weights. After further thought, it was decided that all weights should be picked up to reduce complexity and increase the number of possible points that can be gained.

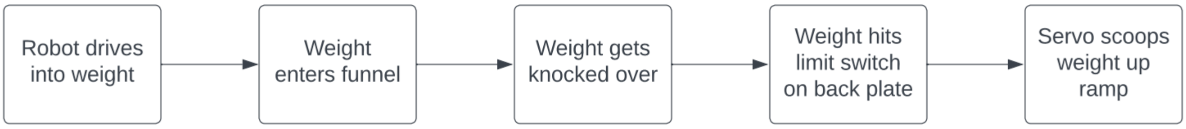


Figure 3: Flow chart showing the distinct stages of pickup mechanism.

A funnel was designed and mounted to the front end of the robot. This funnel stretches the entire front face of the robot to maximise the collection area. The funnel was designed such that weights would be knocked over and in a known orientation when passed within the robot. Multiple iterations were designed and tested in order to find a solution that correctly guided the weights into the correct location. This geometry can be seen in Figure 4.

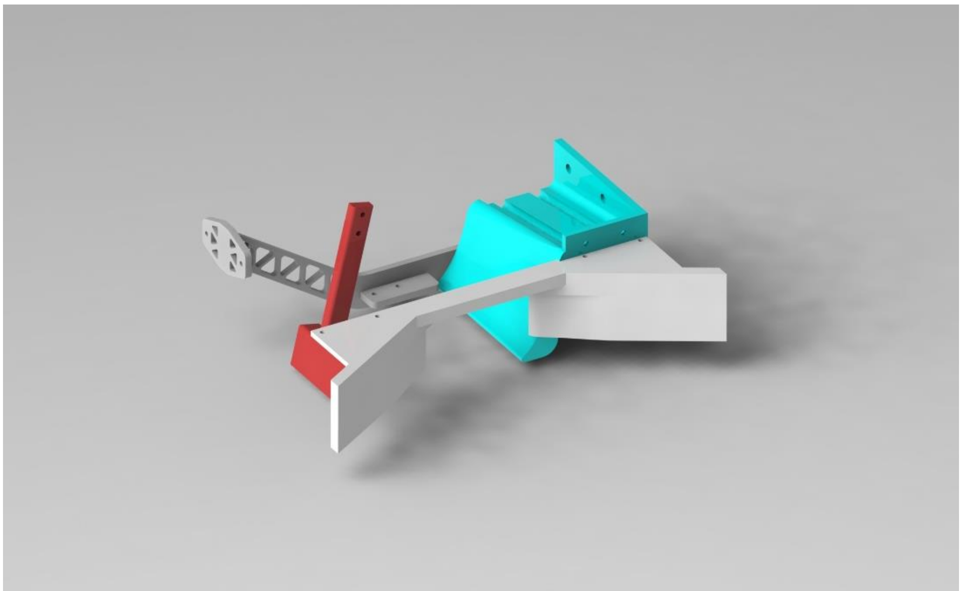


Figure 4: Pickup mechanism with funnel

The initial pickup mechanism design outlined a scoop aligned with the motion of the robot. This scoop would wait until the weight had reached a certain point and come from above, pushing the weight up and onto a sloped platform. This concept was replaced by a scoop mounted to one side of the robot and aligned adjacent to the movement. The current scoop design can be seen in Figure 5. This change allowed for the design to be more compact and for multiple weights to be picked up in quick succession.

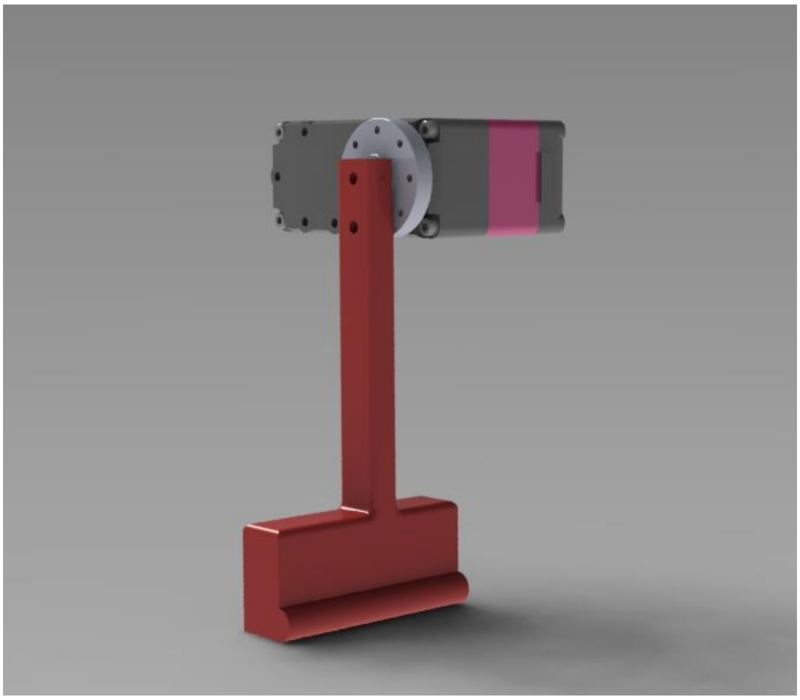
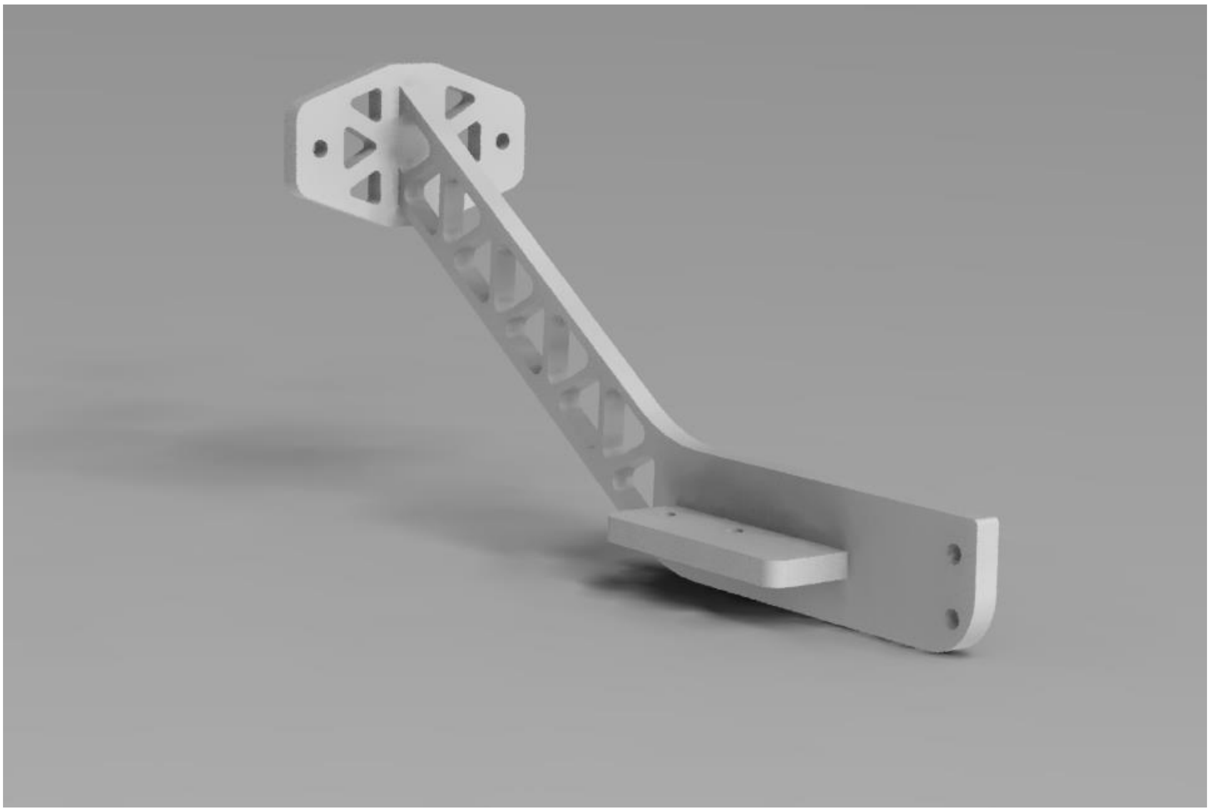


Figure 5: The scoop used in the pickup mechanism.

It was identified that a backplate was needed to ensure that weights that had passed through the funnel were contained and able to be picked up. This part needed to extend from the base of the ramp to the other side of the robot. This assembly would be laser cut out of acrylic due to the ease of manufacture and the tight tolerances possible withIt was decided that this manufacturing method.assembly would be laser cut out of acrylic due to the ease of manufacture and the tight tolerances possible. The assembly consisted of three parts:

- A bracket to attach the back plate to the side of the robot.
- A piece that attaches to the back of the ramp and spans the robot to the bracket.
- A stopper that slots into place on the previous piece.

A picture of the final assembly can be seen below in Figure 6. As the weight passes into the pickup mechanism it will hit the stopper and begin to move with the robot. This allows the weight to be picked up. The bracket is attached to the side plate using bolts with the rest of the assembly slotting into it. Due to the nature of slotting parts together, the bracket had to be manufactured accurately. The technical drawing for this part can be seen in Appendix C. Some detail was omitted from the drawing as several of the cuts were used to reduce weight. This is not essential to the function of the part and so was not dimensioned to improve the clarity of the drawing.



A picture of the final assembly can be seen belove in Figure X. As the weight passes into the mechanism it will hit the stopper and begin to move with the robot. This allows the weight to be picked up. The bracket is attached to the side plate using bolts with the rest of the assembly slotting into it. Due to the nature of slotting parts together, the bracket had to be manufactured accurately. The technical drawing for this part can be seen in Appendix X. Some detail was omitted from the drawing as several of the cuts were used to reduce weight. This is not essential to the function of the part and so were not dimensioned to improve the clarity of the drawing.

