Warwick Mobile Robotics

Technical Report

WMR Search & Rescue 2010/11

4

Sponsored by

Contents

Table of figures

List of Tables

1 Introduction

This report provides more detailed information into the core areas of modification on the teleoperated Urban Search and Rescue robot (USAR-T) and autonomous Urban Search and Rescue robot (USAR-A). It details the design process for all of the modifications made to the sub-systems of both USAR-T and USAR-A robots.

A full SWOT analysis is conducted of the previous year's offerings before identifying the most crucial design elements for the year before discussing the intricacies of each modification. Descriptions are broken down into relevant sections and sub-sections. Each robot is divided into mechanical and electronic and software systems improvements. The software improvements can be considered the systems integration portion of the development process as it ties the hardware, electro-mechanical and electronic systems together to produce the functional platform.

Each design choice is fully justified with regards to the SWOT analysis before a full specification and design requirements are decided upon. After the design of each part, they are completely tested both on their own and within the context of the entire robot system. An evaluation of the performance of the design modifications is conducted and recommendations for next year are drawn from this year's experiences and flaws identified with new systems.

2 USAR-T, Teleoperated Robot

The teleoperated Urban Search and Rescue (USAR-T) robot, piloted manually by a human operator, is capable of tackling significantly more complex and unstable terrain than its autonomous counterpart; the autonomous Urban Search and Rescue (USAR-A). It is also capable of more complex actions such as manipulating objects and interacting in other ways with located victims. As a result of this more complex behaviour, the USAR-T requires additional hardware such as movable 'flipper' tracks and a robotic arm. It also requires more focus on the area of Human-Robot Interaction, allowing the operator to easily control the entire functionality of the robot.

2.1 Evaluation of the 2009/10 Robot

The 2009/10 USAR-T robot was very successful at the competition level. Its rigidity and mobility allowed it to find victims faster and in more hard to reach places than its competitors. In order to effectively determine the areas of this already successful platform, a SWOT analysis was carried out and is detailed in the following section.

2.1.1 SWOT Analysis of 2009/10 Platform

2.1.1.1 Strengths

The 2009/10 platform's success is indicative of its many strengths, particularly its mobility. The chassis and track design is very effective, allowing it to deal with most of the obstacles that it came across in the competition.

As a first design the original arm was quite successful, with ample length, strength and degrees of freedom to enable the operator to move the head to wherever was necessary. The last-minute hook 'gripper' on the end was clearly visible to the operator ensuring picking up objects was relatively easy.

The head itself was a good place to house the sensors and cameras to ensure maximum visibility and allow them to be positioned close to objects for more detailed inspection. The folded aluminium and rapidly-prototyped plastic provided sufficient structure to hold and protect the components without adding significant weight to the end of the arm.

The current robot software has been proven to work over a number of years and has been continually improved upon since the first competition WMR participated in. It provides all the basic functionality required by the operator to manoeuvre the robot and monitor the sensor data. The underlying software that handles the authentication and communication between the client and robot also functions well and is arguably the best feature of the current client software.

The stack's design is compact and very well ventilated, based off a previous, proven, design from Remotec. Wires and connections between components are readily accessible by removing the stack from the chassis.

2.1.1.2 Weaknesses

Testing and examination of the 2009/10 USAR-T robot identified several key areas of weakness. The main mechanical area of weakness was the arm. There was significant looseness or 'play' in the joints, magnified by the length of the arm to result in free vertical head movement of almost 100mm. Potentiometers used for positional feedback were unprotected and vulnerable to contamination which could cause severe damage to the arm and its motors (detailed further in section 2.2.1.2).

The gripper was last-minute and temporary addition to the robot, only able to hook appropriately shaped objects and not at all able to manipulate, for example, bottles of water. It was on a long extension in front of the robot head which could cause problems when trying to investigate victims more closely. It was also, because of its length, is difficult to correctly orientate and too unstable to effectively hold objects.

Locating the head at the end of the arm means that there are a lot of exposed wires running down its length that, if caught or broken, could result in the loss of vital sensors such as the camera feeds.

The plates used to hold the 'flipper' motors in place had distorted in the chassis, as a result of not being sufficiently strong enough to bear the load generated by the flippers.

Although the software was not lacking in functionality, it made no effort to reduce the load on the operator, requiring knowledge of a number of commands and the understanding of the software so problems could be fixed during operation. The software could be unreliable, dealing poorly with disconnections during periods of weak Wi-Fi signal. The organisation of the code makes development incredibly difficult, neglecting to implement paradigms such as object orientation or the model-view-controller architecture that would allow for the logical structuring and separation of different elements of code with different functionality. There was also a large amount of redundant code left from previous years which may not even be ran by the software, adding to the confusion by subsequent teams when they attempt to

study what was already there. The interface itself does not provide a good user experience. There was no real cohesive feel between the different elements that, although does not impact upon functionality, makes quickly assessing the situation of the robot more difficult that it needs to be.

Interconnection between components caused a large number of wires to appear loose and disjointed, despite being connected to the stack though cable ties. Also, the stack had to be removed by lifting the edges of the top stack, making insertion and removal periodically problematic.

2.1.1.3 Opportunities

The initial arm design was a great starting point for the USAR-T, identifying some of the issues arising from that particular arm configuration. Decreasing the play in the joints and improving rigidity of the arm members will greatly increase the arms effectiveness. Larger diameter arm members would not only improve rigidity but would also allow wiring to be internalised to help protect it. Larger joints would also help by protecting the delicate potentiometers and increasing the structural strength of arm.

The use of pneumatic / electro-mechanical gripper would increase manipulator capabilities markedly, allowing the robot to demonstrate more effective manipulation abilities. This can be achieved either through a bespoke design, or an appropriate 'off-the-shelf' gripper. At the competition, a "Best in manipulation" title is available which this gripper would allow the robot to compete for.

Moving the router or an aerial extension into the arm or head and away from the electromagnetic interference of the motors and metal chassis would improve the wireless communication to the robot reducing the occasional wireless dropouts noticed in the 2009/10 robot.

The biggest opportunity in terms of software is a mass restructuring of the code and possibly re-writing some elements. Because the software already works and has been used for years, it is clearly unnecessary to re-write a large proportion of the underlying functionality. This effort would benefit future teams and the process of reviewing the code would possibly highlight areas that may be improved. There is also a great area of opportunity in improving the reliability of the software during usage by implementing fail-safes to handle many of the frequently encountered errors.

There are also opportunities to extend the functionality of the client, adding two-way communication possibly and a notification system to alert the driver. Another opportunity lies in embedding the webcam streams directly into the client application rather than having to have open two extra program windows as is currently the case.

The inclusion of a bracket would contain the wires tied to the side of the stack and the addition of handles would allow for a much easier removal and re-insertion of the stack.

2.1.1.4 Threats

Alongside making changes to the robot, there are also various threats that need to be considered. Increasing the size and strength of the arm will increase the weight of the arm and subsequent load on the chassis and motors. If additional degrees of freedom are considered, the complexity of the arm may require software changes, particularly to the inverse kinematic control. Positioning the router in the head or arm will make it more exposed and vulnerable to damage than when it was inside the body.

The greatest risk for all manufactured components is the reliance on external manufacture and subsequent delays in that. Manufacturing a new arm due to its size and relative complexity will take a considerable amount of time.

The already-functioning code can also be seen as a threat as any modification to it may cause this foundations such as the existing communication systems in place between the client and robot to break. There is also the threat of wasting effort in only improving the visual aspects of the system. It is important to not only improve the usability of the software but also improve the control systems in place to ensure the improvements are not purely aesthetic.

Unintentional short circuiting between components by the addition of the bracket could potentially damage components. Spatial requirements of the stack still need to be maintained with the addition of the brackets, in addition to vertical clearance of the handles. The addition of a bracket could make access to connectors difficult or impossible without their removal. And finally the inclusion or removal of components could change the form of the stack and make the bracket's design obsolete.

2.1.2 Identified Areas for Improvement

Having completed the SWOT analysis, the following areas in the USAR-T robot were identified for improvement this year:

- A new, stronger arm with improved rigidity and reduced play in joints and reinforce chassis at arm base in anticipation of heavier arm.
- Addition of a more effective manipulator.
- Move the router into the head and redesign the head to accommodate this.
- Redesign flipper motor holders.
- Tidy electronics stack and add any additional sensors including a new accelerometer.

- Improve both the Server and Client software for the robot, including an improved HRI, arm control and overall quality of code.

2.2 Mechanical Systems Improvements

2.2.1 Arm Sub-System

The arm is one of the key systems of the teleoperated platform. Its primary purpose in previous versions of the robot was to orientate the main sensor and camera array to allow the operator to look around into areas inaccessible to a more basic camera mount on the top of the robot. The arm also was used by the operator to shift the robot's centre of gravity if situations dictated such a necessity such as shifting the centre of gravity forward to prevent the platform toppling backwards whilst ascending a steep incline.

The introduction of a manipulator this year, and the design decision to house the router in the 'head' has created additional design considerations for the arm, requiring additional strength and more accurate positional control.

2.2.1.1 Identification of Arm Requirements

In a collapsed building, or other unstable emergency situation, it can be highly beneficial for a robot to be able to pick up and manipulate objects around it. Applications range from being able to pass trapped victims bottles of water or first aid kits, turning off a leaking gas main, to removing light debris blocking the robot's progress.

The competition does not have specific guidelines for the arm design itself but has a light payload delivery criterion, giving the robot points for delivering a bottle of water or other payload to the victims (Jacoff 2009). Previous experience of the competition shows that victims can be located in regions up to approximately 1 metre from the ground, meaning that the robot should be able to pass objects up to at least this distance.

In the competition, payload weights are relatively small, but in real world applications, payloads could be considerably higher so a stronger arm may ultimately be of greater benefit.

2.2.1.2 Evaluation of the 2009/10 Arm

The previous year's evaluation of the arm was that despite intentions of rigidity, there was still significant free movement in the connections between the joint shaft, worm wheel and

driven side of the joint (see the looseness and subsequent alignment between shaft keyway and grub screw in Figure 1) which all should have been rigidly connected. This free rotational movement at each joint, amplified along the length of the arm resulted in considerable vertical free movement (and subsequent positional uncertainty) of the head.

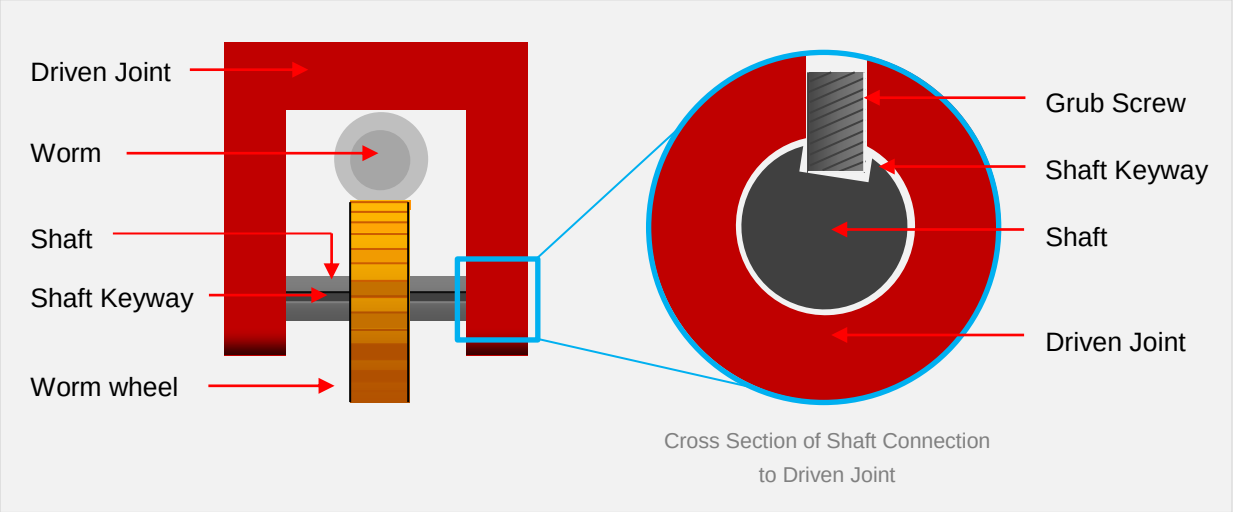


Figure 1: Old joint design highlighting looseness between grub screw and shaft

Over time, the large accelerations during the movement of the head are likely to have increased this play plastically as well as possibly introduced an element of creep deformation.

The high strength to weight ratio of the carbon fibre tubing was highly effective for the arm, an advantage identified in the last year's report (Warwick Mobile Robotics 2010, 43). The layout of the arm members and motor positioning was also effective; the vertical positioning and symmetrical loading meant that torsion in the joints and arm members was minimised.

Arm positional feedback is maintained by the highly compact Thinot potentiometers by Spectra Symbol. These work by a pressure-on-membrane method which makes them susceptible to environmental contamination from, for example, small pieces of debris falling in the wrong place. Such an occurrence can create a false positional reading causing the robot to suddenly compensate to the false position. This movement can result in the robot damaging its joints or motors by trying to move past the physical limits of the joints, or potentially causing the head to hit into its surroundings, damaging the cameras or other sensors.

High torque acting on the arm joints can create sufficient resistance to turning the worm wheel to cause the worm to try to screw linearly away from the motor. To compensate for this movement, a late stage modification was made to last year's arm, adding a small thrust bearing to the end of the worm in order to better secure it into the arm.

The 2009/10 arm was not designed with the intention of lifting additional payloads. As a result, there were limitations in the load that some of the joints, such as the joint attaching the arm to the head and the lowest joint, could hold.

2.2.1.3 Arm Specification

Having examined the arm requirements and lessons learned from the evaluation of the previous arm, several specifications were set for the new design:

Specification	Description
Arm Reach	Arm reach should be capable of reaching at least the same distance as the previous arm (measured at approximately 0.7m from the base to the end of the manipulator). It should also be able to rotate to reach areas within the forward 270° of the robot.
Arm Strength	Due to the larger head and delivery payload, the arm and its joints should be capable of dealing with an end load of [lost original estimate. Need to re-estimate load of head + payload].
Arm Rigidity	The rigidity of the arm should be improved over last year's design. Free end movement should not exceed 1cm.
Potentiometer Protection	The vulnerability of last year's potentiometer to damage or contamination should be mitigated. The potentiometer should be housed in such a way that the surface is not directly accessible from outside the joint.
Integrated Thrust Bearing	The thrust bearing preventing linear movement of the worm should be more integrated into the joint body than the previous design.
Ease of assembly/ disassembly	The arm and its joints should be easily assembled by non-permanent methods in order to facilitate disassembly for repairs / replacement of parts.

Table 1: Arm Specification

2.2.1.4 Development of Arm Design

2.2.1.4.1 Arm Layout

The first area of design was the layout of the arm itself. Concepts for arms were the members folded down 'side-by-side' such as the arm on the iRobot Warrior X700 (Figure 2) were designed to investigate improved vertical compactness, but ultimately excluded because of increased torsion on bottom arm members and joints due to the off-centre loading.

Image Credit: http://www.engadget.com/2007/10/17/irobot-readying-bigger-deadlier-warrior-x700-robot/

Figure 2: iRobot Warrier x700

As compactness was not currently an issue for the robot, the symmetrically loaded layout of the 2009/10 arm was concluded to be the most effective, minimising torsion.

2.2.1.4.2 Arm Members and Connection to Joints

According to the 'Arm Reach' specification in section 2.2.1.3, the arm along with the head should be more than 700mm when fully extended in order to give the robot sufficient reach to access the highest victims. Approximating the sum of the joint dimensions to be greater than 200mm, a 500mm span was required to be covered by the two arm member lengths. Splitting this length equally between the two members, lead to the proposal of two 250mm carbon fibre arm members.

In order to further improve the rigidity of the members and also to allow more space for internal wiring, a wider diameter (70mm) carbon fibre tube was chosen for the arm members.

The members would be secured using the same attachment method as the previous arm design, with radial screws screwing through the tubing into the joint component. This allows a secure attachment that can easily be removed if access to the motors inside the arm members is required. Rigidity on the round surface would be retained with sprung anti-vibration washers.

2.2.1.4.3 Improving Arm Rigidity

An investigation into the causes of rigidity loss in the previous design concluded that it came from mostly the key and keyway connection between the worm gear and joint shaft, and the grub screw connection between the joint shaft and joint component (as shown in Figure 1).

In order to reduce this movement, an examination into better non-permanent methods of securing these joints was conducted. This led to two main possibilities for improving connections; either through use of a splined shaft or a through-pin. Due to the manufacturing complexities of a splined shaft connection, the through-pin method was chosen.

2.2.1.4.4 Fundamental Joint Design

The chosen fundamental arm joint design was comprised of two major halves, demonstrated in Figure 3 below.

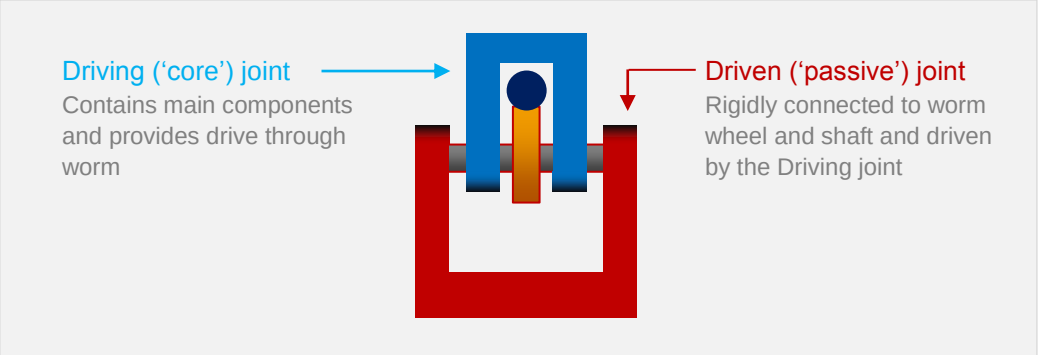


Figure 3: Chosen Joint Layout

The first half was the core joint, housing the motor, bearings and potentiometer for positional feedback. This halve was the 'driving' part of the joint and was made the inner of the two halves, with the second 'passive' half housing and protecting it. This 'driving' half of the joint was designed to be the same across all joints, not only to simplify manufacture and minimise the amount of required spare parts, but also to allow these halves to be interchangeable if faulty without having to replace the entire joint.

Also, by keeping the core joint electronics in the same half and static relative to the arm member, wire movement was minimised, reducing the chances of breakage.

The second, 'passive' joint half would vary, depending on the type of joint, with different designs for the head, elbow and base joints, detailed further in later sections. It rigidly attached using through-pins to the joint axle that in turn was attached to the worm wheel. The worm wheel was held in place against the worm by the load-bearing radial bearings incorporated into the 'driving' half.

2.2.1.4.5 Thrust Management

One of the issues identified in the old arm was the tendency of the worm gear to screw away from the motor when sufficient resisting force (formed by the moment of the arm components supported by that joint) was present on the corresponding worm wheel. This outward thrust risked damaging the joint motor's gear box so the previous year's team had attempted to mitigate this with a post-manufacture modification, temporarily attaching a very simple thrust bearing onto the end of the joint (see Figure 4).

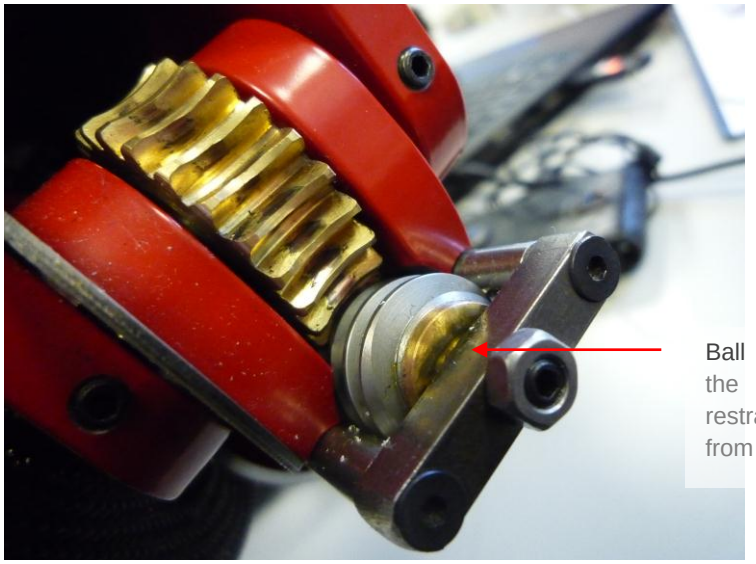


Figure 4: 2009/10 Thrust bearing design

This year an improved system with a better thrust bearing held in a more effectively incorporated bearing holder was developed as shown in Figure 5.