Figure 6: The backplate subassembly for the pickup mechanism.

The servo mount was created with 3D printed PLA and mounted to an open beam aluminium profile on the main chassis. The purpose of creating the mount was to ensure the stability of the servo. This was necessary as the scoop was subjected to a relatively large torque of 1.08Nm. Without adequate stability, the pivot point of the scoop may vary and result in unpredictable behaviour. Figure 7 shows the incorporation of each element in the final pickup mechanism.

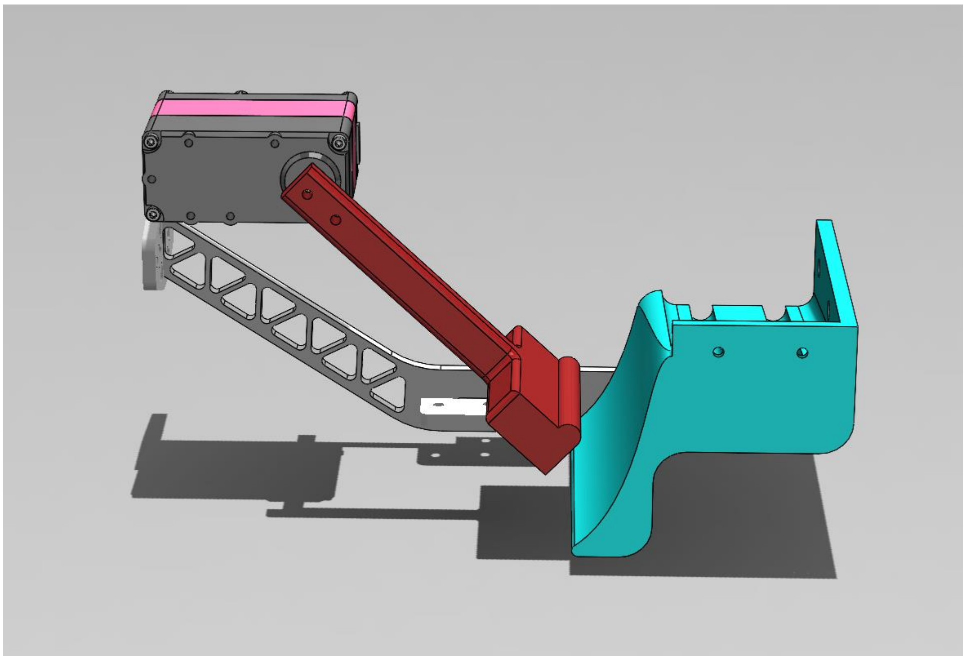


Figure 7: Complete Pick-up Mechanism

Navigation

One major aspect of the robot's design is that it must be able to navigate an arena autonomously. To achieve this a combination of wall following and obstacle detection was used. Data for this was provided by by utilising an array of TOF range sensors. A finite-state machine was used to control what the robot was doing at a given time. This is comprised of six different states with different criteria to swap between them. The finite state machine can be seen in Figure 8 below.

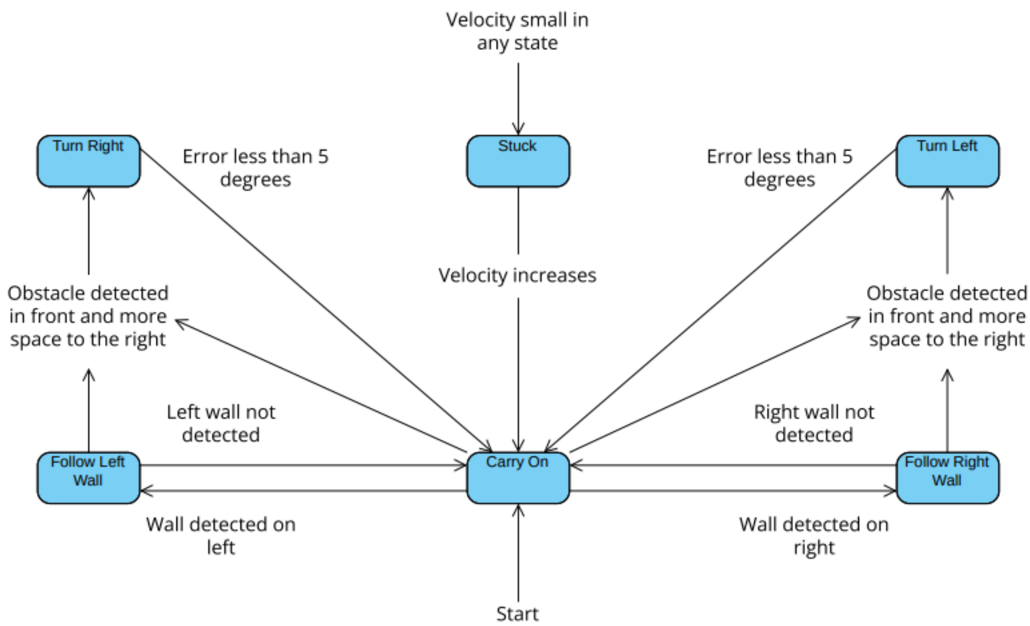


Figure 8: A finite-state machine of the navigation algorithm.

The robot used an array of five VL53L0X/VL53L1X range sensors. These are arranged around the robot with one facing forwards, two directly sideways, and two at a 45-degree angle. These are arranged around the robot with one facing forward, two directly sideways, and two at a 45-degree angle forward. The three sensors facing forwards were used for obstacle direction while the others were used for wall following. Due to the nature of the TOF sensors, they have no appreciable beam angle and so only detect objects directly in line with the sensors. The beam pattern of the array can be seen below in Figure 9.

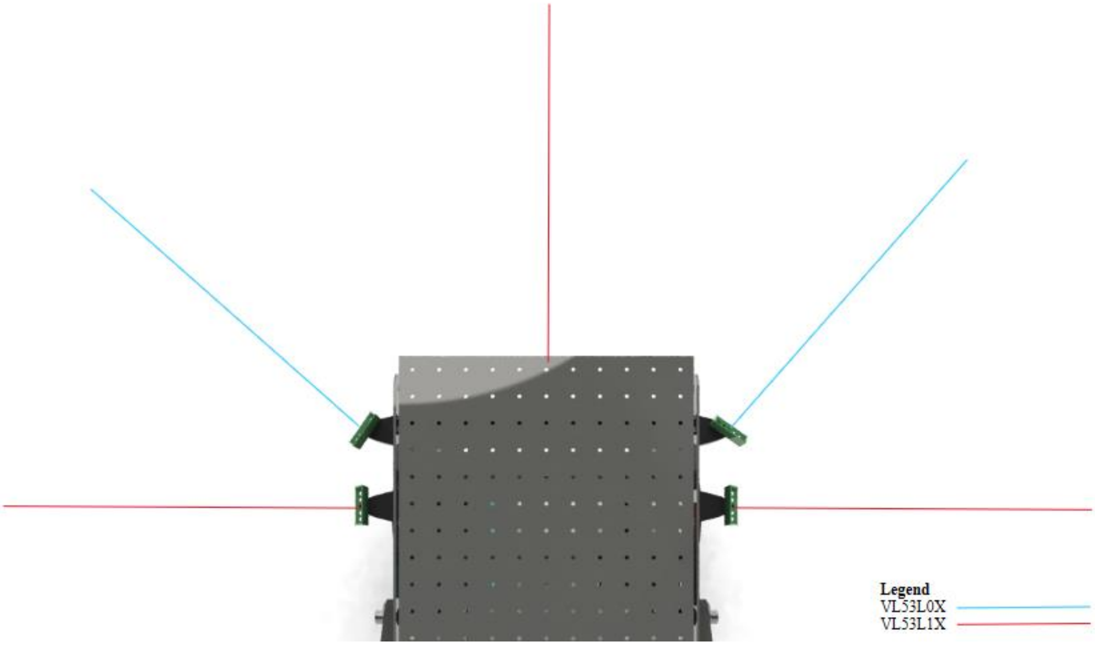


Figure 9: TOF sensor array beam pattern.

The robot uses the two sideways sensors to enable wall following. This is an important part of our navigation technique as it enables the robot to travel to most parts of the arena consistently. The wall-following state is entered if a wall is detected within a certain distance of the robot. A proportional controller was made using a reference distance and the sensor reading. This control output was added to the reference angle in the movement controller. This caused slight changes in heading, making the robot follow a wall at a set distance.

The robot uses the two sideways sensors to enable wall following. This is an important part of our navigation technique as it enables the robot to travel to most parts of the arena consistently. The wall following state is entered if a wall is detected within a certain distance of the robot. A proportional controller was made using a reference distance and the sensor reading. This control output was added to the reference angle in the movement controller. This caused small changes in heading, making the robot follow a wall at a set distance.

Movement

Several requirements for the robot were dedicated to the movement of the robot. These include the following.

- The robot should have a ground clearance of greater than 25 mm.
- The robot shall be capable of climbing a 30% gradient with a maximum height of 100 mm.
- The robot should be able to travel at a minimum speed of 0.4 m/s.

To meet these requirements, an effective movement system needed to be developed. Tank style tracks were used to achieve this. They were powered by two DC motors around a series of bearings and 3D-printed rollers with an 80mm diameter. These rollers gave the robot better ground clearance as well as more torque to the ground. These rollers gave the robot a better ground clearance of 25mm as well as more torque to the ground. A picture of the system can be seen below in Figure 10.