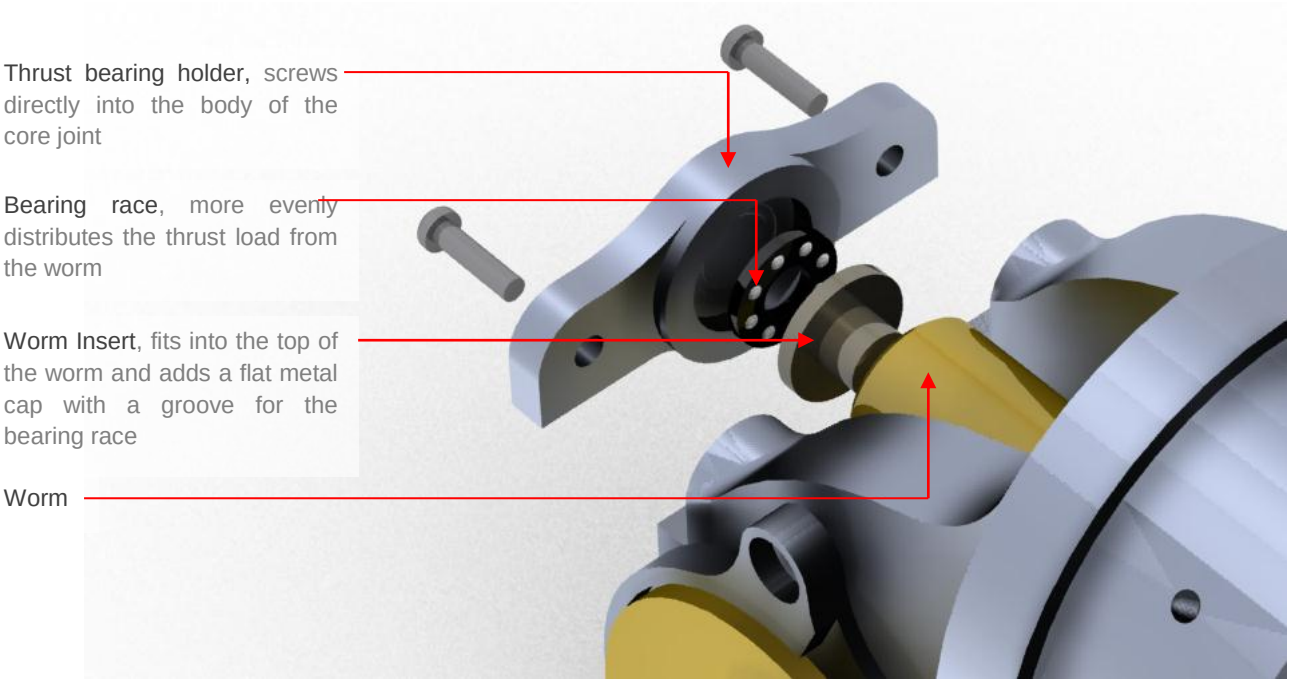


Figure 5: New Thrust bearing and holder

2.2.1.4.6 Potentiometer Protection

The ThinPot radial membrane potentiometers used in the previous arm work by sending out a voltage based upon the position of a sprung probe pushing on its active surface. The vulnerability with this type of sensor is that if a contaminant or piece of debris presses against the potentiometer, the robot will get a false positional reading on the arm which could cause it to try to move the joint to a position outside of its working range, subsequently damaging the arm, head or joint.

To prevent such an occurrence, a two-sided container for the potentiometer was designed that fully encloses and protects it. The two parts were designed to overlap each other in such a way that they couldn't tighten together, sandwiching the potentiometer. Sufficient clearance between the two halves prevented them rubbing together and generating small metal flakes that may cause contamination. A hole was made in one of the caps in order to allow the probe access to the potentiometer.

Originally the back plate on to which the potentiometer is attached was going to be part of the core 'driving' joint half mentioned in section 2.2.1.4.4. As the design of this component progressed however, it was determined that for machine accessibility the holes for bearings needed to be positioned on the outside of the joint part. This conflict was solved by creating an additional plate that fitted over the bearing hole of the active joint, upon which the potentiometer was attached. This is better explained visually in Figure 6.

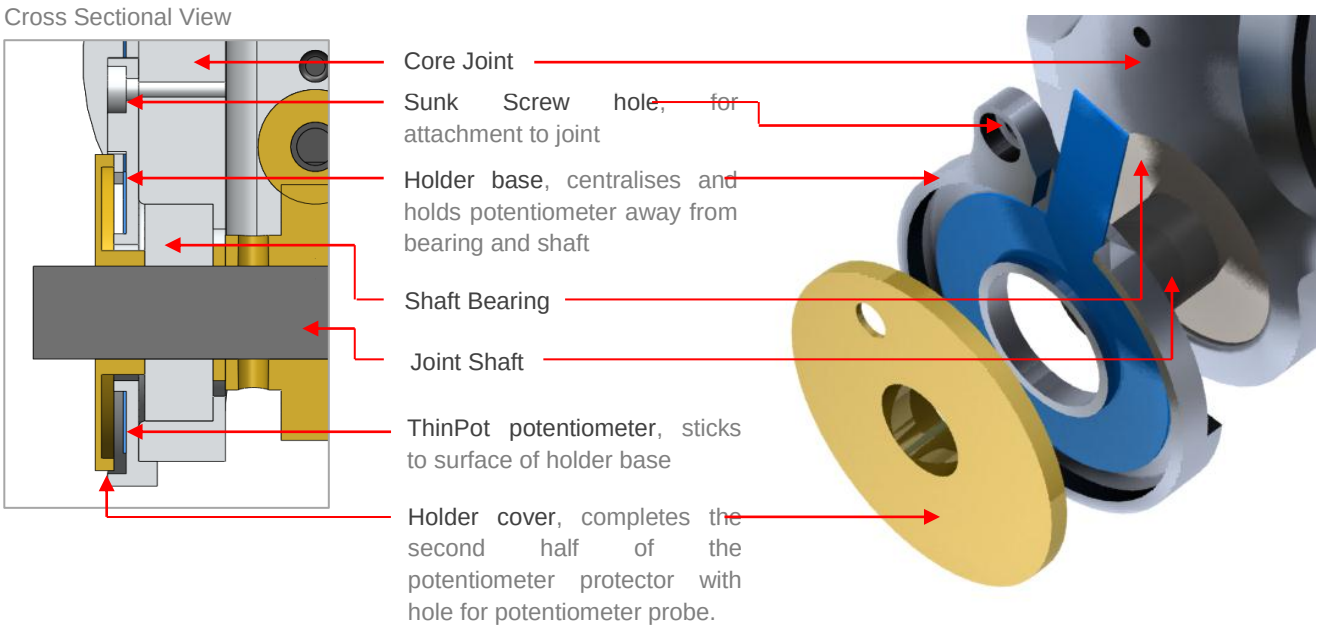


Figure 6: Potentiometer Holder

2.2.1.4.7 Motors and Gearing

The 2009/10 arm joints are driven by Maxon RE30 (310007) motors connected to Maxon GP32C 23/1 (166936X) planetary gear boxes. With a stronger arm this year, the original worm wheels were replaced by a larger, 30 tooth gear to help further reduce the torques on the motors.

If the arm is modelled as two concentrated ***M***kg mass joints positioned along a light 700mm beam with an approximate 3kg end load for the combined head and payload, the 'worst-case' system can be modelled approximately as shown below in Figure 7.

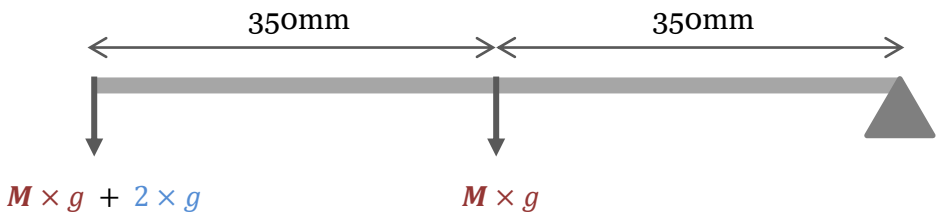


Figure 7: Arm Torque Model

Using the maximum continuous torque of the RE30 motor ($85x10^{3}$Nm) with a 1/690 gear reduction (due to the combination of the worm and worm-wheel and GP32 C gearbox), the equation for the maximum mass for a static arm in that position can be approximated as follows:

Head torque | Joint torque | Max cont. motor torque

$$0.7 \times 2 \times g + (0.35 + 0.7) \times \boldsymbol{M} \times g = 23 \times 30 \times 85 \times 10^{-3}$$

$$\therefore M \cong 2.35\text{kg}$$

So, provided the mass of each joint (other than the base joint), including the motors and gears, does not exceed 2.35kg, the current motor and gearboxes will be able to support the arm. As the loads are expected to be significantly less than this, the existing arm motors should have no difficulty holding the arm in even the most strenuous of positions.

Due to the high inertial effects and also the large frictional values likely to be inherent in the system which can only accurately determined once the arm is built, testing will be required to determine the acceleration which the arm is capable of reaching without exceeding the advised motor loads. The arm joints will be designed to accept new, more powerful motors if such testing indicates the current motors aren't able to cope.

2.2.1.4.8 Core Joint Design

Incorporating all the previously designed elements, the core, driving joint component, found in all three arm joints was designed. The design can be seen in Figure 8 and Figure 9 below:

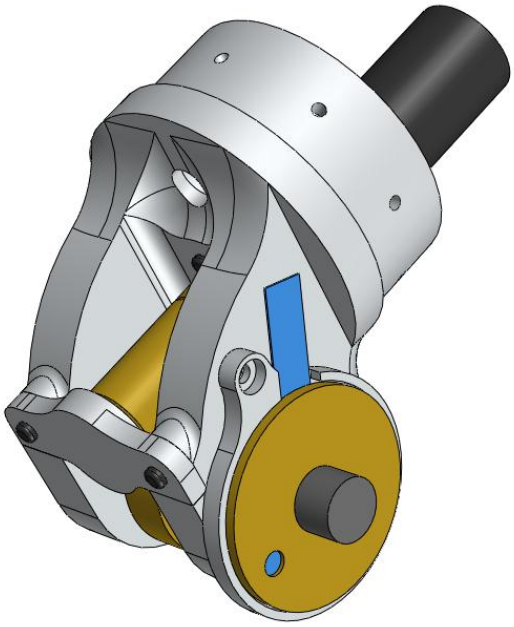


Figure 8: Core Joint Design

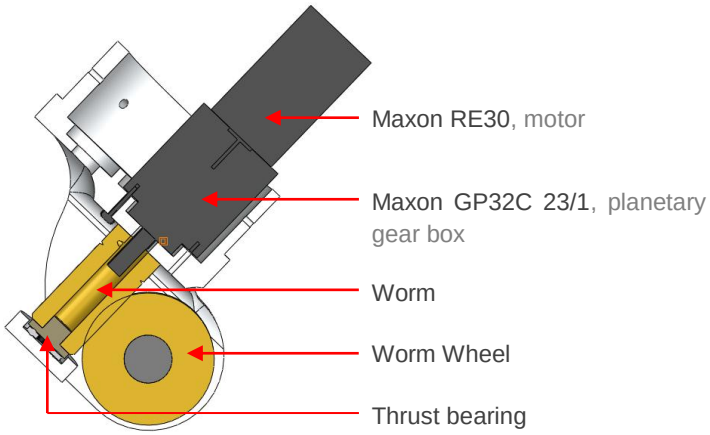


Figure 9: Core 'driving' joint main components

2.2.1.4.9 Base Joint Component Design

The base component is the largest of the passive (driven) joint halves. It carries the full weight of the arm and also requires being tall enough to allow just over 180° of free motion. It also is designed to house an additional motor, bearing, race and annulus gear arrangement that allows it to fully rotate about the robot's central vertical axis as well. This annulus attaches to the larger base plate that will be discussed in greater detail in section 2.2.2 and can be seen in the form of a cross section below in Figure 11. The final base plate design is shown below in Figure 10.

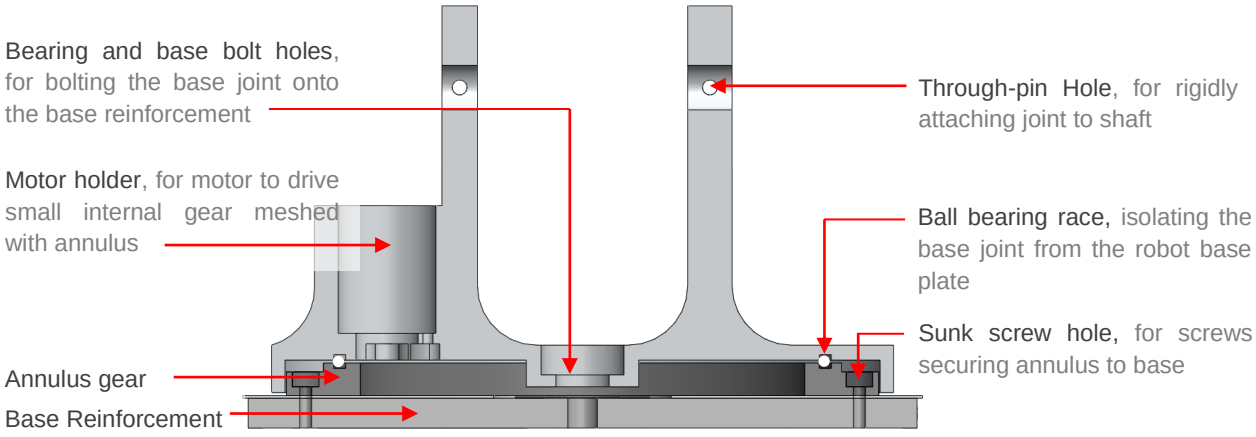


Figure 10: Base Connections to Body

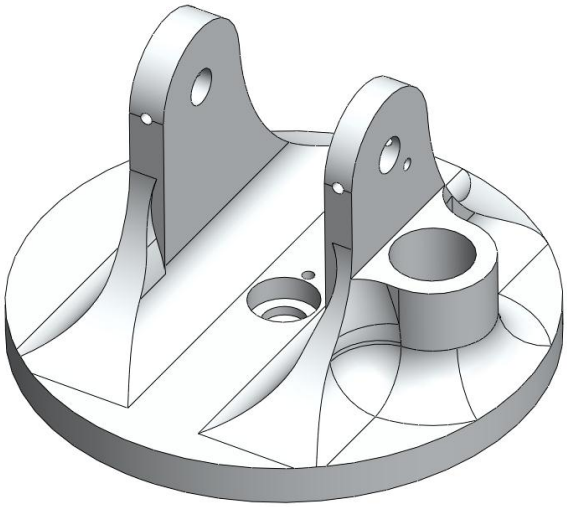


Figure 11: Base Joint Design

2.2.1.4.10 Elbow Joint

The elbow joint was designed to attach to an arm member and allow 180° of free motion. The design is shown in Figure 12 below.

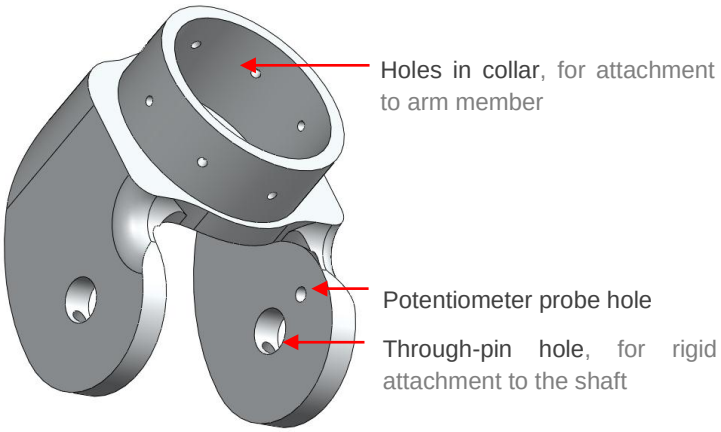


Figure 12: Elbow Joint Design

2.2.1.4.11 Head Joint

The head joint was designed to allow a greater freedom of head motion (~270°) in order to ensure maximum visibility for the operator. The head sits above arm for to ensure a complete, unobscured view around the robot. The head joint incorporates a thrust bearing to support the head and holes that align with the Dynamixel RX-64 servomotor's plate that is used to rotate the head. Further details on this can be found in the head section (2.2.3) . The head joint design is shown in Figure 13 below.

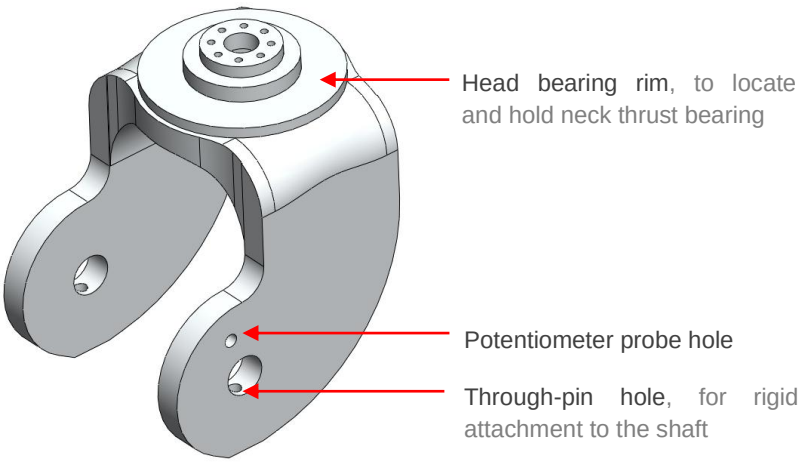


Figure 13: Head Joint Design

2.2.1.5 Final Arm Design

Having assembled all of the designed components along with the head developed in section 2.2.3, the final arm is assembled as shown in Figure 14.

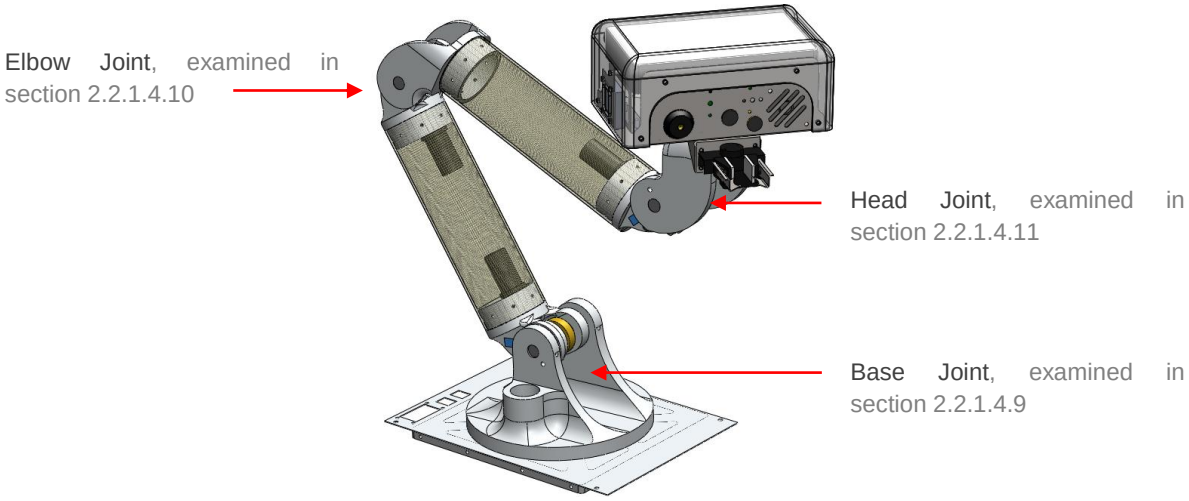


Figure 14: Final Arm Design (with head and base plate)

2.2.1.6 Testing & Evaluation of Arm

2.2.1.6.1 Unanticipated Force Vector

The force vector acting on the worm from the worm wheel can be broken down into two perpendicular components, one acting along the worm shaft and the other acting radially on it (see Figure 15).