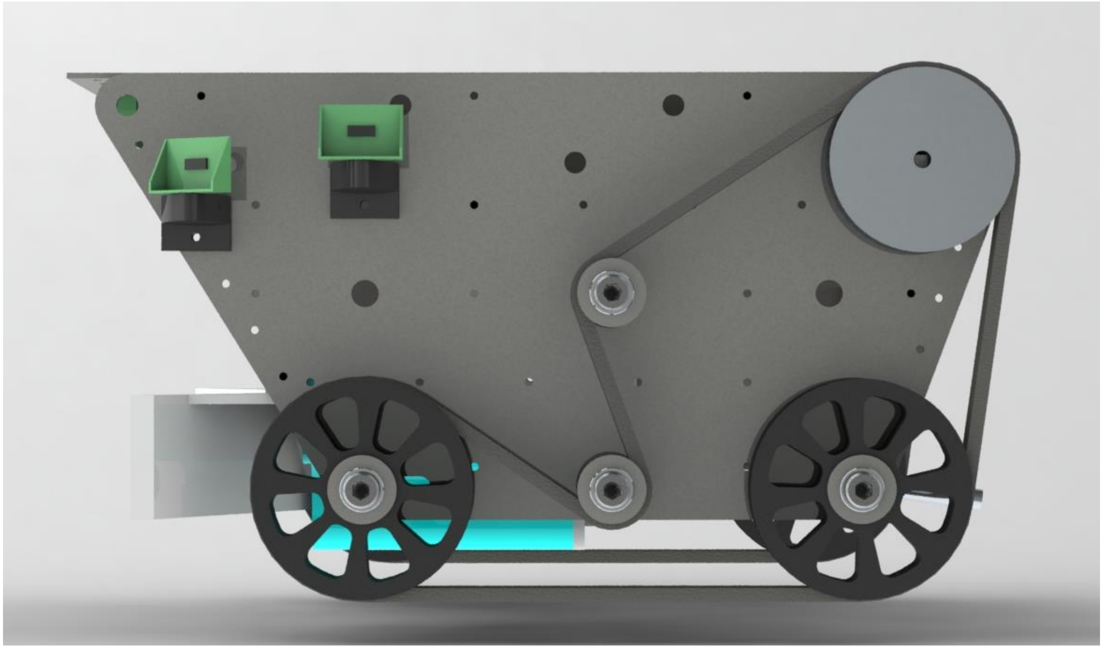


Figure 10: A CAD model of the drive system.

For the tracks to be effective they need to be correctly tensioned. The positioning of the bearings and rollers seen in Figure 10 allowed for the system to be easily tensioned by changing the position of the roller at the front of the robot. The rollers are a subsystem that consists of press-fit bearings that attach to the robot by a bolt. This allowed for the easy movement and installation of the movement system. An exploded view of the roller assembly can be seen in Figure 11.

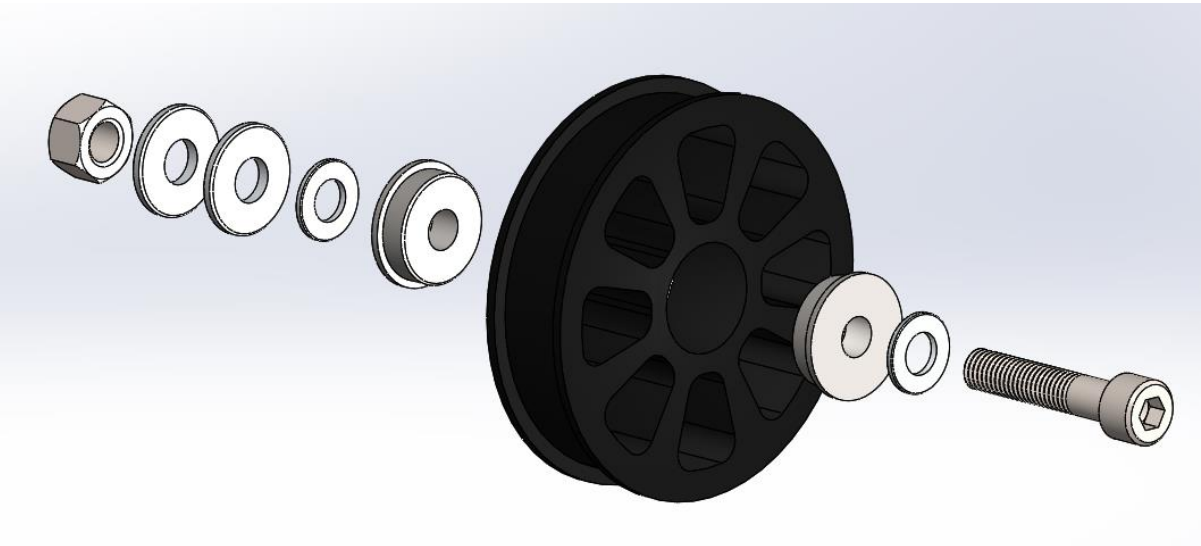


Figure 11: Exploded view of the drive track roller assembly.

The movement was controlled by two DC motors that received servo pulse signals from a microcontroller. The pulsed value could range from 1000 microseconds at full reverse to 2000 microseconds at full speed forward. The motion was controlled by a reference speed and angle. The angle control had feedback provided the IMU's orientation sensor. From this a proportional controller was implemented. The control value was added to the left-hand motor and subtracted from the right-hand motor. This allowed the robot to accurately turn simply by changing the reference angle. This also differences in the it still in a straight line at given heading. The reference speed was added to both motors so that it could drive forward and by changing this value.

Storage/Release

Once weights were collected and securely situated on the ramp, a method of storage had to be implemented. This must allow 3 weights to be kept inside the robot until it returns to its home base where they would be released. The initial design proposed was for the weights to slide down the left side of the robot from the existing ramp, before rolling across, from left to right, at the back of the robot. The rear of the robot would be covered by a vertical gate, latched with a solenoid that would release when the robot was at the home base. This initial design is visible in Figure 12.

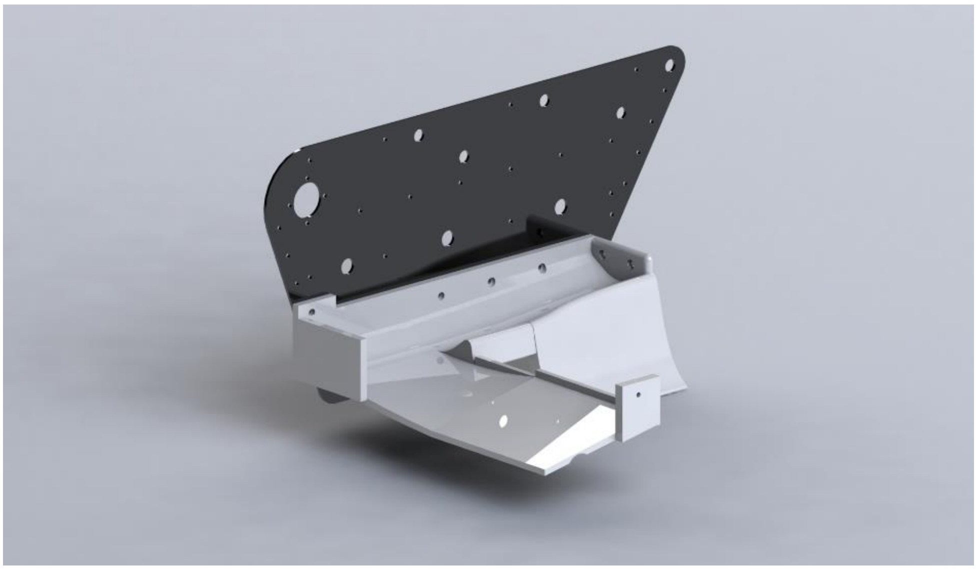


Figure 12: An initial design for weight storage.

This design had two major issues, firstly it required complex geometries to be 3D printed. These complex parts were weak and difficult to attach to others. Test fitting was completed with hot glue, but this was not practical for the final assembly. The second issue was the friction between the weight and the 3D printed components. Space constraints inside the chassis meant that getting a sufficient slope for weights to slide down was impossible without a complete redesign of the collection mechanism. The ramp was altered to try and allow weights to slide better, lubrication and sanding of the ramp were not effective, so a new design was required.

The next iteration of storage design was inspired by changing the material the weights slide on from plastic to metal. This resulted in a decrease of friction and thus the force needed for the weights to slide down. To achieve this the ramp from the collection method was altered to allow the insertion of two metal rods. These rods run down the left side of the robot, allowing 3 weights to be stored. The design of the new ramp and rod assembly can be seen in Figure 13.

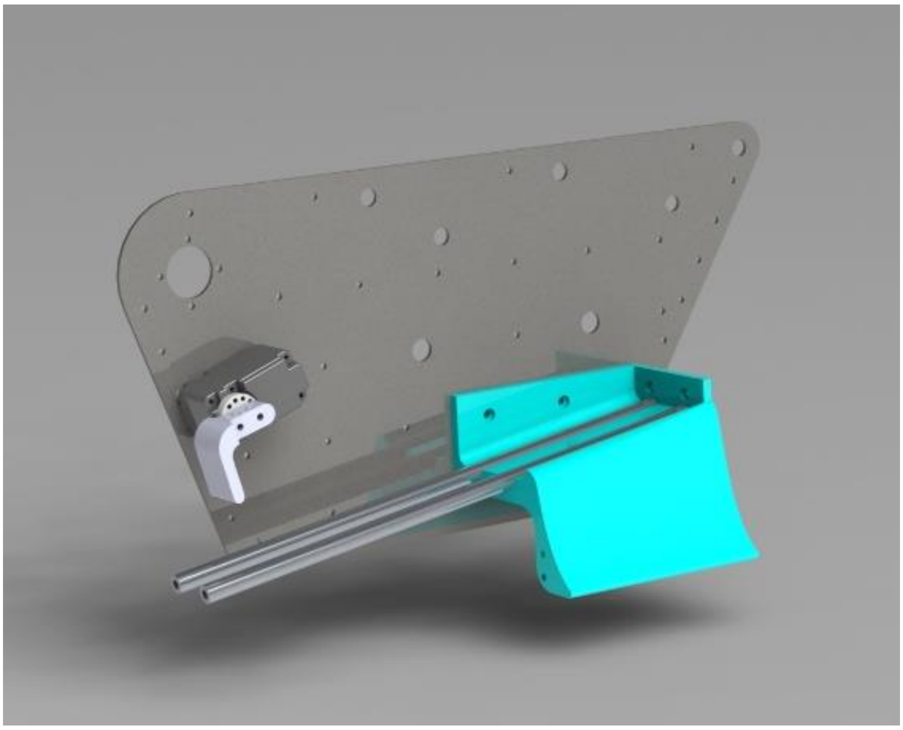


Figure 13: The current design for weight storage.