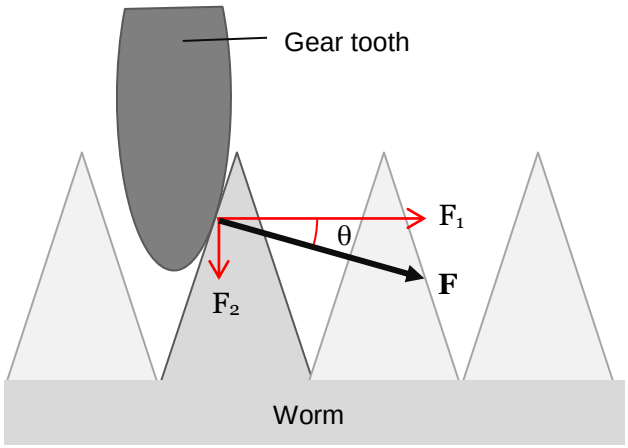


Figure 15: Force Vectors acting on worm at Worm/Worm-wheel Interaction

During the design of the joint, the F_2 force component had been thought to be negligible and subsequently only a simple, unidirectional thrust bearing was used to hold the F_1 thrust force component.

The large magnitude of the force caused the F_2 component to be more significant, resulting in approximate cantilever bending on the worm shaft. The thrust bearing currently holding the end of the worm in place was only designed in anticipation of a unidirectional load. To restrain the worm from this deflection in future, the current thrust bearing may need to be replaced by a tapered roller bearing that will accommodate both this force and the original thrust force (F_1).

2.2.1.6.2 Additional frictional component

When the movement direction is reversed and the F_1 component in Figure 15 acts towards the base of the shaft. With the heavier arm, the magnitude of the F_1 force will be larger than previous arms and may cause significantly higher friction between the base of the worm and the joint face against which it is mounted. An additional thrust bearing may be required in this section to reduce any such friction.

2.2.1.6.3 Reduced Backlash

The measured backlash for the new design showed a free vertical movement of only 10mm that was considerably less than previous years. This additional freedom was investigated and decided to be caused by backlash between the worm and worm wheel. In future years this could be reduced by the use of anti-backlash gearing.

It is also thought that this looseness between the gears was caused in part by the bending of the worm shaft away from the tooth, caused by F_2 load identified and addressed in section 2.2.1.6.1.

2.2.1.6.4 Potentiometer Issues

During the course of the competition, there were several issues with broken potentiometers or wires giving wrong positional values to the motor controllers and subsequently causing unexpected and potentially damaging arm movements.

A pair of potentiometers on each arm joint could be used to add redundancy to the system to reduce the impact of such occurrences.

2.2.1.6.5 Strain on Motor and Control Boards during movement

As discussed in section 2.2.1.4.7, although the arm motors were easily capable of holding the arm in a static state, the inertial and frictional loads on the motor would require real world testing to truly evaluate. Manufacturing delays resulting in the arm arriving the day before the robot left for the competition meant that there was no time available for testing to ensure the motors were capable of such motion.

During strenuous PID loop testing and configuration, which subjected the arm to very large peak accelerations in a short timespan, the motor control boards (Figure 16) and subsequently the shoulder joint motor, proved incapable of handling the stress of the heavier arm.

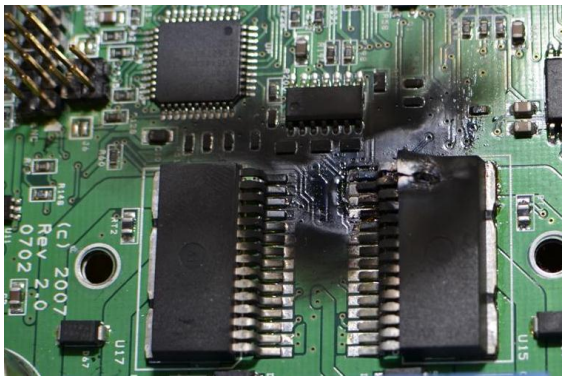


Figure 16: Burnt out H-Bridge on Motor Control Board

The solution to this problem was simply to upgrade the motor control boards and motors, however this change needed to be made after the competition which unfortunately limited the arm movements to only using the top 2 joints.

2.2.2 Arm Base Sub-System

The base support is located at underneath the base joint of the arm. This part distributes some of the weight in the arm to the chassis. It also supports the arm on the teleoperated robot and gives the chassis some structural rigidity. This plate on the previous years' arm can be seen in Figure 8.

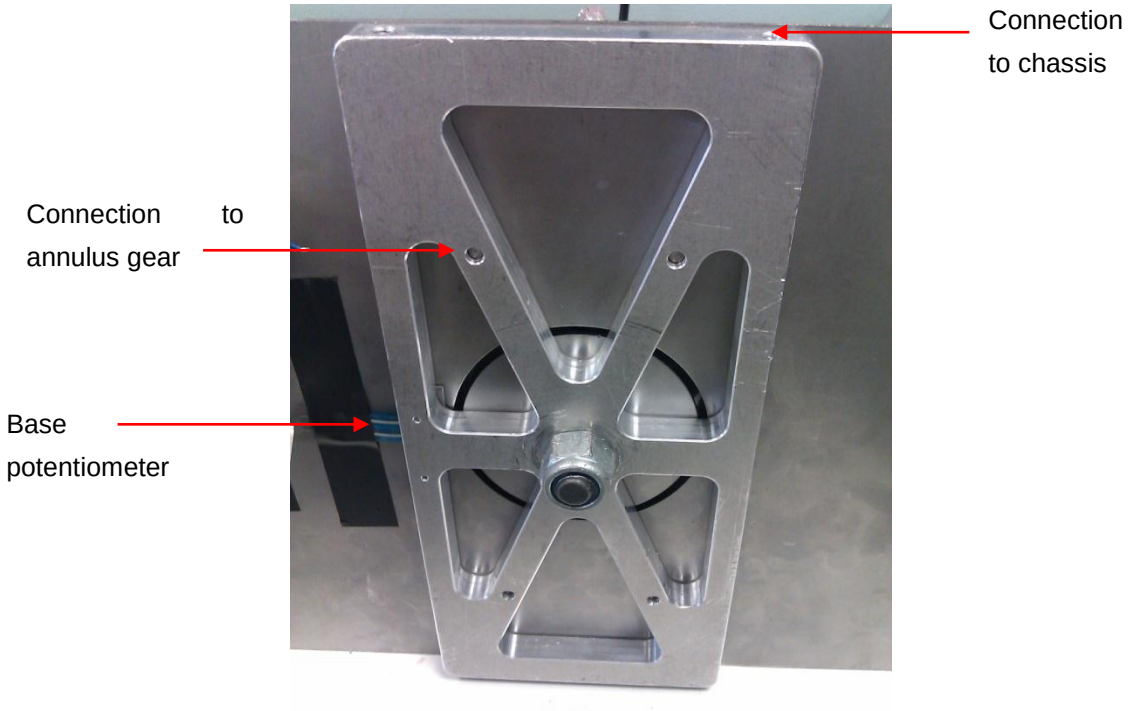


Figure 17: Old Base Support

2.2.2.1 Base Support in 2009/10 Robot

Although the base support had not deformed under the stress of the arm, it did need redesigning as it was not preforming well enough due to the base supports poor connection to the USAR-T chassis, giving much of the free movement for the arm. This obviously needed redesigning when considering the new arm. The new arm needs more support as it was significantly larger.

2.2.2.2 Specification

The new base has to have a large footprint but have the same depth of the previous base support. This is due to the space constraints inside the robot. The base support also needs to have a simple design that is easy to manufacture.

2.2.2.3 Final Design

The final design can be seen below. This is a very simple modification, widening the original base support to allow more of support to be attached to the chassis. This attachment came in the form of bolts, as they allow easy removal of the base support and attached arm.

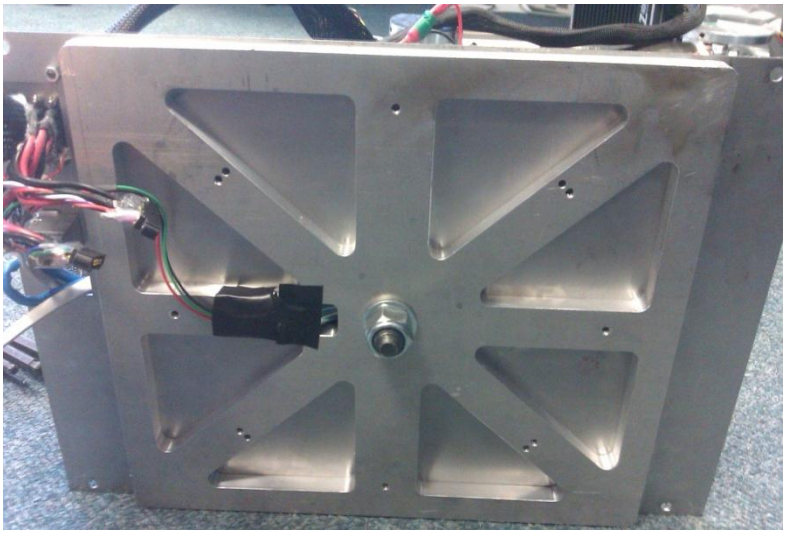


Figure 18: New Base Support

Figure 19 shows how the base support was secured to the chassis. This used simple cap-head bolts.

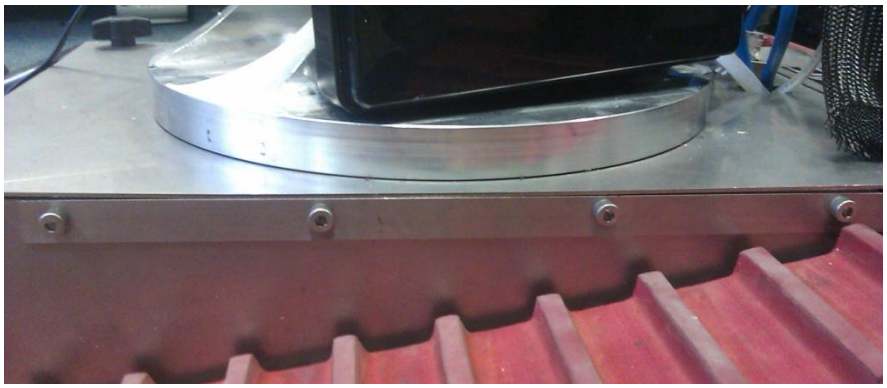


Figure 19: Base Support Fixture

The Figure below shows how the base support is assembled. The annulus gear is attached to the base support, through the top plate of the USAR-T.

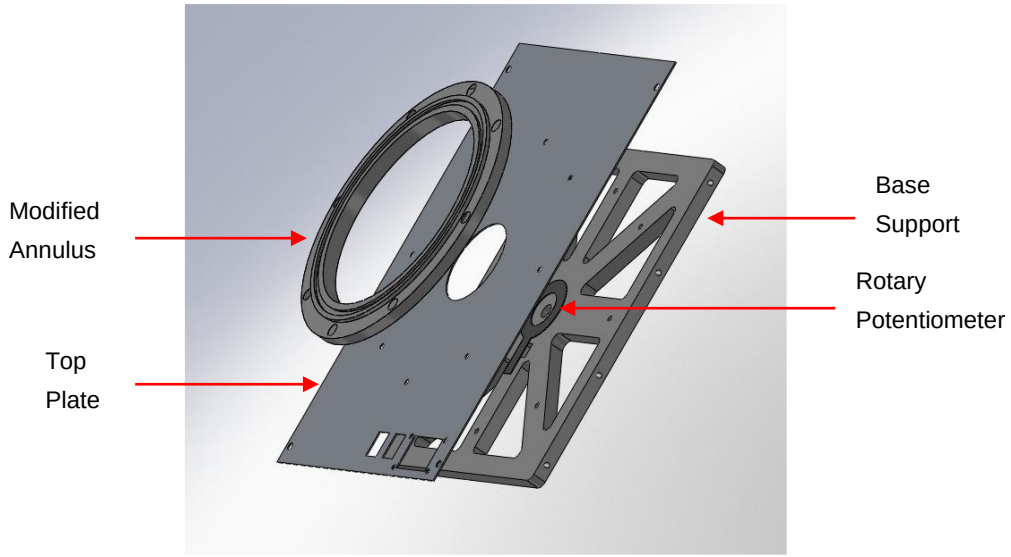


Figure 20: Base Support Assembly

2.2.2.4 Evaluation and Testing

The base support for the arm was affixed to the bottom of the base joint and arm. This greatly reduced the free movement that the arm had on the chassis, even with the increased weight of the arm. However, the added fixings to the chassis mean that the whole robotic arm takes longer to attach and remove. This proved frustrating during the competition as WMR team had to disassemble and reassemble the robot on many occasions.

2.2.3 Manipulator Sub-System

2.2.3.1 Previous Year's Manipulator

Last year's team was the first to create a basic manipulator for the competition. The head design, particularly the neck rotational freedom (rotating on an Rx-64 Dynamixel servo) was not sufficiently strong for larger loads, limiting the maximum load on the neck to be to 1kg.

In the competition the points system indicates 20 points are available if a robot is able to pick up an object from a designated area and drop it off in a box in front of a victim. This is equivalent to four times the amount of points awarded for identifying a victim with the webcam, or equivalent to mapping the entire arena, so last year's team manufactured a hook whilst at the German competition to attempt to gain extra points.

In terms of real world use, a manipulator would help to bring water to a victim, help move rubble out of the way, or bring a means of communication to a victim (such as a two-way radio).

Figure 21: 2009/10 Head Design with Hook

Manipulators in the past have failed to perform due to the arm being too unstable. A redesign of the arm this year has resulted in added strength, stability and better control using inverse kinematics.

2.2.3.2 Manipulator Specification

Drawing up a specification for the manipulator was necessary to ensure the robot would be able to pick up a selection of objects in the competition. These objects are a 500ml bottle of water, a block with an eyelet, and a two-way radio.

Specification	Description
Grip distance	The grip should be able to open wider than 35mm to accommodate a bottle top or block eyelet. It should be capable of closing to 0mm.
Grip strength	100N, based on approximately one quarter of the median grip strength of an average male (430N (Massy-Westropp, et al. 2004))
Cost	The cost of materials, components and manufacturing should not exceed the budgeted £200.
Manufacturing time	One week.
Size	Maximum of 100g in order to not add any significant weight onto the head. A compact size of less than 75mm^3 would fit well onto the size of the head.

Table 2: Manipulator specification

2.2.3.3 Manipulator Design

In selecting the final design for the gripper, several different solutions were identified and evaluated including off-the-shelf manipulators and custom in-house designs.

2.2.3.3.1 Off-the-shelf Solutions

There are predominantly two gripper types for this scale of robot; electro-mechanical and pneumatic. Electro-mechanical solutions would require extra space on the power board and additional control systems but are easily integrated into the existing systems. Pneumatic solutions would require an air source, pneumatic circuitry, and simple electrical control.

One consideration of deciding between these was pressure control. An electro-mechanical solution would require pressure pads and feedback in order to control grip strength and avoid damaging objects being held. This would also avoid the motors driving against themselves for extended periods of time. In comparison, a pneumatic system would be able to hold an object by simply maintaining the air pressure in the manipulator. Both systems were considered and possible solutions were discussed.

In terms of the types of actuators available for purchase, two main types were considered – parallel grippers and angle grippers. Both were available in pneumatic and electro-mechanical systems.

Figure 22: SMC Parallel Gripper (RS Online n.d.)

It was possible to acquire parallel grippers from RS Online with adequate gripping strength (RS Online n.d.) but with a maximum stroke of 25mm. This stroke could be extended by manufacturing wider grippers to attach onto the fingers. A better solution than this was the angle grippers that gave a larger and more easily modified stroke, by simply extending the length of the fingers.

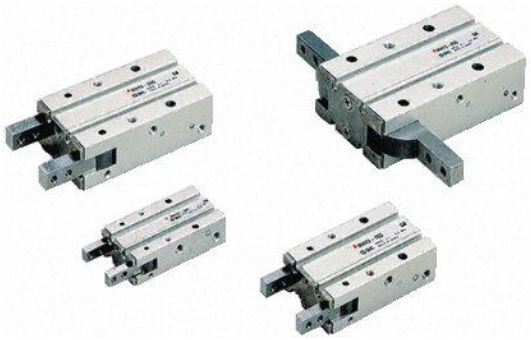


Figure 23: SMC Angle Gripper (RS Online n.d.)

A main consideration was cost as the cheapest option (the angle gripper) was £298 for the actuator alone. On top of this, a pneumatic solution would require further investment in a compressor and/or an air source with pneumatic tubing too.

Various ready-made manipulators were considered from robotic hobbyist websites. Industrial machinery manipulators were found to be overly costly for WMR's budget.

Figure 24: Robot Hand (Active Robots 2011)

This robot hand is designed to mimic a human hand and would be capable of lifting any of the objects required in the competition. There is a possibility of any smooth objects from slipping out the bottom of the gripper; therefore it may require additional degrees of freedom to tilt once an object is gripped. This also does not come provided with any servos which will add extra cost.

Figure 25: Lynx Gripper Kit (Active Robots 2011)

The Lynx Gripper Kit is adequate to grip simple objects and is easily assembled and installed. The main disadvantage is the lack of freedom in modifying this design.

Figure 26: Little Gripper Kit (Active Robots 2011)

The Little Gripper Kit was the best off-the-shelf solution found as it had the correct stroke distance to grip a bottle head, was low cost at £46.64, and had the potential for modifications. The casing also had holes free for attachment onto a robot unit such as the USAR-T's head.

2.2.3.3.2 Pneumatic Solution

2.2.3.3.2.1 Pressurised Air Source

Various options were considered for air sources. Although used in mobile vehicles such as cars, it is common to have a compressor on board to act as an air supply. This is found in systems such as pneumatic paddle gear shifting (Geartronics 2011) which uses a heavy-duty 12V air compressor. This would be excessive for just a manipulator so a removable and replaceable air source would be preferred. Two systems seemed easily accessible and common; these were CO_2 sparklets (such as paintball gun cylinders) and N_2O sparklets (such as whipped cream chargers). CO_2 sparklets have the potential to interfere with readings in the CO_2 sensor if a leak occurs, so an N_2O sparklet would be preferable.

When exploring the possibility of using pneumatics, there where additional things to consider. The first was one of connecting a high-pressure canister to a pneumatic system. This would require a specially manufactured connector fed through to a regulator and then attached to the pneumatic system, therefore adding to manufacturing times and resources. Another would be that sparklets have a finite number of shots before they run out, therefore it must be able to run for the duration of a 20 minute search and rescue run and also be easily removed and replaced. The final issue being that a pneumatic system is an altogether separate sub-system to integrate into the robot. This increases complexity and increases the possibility for complications during the life of the system. One such complication would be pneumatic tubing running up the arm and any additional packaging to consider.

2.2.3.3.2.2 Pneumatic Circuit

The proposed pneumatic circuit for a pneumatically actuated gripper system is shown in Figure 27 below.