At the rear of the robot, the weights are kept on the rods through the use of a gate implemented with a Herkulex DRS-0101 smart servo. This servo sits at the rear of the robot, above the weights. When the colour sensor detects the home base the servo will turn on and release the weights. Engineering drawings of the gate attached to the servo are present in Appendix C.

Software

The robot is controlled by a microcontroller circuit using the Teensy 4.0. Software for this embedded system was developed using the Arduino IDE. The code uses a timer-based scheduler where each of the tasks can be registered and called at a constant frequency. This was achieved by using one of the inbuilt scheduler libraries on Arduino [1]. Four main tasks are completed by the code which can be seen in Figure 14. Task one controls the motors with a closed loop controller. Task two updates the navigation finite state machine. Task three updates all the sensor values. Task four calculates an odometer value and an estimated location. The frequency of each of these tasks was selected by looking at how long each of the tasks took. A period significantly larger than that was chosen to ensure that each task had enough time to finish in each cycle.

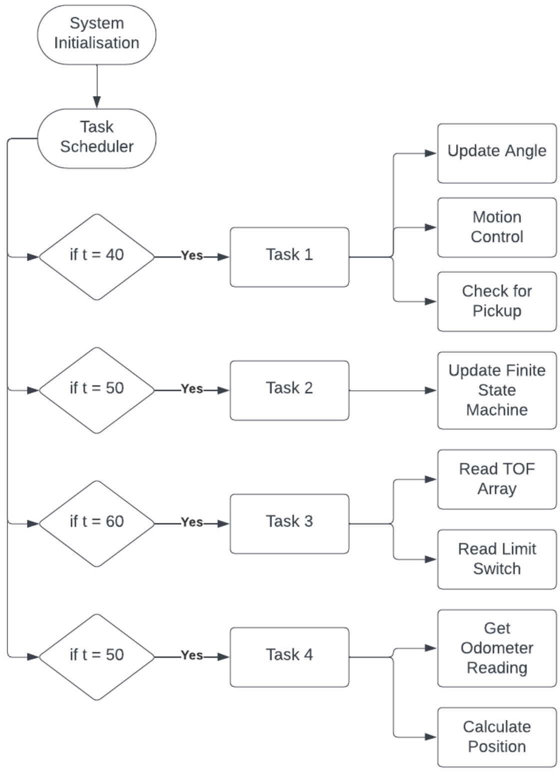


Figure 14: The task scheduler flow diagram for the robot.

The functions of tasks one and two have been explained in depth in previous sections. Task three updates sensor values including the array of VL53L0X/VL53L1X TOF sensors, the TOF serial mini sensor, and the limit switches. The array of TOF sensors is read through the I2C bus directly to the microcontroller. Settings for these sensors can be updated using their respective sensor libraries. This includes setting the timing budget and the range target. The readings of these sensors are filtered using a first-order autoregressive filter. The readings of these sensors are filtered using a first order autoregressive filter.

Task four calculates the approximate location of the robot within the arena relative to its home base. To achieve this an odometer reading and a heading reading are needed. The heading was taken directly from the IMU and is accurate as it is the result of a fusion of multiple sensors. The odometer reading was taken from the TOF serial mini sensor. This was done by comparing two readings at a set period. This difference was used to increment a value that represented the total distance travelled. The odometer reading could be combined with the heading using equations 1 and 2 to get an estimated location.

$$X_t = X_{t-1} + (Odometer_t - Odometer_{t-1}) \cos(\theta_t) \quad \text{Equation 1}$$

$$Y_t = Y_{t-1} - (Odometer_t - Odometer_{t-1}) \sin(\theta_t) \quad \text{Equation 2}$$

This provided a reasonable estimate of the robot's location as it navigated through the arena. This method assumes that the robot turns on the spot and only moves in straight lines between each function call. If the robot makes multiple turns while moving the estimate starts to drift from its true value. Further digital processing of the odometer reading may improve this drift. The path estimates from one of the test-runs around the arena can be seen below in Figure 15. This path started at one of the bases and travelled around several obstacles. This was achieved using manual control of the robot from a laptop instead of the autonomous navigation control system.

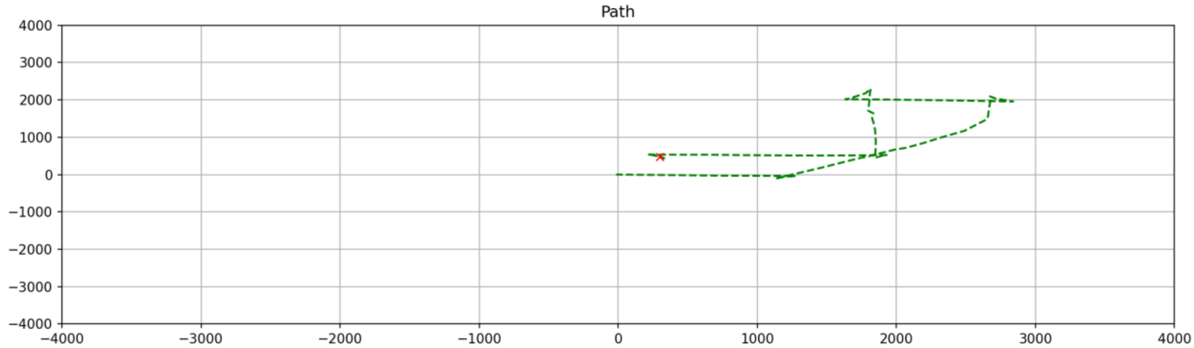


Figure 15: A path estimation of the robot.

Results

Pickup

The pickup mechanism was tested by a series of desktop tests. The tests involved placing each of the four different weights into the collection mechanism ten times and measuring how many times the pickup was successful. These tests measured the functionality of the mechanism in isolation. The results of these tests can be seen in Table 1.

Table 1: Desktop Tests for the Pickup Mechanism.

	Plastic Weight	0.5 Kg Weight	0.75 Kg Weight	1 Kg Weight
Number of successful pickups	8 of 10	9 of 10	10 of 10	10 of 10
Percentage success	80.00%	90.00%	100.00%	100.00%
Total Average Success Rate	92.50%			

These tests allowed us to determine any weak points of the system and unexpected behaviour. During testing, the scoop would get stuck in the up position if the limit switch was activated again before it returned to its original position. This was due to the way that the code had been written, this bug has since been overcome. When this occurred during testing the controller was reset and the tests were continued.

The movement of the scoop was chosen to decrease the maximum time between pickups. Table 2 presents specific results from testing, recording the average time interval between the time the servo motor initially engages and the time when the mechanism clears for subsequent pickups. Ten tests for each weight were undertaken.

Table 2: Timed Tests for the Pickup Mechanism

	Plastic Weight	0.5 Kg Weight	0.75 Kg Weight	1 Kg Weight	Average
Time (s)	8.1	8.4	8.6	9.0	8.5
Variance	16.5	17.6	10.7	14.2	14.75

The variance of these tests is very high, at approximately 14.75. For the pickup mechanism to completely empty, the robot needs to move so that the vibrations cause the weight to move down the ramp. This makes up majority of the pickup time and is highly dependent on the sensor input. The variance is an incredibly useful measure in determining the minimum time between pickups set in the code [3]. This has not yet been utilised but will be in future.

Navigation

The navigation can be tested by observing the robot's behaviour through different arenas. The arena has a total dimension of 2.4 x 4.9 m. This can be divided into a 4 x 8 grid of rectangles. The number of sectors that the robot enters during a test can be used to calculate an approximate percentage of the arena covered. The number of collisions, battery voltage used, and times the robot entered the stuck state were also recorded for each test. The results from the tests can be seen in Table 3. Each test only went on for 2 minutes to simulate a competition round. The three different arenas used can be seen in Figures 16.1, 16.2, and 16.3 below. The different arenas consist of an open arena with no obstacles, an arena with some obstacles, and an arena divided into two parts with a 400 mm path in the middle.

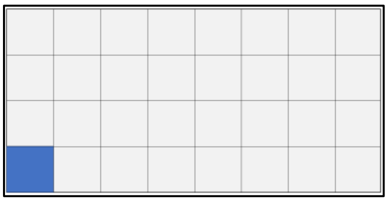


Figure 16.1: Open arena.

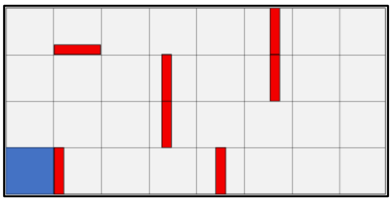


Figure 16.2: Obstacle Arena

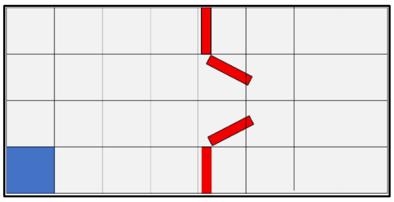


Figure 16.3: Divided Arena

Table 3: Results of the navigation algorithm testing.

	Open Arena	Obstacle Arena	Divided Arena
Number of Sectors Entered	20	19	17
Percentage Arena Covered [%]	62.6	59.3	53.1
Number of Collisions	0	4	0
Number of times stuck state entered	0	2	0
Battery Voltage used [V]	0.11	0.05	0.04
Battery Percent used [%]	0.89	0.41	0.32

In the open arena, the robot followed the wall around the outside of the arena without entering any of the interior sectors. This is due to the wall-following nature of the navigation algorithm. In the obstacle arena, the robot navigated around the first half of the arena well, occasionally colliding with obstacles when they entered the sensors' blind spots. When this happened, the robot entered the stuck state reliably and freed itself by driving backward and turning. It did not enter the sectors to the right-hand side of the arena due to the direction it would turn when it encountered an obstacle. For the divided arena, the robot did not follow the sharp turn in the wall into the second part of the arena.

Movement

The speed of the robot is one key design feature of the robot. To test for this the robot was timed moving five metres. This was done at full speed forwards and backward. This test was repeated three times in order to get the average speed the robot could travel. The results from this test can be seen in Table 4 below.

Table 4: Average speed of the robot.

	Full Speed Forwards	Full Speed Backwards
Run 1 [s]	14.9	13.8
Run 2 [s]	14.3	13.7
Run 3 [s]	14.3	14.2
Average Time [s]	14.5	13.9
Speed [m/s]	0.35	0.36

Software

The software utilises a timer scheduler with four tasks in total. Profiling data from the four tasks was collected by using the inbuilt 'micros ()' function at the start and end of each task and then printing this data to the serial port. The 'micros ()' function takes 106 clock cycles and printing to the serial port takes 139 clock cycles [2]. Accounting for this extra time the maximum period for each task can be found. Using this data, the CPU load can be estimated as displayed in Table 5.

Table 5: Timing data for the current program

Tasks	T_k	$T_{desired}$	$F_{desired}$	L_k
Task 1	$2\,ms$	$40\,ms$	$25\,Hz$	0.08
Task 2	$9\,\mu s$	$50\,ms$	$20\,Hz$	$1.8e^{-4}$
Task 3	$3.3\,ms$	$60\,ms$	$16.667\,Hz$	0.055
Task 4	$2\,\mu s$	$50\,ms$	$20\,Hz$	$4e^{-5}$

These calculations gave a total CPU load of 0.135, this value is significantly less than the maximum allowed load of 0.69 with rate monotonic priority scheduling implemented.

Fault Tree Analysis

The fault tree in Appendix B, shed light on many potential issues that would have otherwise not been identified. The most prominent being that if a weight were to get stuck in the collection mechanism, the robot would not be able to pick up any more weights for the rest of the round. This issue was present during the pickup method testing and resulted in the failed pickups recorded. The failure occurred when the collection servo picked up plastic weights especially, the weights could get wedged in the collection area, resulting in a jam of the scoop/ramp system. If the fault tree were drawn up earlier, this potential issue would have been identified.

There is a section outlining issues causing the robot unable to reach weights in the FTA. Examples of these issues would be if the weight is in a corner, or if the robot isn't moving. The fault tree analysis was invaluable in identifying the causal factors of the robot not reaching weights. Due to the shape of the robot, and its need to physically move over the weight to pick it up, these issues can be resolved in code. A Solution would be to identify when the robot was not moving, and when walls were close to both 45° sensors these factors would signal what issue was arising. This would imply the robot is unable to reach the weight and that specific measures need to be taken.

Opponent interference is another likely factor that may cause issues when competing against other robots. Before the competition, the robot should be thoroughly tested in the arena with other robots. If there are any concerns about the tracks being jammed or components being damaged by opponents, the mechanical design of these components may need to be altered. Care could also be taken in software to ensure the robot can handle another object moving around the arena.

Evaluation/Future Improvements

Chassis

The chassis of the robot is fit for purpose and meets the goals of maximising internal space and providing mounting points for any type of componentry. No further improvements will likely be made to the chassis before the competition. Assembly of the chassis will just be checked for stiffness before the robot competes, with bolts being tightened as needed.

Pickup Mechanism

In its current state, the pickup mechanism works the majority of the time. Heavy weights were picked up without issue, but lighter weights were occasionally hit to the side and jammed the mechanism. This is problematic as a jammed pickup mechanism means no further weights can be stored or dropped off. A method of preventing this may be to "randomly" move the servo around for a period in hopes that the mechanism will release the trapped weight.

During testing, it also became apparent that if weights were picked up in relatively quick succession, the device would jam and become unusable. This was somewhat mitigated within the code but would benefit from preventative mechanisms. The issue stems from the weights only sliding down the ramp with external input from the movement of the robot. Modifying the ramp angle, lubricating the ramp, or adding a small, secondary actuator to overcome the static friction may fix this issue.

Navigation

The navigation algorithm currently focusesfocusses on following wallswails and making turns when an obstacle is directly in front of it. Some issues are present with the current design that could be fixed with future development. One issue is the percentage of area the robot can cover in two minutes. Results from the testing show that the robot will only cover approximately 60 % of the arena as it will circle in the same path in one section of the arena. This is not ideal as it leaves a large area of the arena untouched, reducing the possible number of weights the robot can pick up. Another issue with the navigation currently is the blind spots the robot has. If an obstacle is present at the front of the robot but off centred, then none of the front sensors will detect it. During testing this was the cause of the collisions in the obstacle arena. Currently, this is overcome using a speed check to see if the robot is stuck or not.

To improve the robot's navigation several changes could be made. To ensure a larger amount of area is covered, the reactive navigation method could be altered to a more proactive method. This could involve using the localisation functions to determine where the robot has not been yet. Using this information, a heading could be provided to the robot that it should aim for when making turns or not following a wall. The issue of having blind spots could be improved by adding more sensors, however, there is a significant timing budget associated with TOF sensors. So, the benefit of adding extra sensors compared to reducing blind spots will need to be tested.

Movement

The movement subsystem had three requirements it aimed to meet. Firstly, to have at minimum 25mm of ground clearance which has been achieved through the use of 80mm diameter rollers, giving the robot 25mm of ground clearance once tracks are in place. The implemented track arrangement works well but tends to slip over time as nuts come loose through vibration. However, this is easily fixed through tensioning of the track. The second requirement was for the robot to be capable of climbing a 30% gradient, this has also been achieved with the main issue being the robot successfully lining up with the ramp before it tries to climb the slope. The final requirement was for the robot to move with an average speed of 0.4 meters per second. The robot does not reach this threshold in either forward or reverse movement, reaching average speeds of 0.35 and 0.36 meters per second, respectively. However, this failure is by a narrow margin and testing shows the robot can cover much of the arena within 2 minutes at its current speed. Most improvements to movement will occur through refinement of the navigation algorithm in the time before the competition.

Storage/Release

The current iteration of the storage system meets the criteria set out in the functional requirements as it successfully stores three weights in the robot. Testing shows that the storage and release system integrates well with the pickup mechanism. All weights stored can be deposited at the home base once three weights are collected. The latch is triggered when the colour sensor detects the same colour it started on. Further improvements would be to work on a more permanent attachment system for inserting metal rods into the pickup ramp. Currently, the rods are press-fit, this works well in the short term but is not reliable for the continual use of the robot. Some solutions might include glue, or the use of a grub screw to keep the rods in place. The release latch could also be improved by a different servo mount. This would allow for a better attachment between the latch and servo. The current latch is bolted straight into the servo rather than a mount and the bolt is prone to slipping. A single bolt design is also not ideal when dealing with angular loads, such as will be experienced by the latch.

Software

The software currently utilises a timer scheduler with a very low CPU load of 0.135. This leaves significant room for the performance of the code to be increased. This could involve the increased performance of the controllers being used to drive the movement of the robot. Currently, only proportional control is being used; however, more complex control could be implemented to increase its performance. Another improvement that could be made is to the settings that the TOF array currently uses. This could provide more accurate readings, thus improving the navigation of the robot. Another method of doing this could be to introduce better signal processing into the software. Presently,Currently autoregressive filters are utilised. Better filtering methods could be used such as a moving average filter.

Requirements

In the initial stages of the design process, a specification of requirements was formed based on initial testing and competition rules. The current design has been evaluated against this set of requirements using a pass or fail metric. The details of this evaluation can be seen in Table 6X. Overall, this prototype is largely successful as it passes 18/23 of the initial requirements. The requirements that it failed were largely involved with higher functionality of the robot such as being able to return home and being able to distinguish between plastic and metal weights. This functionality has not yet been included in the design as it was deemed more important to get a basic robot working first.

Table 6: Evaluation of the current design against the initial specification of requirements.