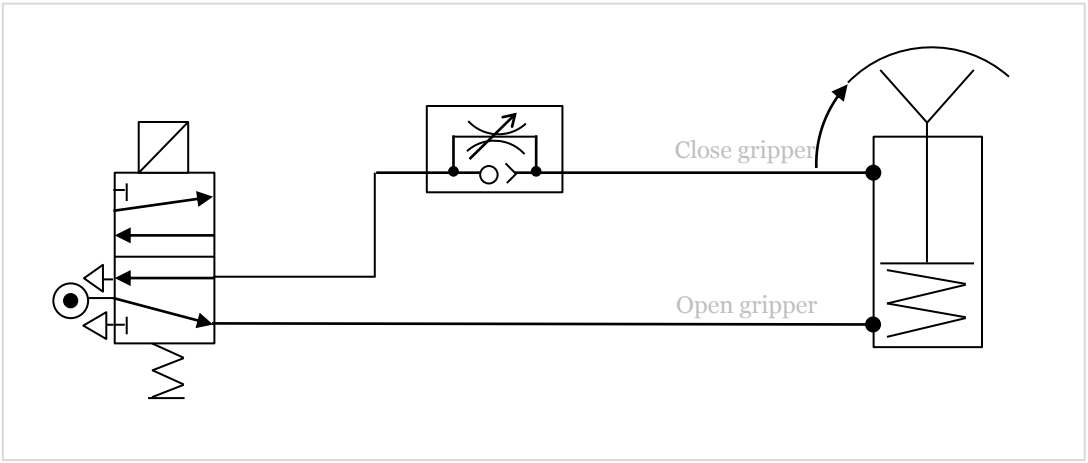


Figure 27: Pneumatic Circuit

The pneumatic circuit for the gripper is comprised of 3 components; an electrically actuated, spring-return 5-port valve, a one-way flow restrictor valve and the gripper itself. In the default (spring return) valve state and the non-pressurised state, the gripper is open. When the valve is actuated through a signal from the electronic circuit, the gripper will close at a speed limited by the adjustable restriction in the flow restrictor. Upon loss of electronic signal, the 5-port valve returns to the default position causing the gripper to reopen. The unidirectional nature of the flow restrictor means that the opening speed of the gripper is left unrestricted.

2.2.3.3.2.3 Possible Modifications to Circuit

There are several additional modifications that could be made to the circuit if it were developed further. A 'latching' gripper state system would mean that the input air pressures would not need to be maintained to keep the gripper closed, subsequently saving pressurised air. The simplest solution for this would be removing the spring return on the gripper actuator, although this would not maintain gripping pressure. Another modification would be an electronically controllable pressure regulator between the pressure source and the rest of the circuit that would allow the operator to control the ultimate closing pressure of the gripper, allowing the robot to pick up more delicate objects.

2.2.3.3.2.4 Analysis of Pneumatic Solution

The implementation of a pneumatic circuit on the robot adds a large amount of complexity, requiring a reliable sufficiently large pressure source, such as the sparklets discussed in section 2.2.3.3.2.1 above. It was decided that the implementation of such a solution on the

robot would be more complex, space consuming and expensive than alternative solutions so the pneumatics solution was ruled out.

2.2.3.3.3 Electro-Mechanical Solution

After investigating pneumatic systems the team decided it was overly complex and opted for an electromechanical solution. The design aimed to incorporate the best feature of the SMC angle gripper (the large, adjustable stroke distance), but without the high costs.

The initial concept gripper design features were:

- Angular gripper means larger stroke for the same motor size and therefore more versatile grip
- Uses motors we already had (RX10) and also the same worm/gear as in the joints. This means parts are universally swappable which simplifies repairs and ordering spares
- Bent sheet metal and spacing rods are easily manufactured using available WMG facilities.

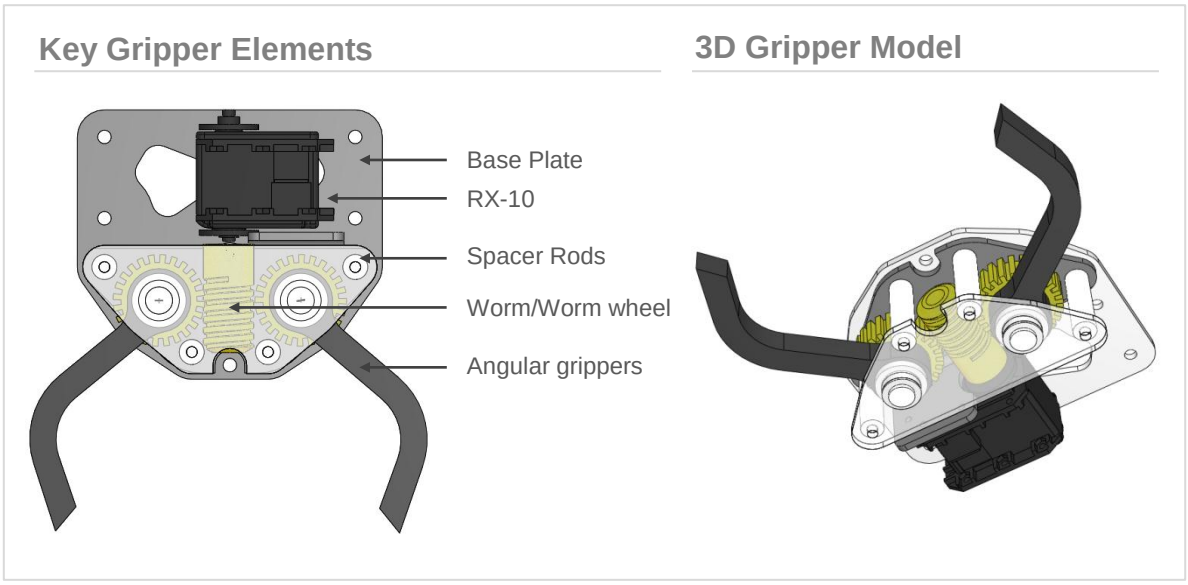


Figure 28: Manipulator Concept

2.2.3.4 Final Manipulator Design

Due to strict manufacturing time restrictions, it was decidedly quicker and easier to buy a ready-made solution from Active Robots rather than to build a bespoke pneumatic or electromechanical system. The subsequently chosen option was the best 'off-the-shelf' solution (the Active Robots 'Little Gripper Kit'), identified in section 2.2.3.3.1, consisting of two servo motors controlling 'roll' orientation and a parallel closing motion for the gripper. The 'roll' freedom was considered an unnecessary complexity so was removed to leave the

single servo closing the gripper (Figure 29). Circuit integration was achieved simply by wiring the gripper servo into a spare slot on the existing servo control board in the robot.

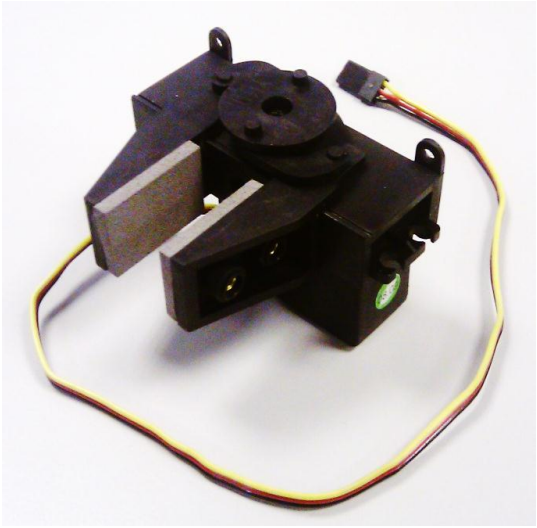


Figure 29: 'Little Gripper Kit' Off-the-shelf Manipulator with one servo removed

To reduce the strain on the motor, it is possible to add a modification (shown in Figure 30) to the gripper that closes under the bottle lip, holding it in place by physical constraints as opposed to purely frictional, subsequently reducing the strain on the servo. This design would also be able to lift blocks with eyelets. This gripper modification can be easily manufactured in-house; laser cut out of 1.2mm stainless steel sheet and then bent into shape. An additional ridge bent at the back reduces the gripper modifications' tendency to bend outwards when gripping an object.

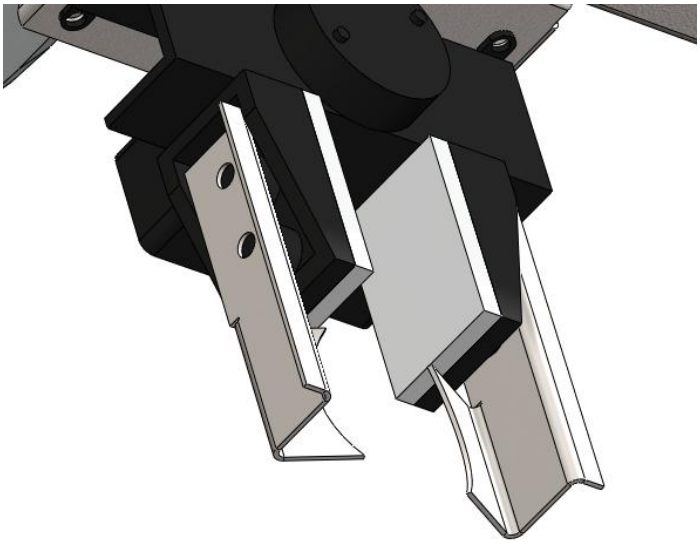


Figure 30: Final Manipulator Design with Gripper Attachments

2.2.3.5 Testing & Evaluation of Manipulator

The final design was tested in-situ at the competition, as it required the full assembly of the arm to fully test it. However, unit testing was completed on the electronic control of the gripper movement. A simple command was programmed in to open and close the gripper that allowed on and off control of the gripper.

Mechanically the manipulator was more than sufficient to grip an eyelet at the competition and was tested to grip a full water bottle in the WMR lab. After checking the visibility of the gripper from the front robot webcam, a slight adjustment was made to the orientation of the gripper extensions, swapping them over so the main metal grips were positioned higher and subsequently more visible to the operator.

2.2.4 Head Sub-System

2.2.4.1 Head Requirements

The head of the USAR-T robot acts a housing for the large variety of sensors used by the operator to navigate the robot, locate potential victims and even communicate with them. The head also contains within it a rotating 'neck' joint that gives the operator the ability to look around without rotating the arm about the base joint. With the addition of the manipulator this year, the head will also be required to contain the manipulator and support the weight of the payload. This year's head also will need to carry the router, as decided upon in section 2.1.2.

2.2.4.2 Head Design

The core structure of the head is a bent stainless steel sheet base (Figure 31). This base is designed to holds the majority of the head components, with screw points which all parts attach to.