Requirements	**Pass/Fail**
1.1 Functional	
1.1.1 The robot shall be able to pick up metal weights of differing mass (0.5, 0.75, and 1 kg).	Pass
1.1.2 The robot should be capable of storing three metal weights maximum.	Pass
1.1.3 The robot should distinguish between metal and plastic, dummy, weights.	Fail
1.1.4 The robot should have a mechanism to avoid capture of the bludger.	Pass
1.1.5 The robot shall have a width no larger than 350 mm.	Pass
1.1.6 The robot should have a ground clearance of greater than 25 mm.	Pass
1.1.7 The robot shall be capable of climbing a 30 % gradient with a maximum height of 100 mm	Pass
1.1.8 The robot should be able to distinguish between two base colours and store this information at the start of each round.	Pass
1.1.9 The robot should be capable of identifying and avoiding obstacles. These include the following: 1.1.9.1 Walls and cylinders. 1.1.9.2 The opponent robot.	Pass
1.1.10 The robot should be able to return to the home base and drop weights on arrival.	Fail

1.1.11 The robot shall not pick up weights in either base during the round.	Fail
1.2 Performance	
1.2.2 The robot should be able to travel at a minimum speed of 0.4 m/s when fully loaded with weights.	Fail
1.2.3 The robot should be able to identify and successfully pick up a weight within a time span of 25 seconds.	Fail
1.2.4 The pickup mechanism should have a success rate of greater than 90 percent during a round in competition.	Pass
1.2.5 The robot should be able to complete a full turn at any location in an arena.	Pass
1.3 Nonfunctional	
1.3.1 The robot shall be robust enough so that any collisions with an opponent robot will not cause permanent damage.	Pass
1.3.2 If a collision occurs, no components should be lost as a result.	Pass
1.3.3 Wiring should be secure enough to stay in place during a round.	Pass
1.3.4 The robot should be easy to assemble and disassemble within five minutes.	Pass
1.4 Constraints	
1.4.1 The robot should be made of supplied parts and bought parts within a $50 budget.	Pass
1.5 Operational	
1.5.1 The robot should not use a full battery within one round of the competition.	Pass
1.5.2 The robot should be able to have the same functionality independent of which side it starts from in the arena.	Pass
1.5.3 The robot should be easily repairable in between rounds within five minutes.	Pass

Contribution Statement

Henry:

I have contributed to different areas of the robot's design and development, as well as writing sections of this report. I have written the code that the microcontroller uses, including the movement, sensing, navigation, and main modules. I have spent time refining and testing the FSM used in the navigation algorithm. For the mechanical aspect of the robot, I have designed the back plate assembly, the ramp in the pickup mechanism, and the new movement system. In the report, I wrote multiple sections including the navigation, movement, back plate, and software design descriptions, the navigation and software evaluation, the navigation, pickup, and software results sections, the evaluation against requirements, and the technical drawing of the back plate, the executive summary, and the bill of materials.

William:

I have mainly worked on the physical design for the robot. I have designed and tested various iterations of the storage subsystem, I also designed all the mounts used for sensors on the robot and the servo actuated gate that controls weight release. As part of this I created an engineering drawing for the release gate as well as testing and writing code for the Herkulex smart servo, I worked to test the movement and navigation subsystems. For the report I wrote the introduction, design overview, storage/release and chassis descriptions and the evaluations for the chassis, movement, and storage/release subsystems.

Alister:

I focused on the pickup and collection side of the robot. I designed and tested the inlet funnel, the scoop servo attachment, and the servo housing. I also played a role thinking and developing ideas on how the software was to be implemented. I tested the pick-up mechanism and came to relevant conclusions. Regarding the report, I drew up the complete fault tree analysis and wrote sections on the pickup mechanism, evaluation, and future development.

References

[1] A. Arkhipenko, “Task Scheduler”, Arduino, https://github.com/arkhipenko/TaskScheduler

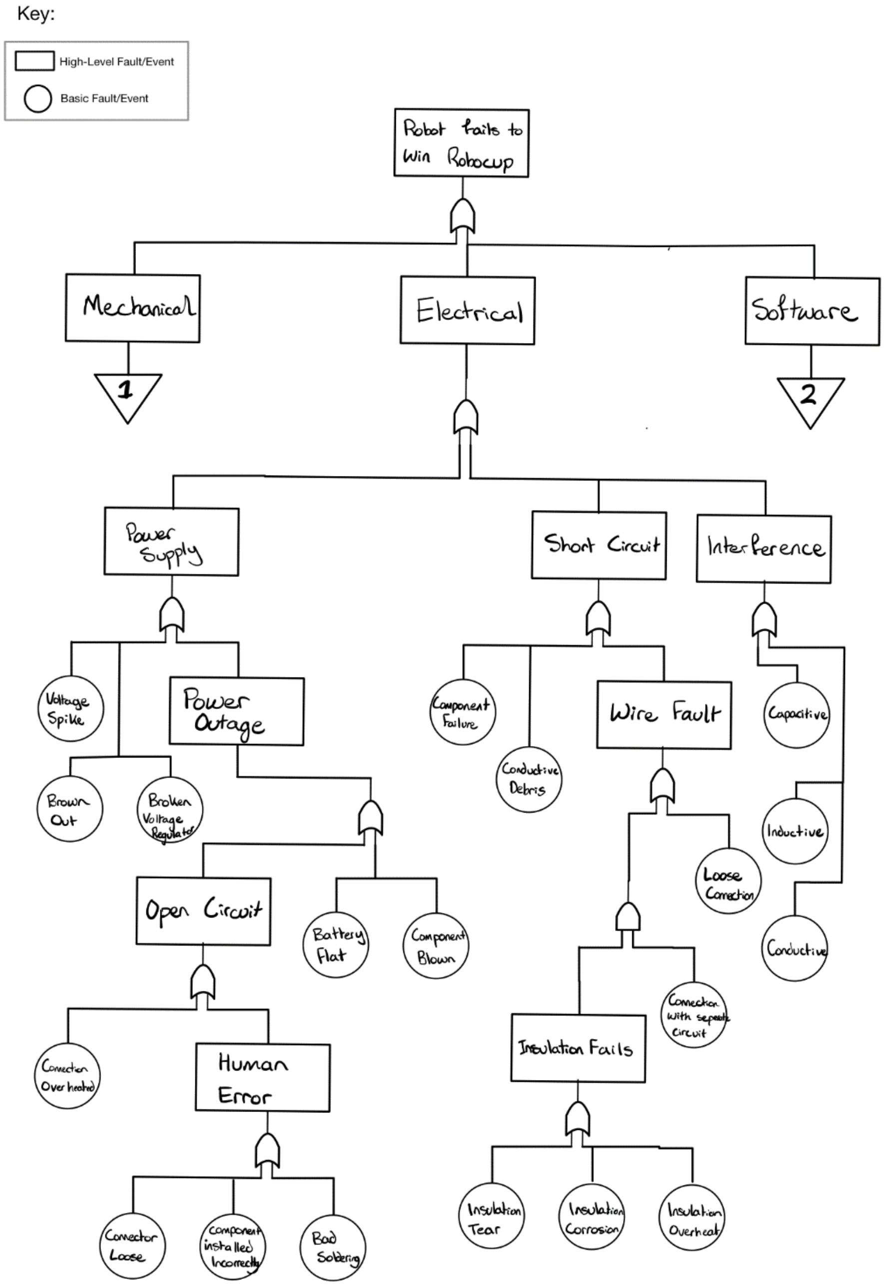
[2] Nebel, 2021, “Timing the Timekeeper”, Arduino Craft Corner, https://arduino-craft-corner.de/index.php/2021/10/20/timing-the-timekeeper/

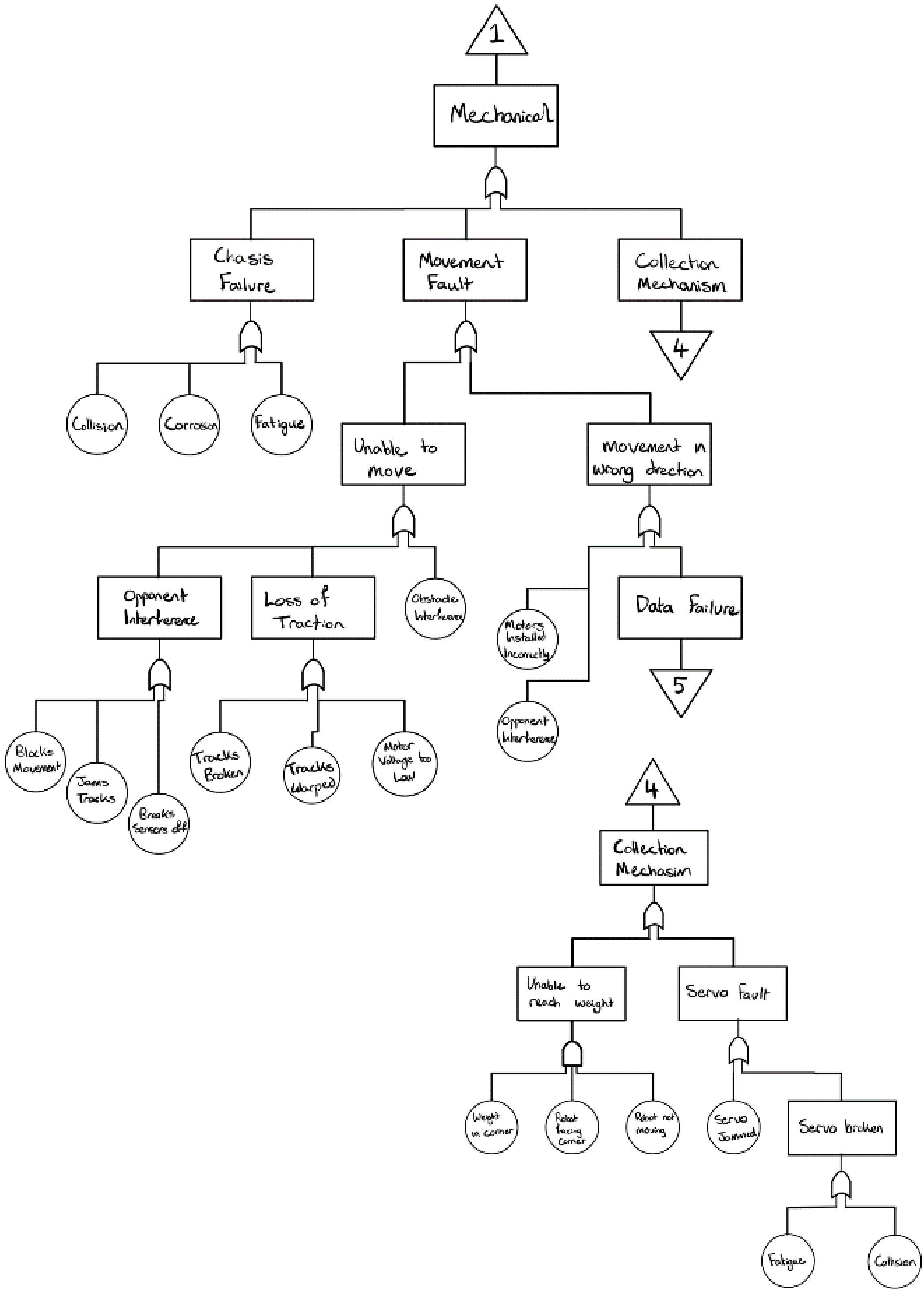
[3] Ryan Lober, Vincent Padois, Olivier Sigaud. Variance Modulated Task Prioritization in Whole-Body Control. 2015. ⟨hal-01180011⟩

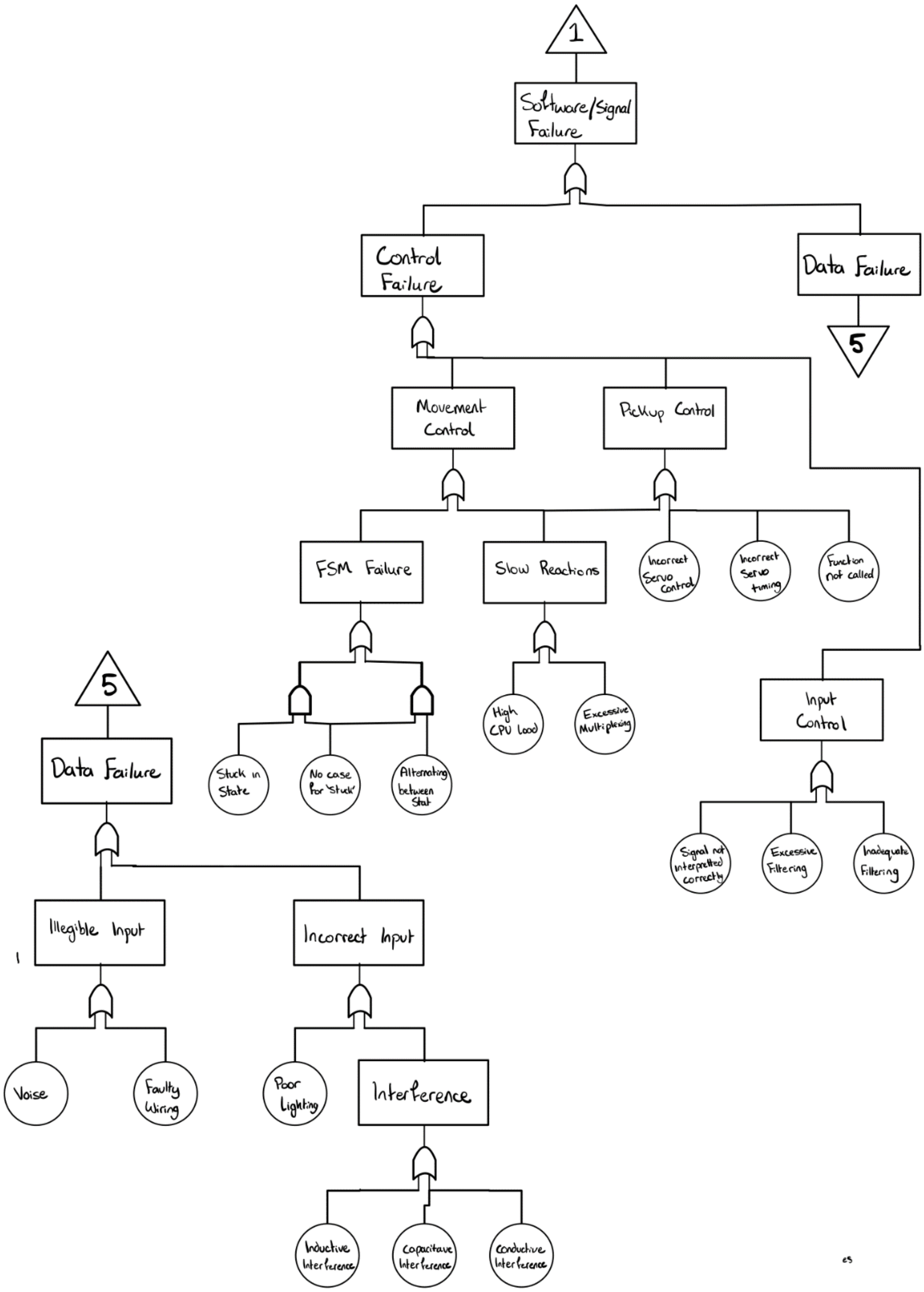
Appendix A – Bill of Materials

Part	Quantity	Material	Total Weight	Total Cost
Roller	4	3D Printed PLA	120 g	NA
Sensor Mount	4	3D Printed PLA	17.2 g	NA
Servo Mount	1	3D Printed PLA	27 g	NA
Scoop	1	3D Printed PLA	15 g	NA
Ramp	1	3D Printed PLA	69 g	NA
Funnel	1	3D Printed PLA	51 g	NA
Gate	1	3D Printed PLA	3.5 g	NA
Back Plate Assembly	1	Acrylic	21 g	NA
Robot Side plate	2	Aluminium Sheet metal	NA	NA
Robot Top plate	1	Aluminium Sheet metal	NA	NA
Drive Track Support Hardware	8	Mixed	NA	$24
Robot tracks – 880-8M	2	Rubber	NA	$100
Open beam aluminium profile	5	Aluminium	100 g	$25
8mm round bar	2	Aluminium	44 g	NA
DC Motor – 28PA51G	2	NA	NA	$140
Smart Servo – SG90	1	NA	NA	$58
Large Servo – RDS5160	1	NA	NA	$60
TOF I2C – VL53L0X	2	NA	NA	$10
TOF I2C - VL53L1X	3	NA	NA	$30
TOF Serial - TFmini	1	NA	NA	$50
Colour Sensor – TCS34725	1	NA	NA	$14
Limit Switch - SV-163-1C25	1	NA	NA	$0.5
Arduino Teensy	1	NA	NA	$60
Stop Go button	1	NA	NA	NA
Motor drive board	1	NA	NA	NA
Power supply board	1	NA	NA	NA
Digital level shift	1	NA	NA	NA
Smart servo breakout board	1	NA	NA	NA
Serial Level Shift	1	NA	NA	NA

Appendix B – FTA



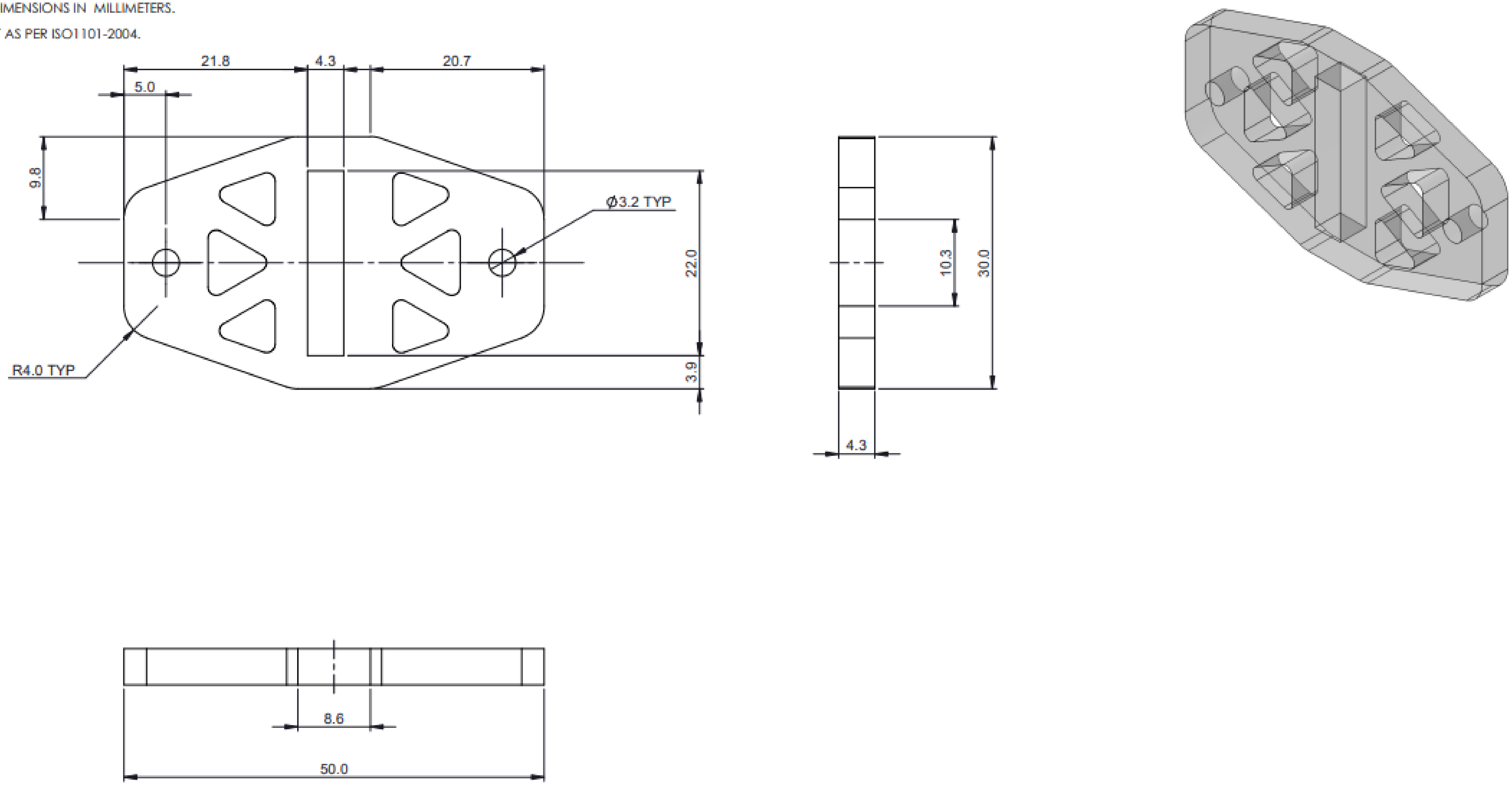

1
Mechanical
Chasis Failure
Movement Fault
Collection Mechanism
4
Collision
Corrosion
Fatigue
Unable to move
movement in wrong direction
Opponent Interference
Loss of Traction
Obstacle Interference
Motors Installed Incorrectly
Data Failure
5
Opponent Interference
Blocks Movement
Jams Tracks
Breaks Sensors off
Tracks Broken
Tracks Warped
Motor Voltage too Low
4
Collection Mechasim
Unable to reach weight
Servo Fault
Weight in corner
Robot facing corner
Robot not moving
Servo Jammed
Servo broken
Fatigue
Collision


1
Software/Signal Failure
Control Failure
Data Failure
5
Movement Control
Pickup Control
FSM Failure
Slow Reactions
Incorrect Servo control
Incorrect Servo timing
Function not called
Stuck in State
No case for 'Stuck'
Alternating between Stat
High CPU Load
Excessive Multiplexing
Input Control
Signal not interpretted correctly
Excessive Filtering
Inadequate Filtering
5
Data Failure
Illegible Input
Incorrect Input
Voise
Faulty Wiring
Poor Lighting
Interference
Inductive Interference
Capacitave Interference
Conductive Interference

Appendix C – Technical Drawings

NOTES - (UNLESS OTHERWISE SPECIFIED)

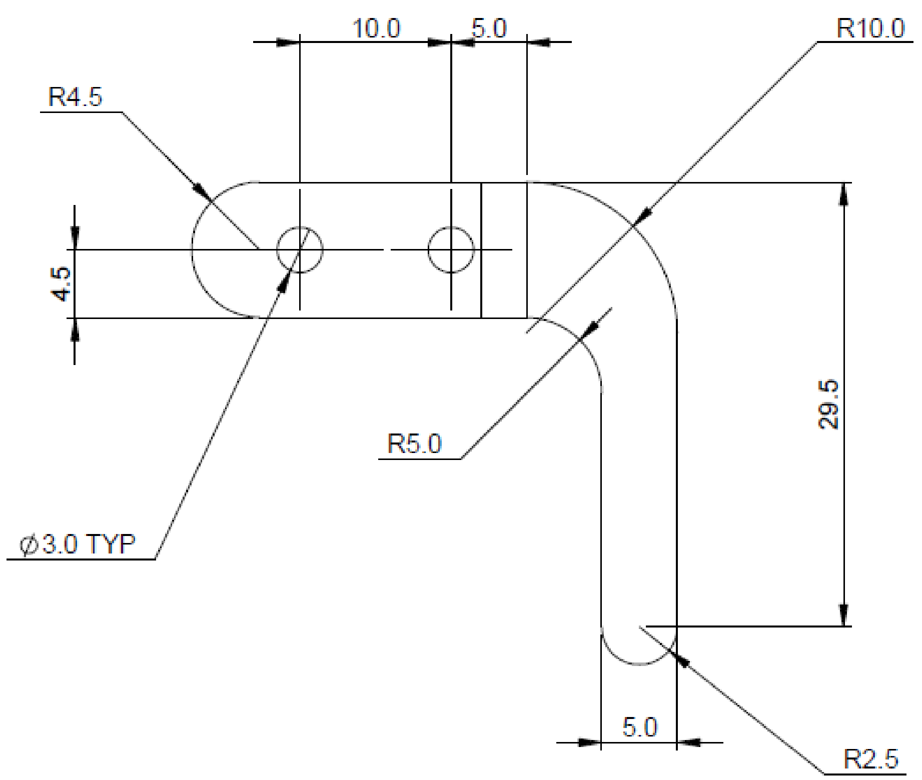
1. ALL DIMENSIONS IN MILLIMETERS.
2. GD&T AS PER ISO1101-2004.

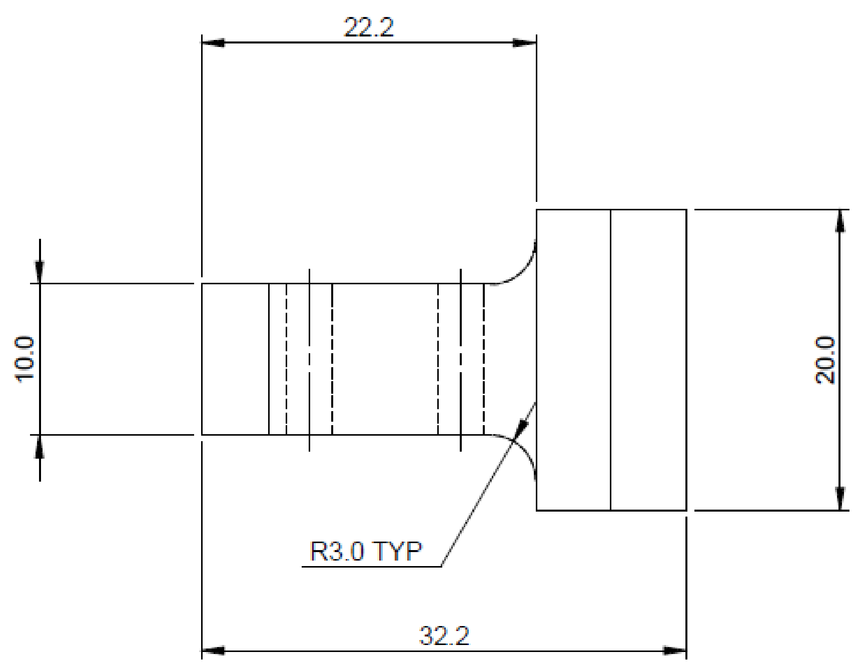


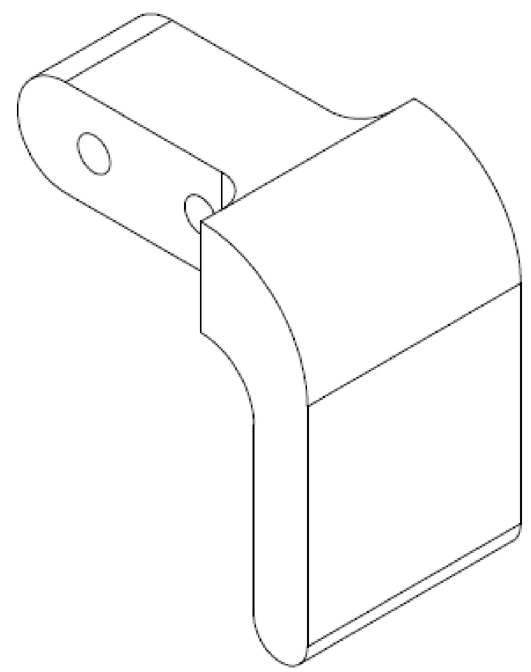
THIRD ANGLE PROJECTION	THIS DOCUMENT IS ISSUED IN STRICT CONFIDENCE ON CONDITION THAT IT IS NOT COPIED, REPRINTED, OR DISCLOSED TO A THIRD PARTY EITHER WHOLLY OR IN PART WITHOUT THE WRITTEN CONSENT OF UNIVERSITY OF CANTERBURY	UNIVERSITY OF CANTERBURY © DEPARTMENT OF MECHANICAL ENGINEERING	UC UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND
MATERIAL: ACRYLIC		Back Plate Bracket	
FINISH: POLISHED		DESIGN: H. HALL; DWG NO. BP Part 3	REV: REV1
TOLERANCE (UNLESS OTHERWISE SPECIFIED) DECIMAL mm		DRAWN: H. HALL; PROJECT: ROBOCUP	
.X ± .1 .XX ± .05 .XXX ± .010 ANG. ± 1°		SUPERVISOR: C. PRETTY; ISSUE DATE: 07SEP23	DRAWING NOT TO SCALE; SHEET 1 OF 1

NOTES - (UNLESS OTHERWISE SPECIFIED)

1. ALL DIMENSIONS IN MILLIMETERS.
2. GD&T AS PER ISO1101-2004.







THIRD ANGLE PROJECTION	THIS DOCUMENT IS ISSUED IN STRICT CONFIDENCE ON CONDITION THAT IT IS NOT COPIED, REPRINTED, OR DISCLOSED TO A THIRD PARTY EITHER WHOLLY OR IN PART WITHOUT THE WRITTEN CONSENT OF UNIVERSITY OF CANTERBURY	UNIVERSITY OF CANTERBURY © DEPARTMENT OF MECHANICAL ENGINEERING		UC UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND
MATERIAL PLA		Weight Storage Latch		
FINISH 3D-Print		DESIGN W. Thorpe	DWG NO. 1	REV REV1
TOLERANCE (UNLESS OTHERWISE SPECIFIED) DECIMAL mm		DRAWN W. Thorpe	PROJECT Robocup	
.X ± .1 .XX ± .05 .XXX ± .010 ANG. ± 1°		SUPERVISOR C. Pretty	ISSUE DATE 13SEPT23	DRAWING NOT TO SCALE — SHEET 1 OF 1