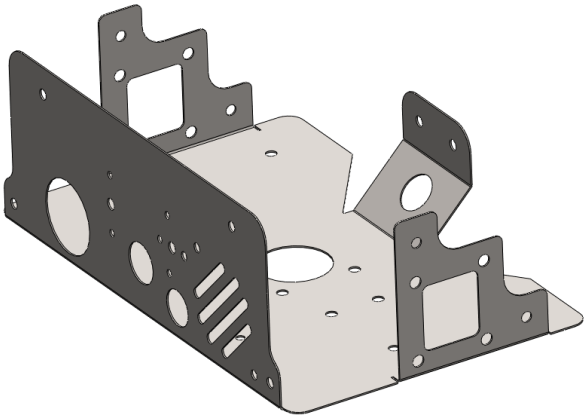


Figure 31: Head Base Plate

A second bent sheet metal mount (Figure 32) acts as a platform on which the router sits, held above all the other components in order to minimise the head's spatial footprint, and another mount (Figure 33) screws onto the base of the head, designed to hold the manipulator, detailed in section 2.2.3.

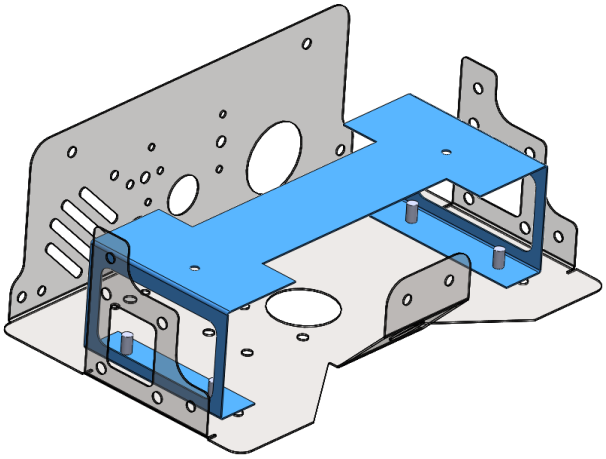


Figure 32: Router holder (Highlighted)

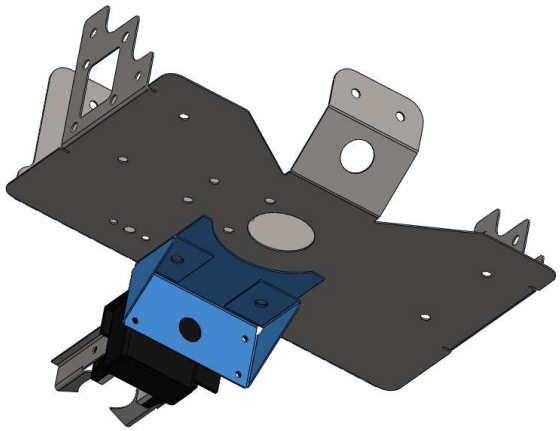


Figure 33: Manipulator Holder (Highlighted)

A large thrust bearing is used to support the 'neck' joint (which was the limiting factor for the load in the previous arm design), from the offset load expected to be generated by a payload held by the manipulator.

A Rapid Prototyped head cover fits over the remainder of the head, protecting the internal components and adding additional structure to the main sheet metal component. Holes at the back allow the router aerials to poke out instead of being internally housed, further reducing the overall size of the head.

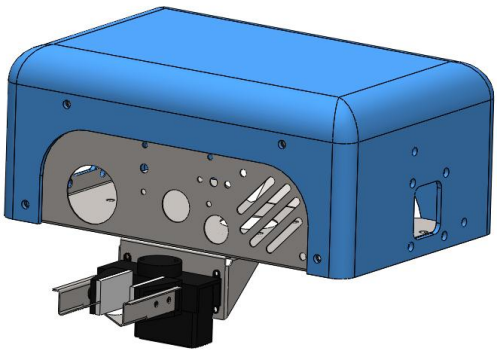


Figure 34: Head Cover (Highlighted)

2.2.4.3 Final Head Design

The final, assembled head design can be seen below in Figure 35.

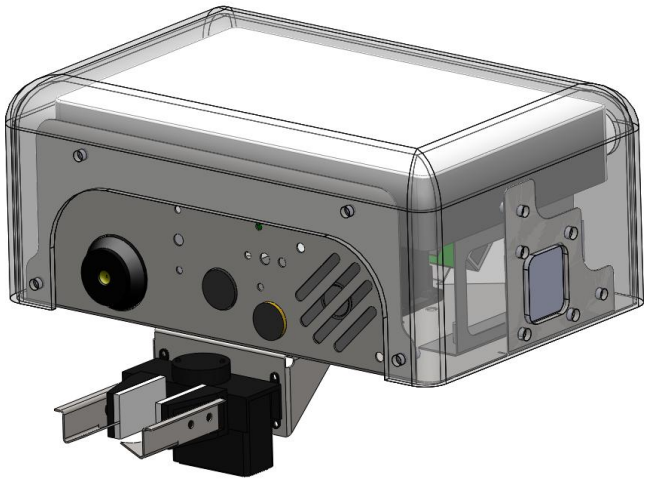


Figure 35: Final Assembled Head Design with Sensors and Router

2.2.5 Stack Casing Sub-System

2.2.5.1 Introduction

One of the issues identified with the USAR-T's electronics stack was the disorganisation of wires around the outside of the stack (connecting components inside the stack) and the frequency that they became disconnected when the stack was removed. It was decided that a frame should be designed to fit around the outside of the stack to keep the connectors and wires close to the stack, and that handles should be placed atop the stack to allow for ease of removal.

2.2.5.2 Specification for Stack

From the SWOT analysis of the robot in section 2.1.1, several improvements were identified for the stack:

- The bracket should not compromise the USAR-T's stack design structure for compactness and ventilation.
- The bracket should be designed with accessibility to the connectors it will be covering in mind, and yet it should contain them adequately.
- The bracket should be electrically isolated to prevent short-circuiting between stack components via the wires that may connect with the bracket.
- The handle design should be ergonomic (suitable for a comfortable grip) and conform with the vertical clearance with the top of the stack.
- Finally the design of the bracket should be slightly flexible with regards to the possible change in dimensions of the stack.

2.2.5.3 Design Methodology

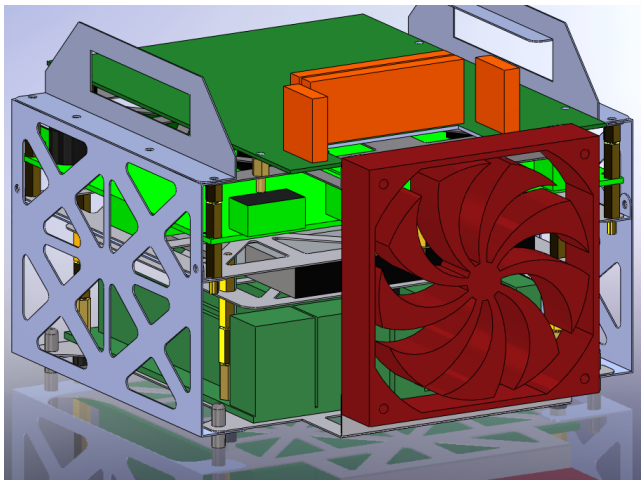


Figure 36: The Solidworks model of the USAR-T's stack with brackets and handles

The Solidworks model of the USAR-T stack (Figure 36) was already available from the previous year's files, and so the bracket for the USAR-T's stack was modelled directly on top of this design, after of course ensuring that the dimensions and layout of the stack had remained relatively unchanged.

The stainless steel bracket was designed so that it could fit around the top and bottom plates of the stack. This would allow for flexibility in slight changes in the height of the stack as the stainless steel was tall enough to bend and flex small distances. The vents in the side of the bracket were not designed for ventilation; as they were to be adjacent or touching the sides of the chassis; instead they were for access to the interconnecting wires on the side of the stack should they come loose, and the wide curves in their corners were designed with finger access in mind. Finally the brackets were to be powder-coated with an electrically insulative layer.

The handles were designed separately so that they could be easily attached or removed without the need for the bracket (just in case said bracket became defunct through a change in the stack structure). The apertures were designed so that fingers could comfortably fit through and grasp the tab on the other side. The handles were also designed to protrude upwards partway into the unused side sections of the top of the stack, so that the fingers of the user gripping the handles were away from the edges of the chassis.

2.2.5.4 Evaluation

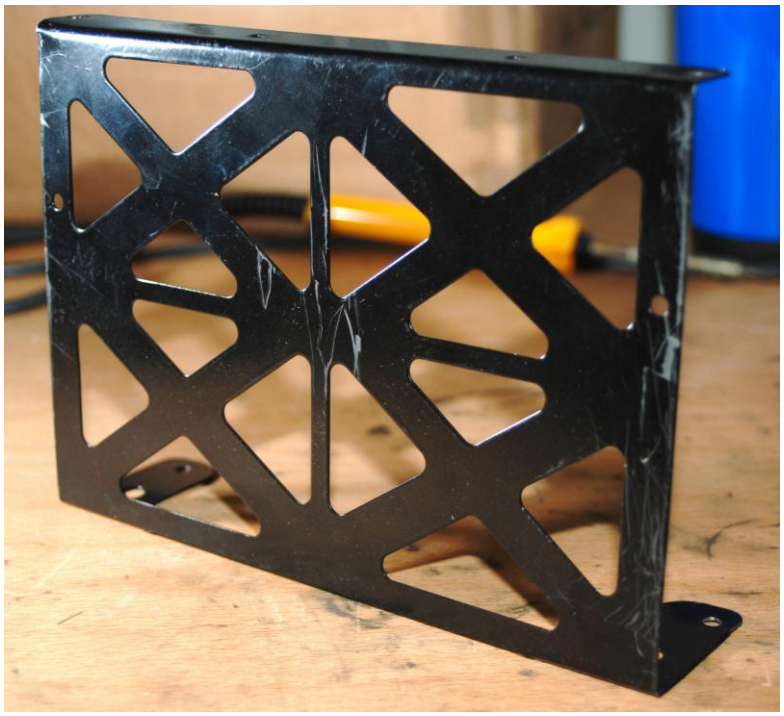


Figure 37: The manufactured stack bracket

The bracket could easily be attached to the sides of the stack and despite a small amount of shearing of its coating on the side of the USAR-T's chassis it was a very tight fit. However, as the competition drew nearer it was required that the stack had to undergo several late modifications and continual maintenance through the competition itself, which resulted in the brackets not being reattached as the work on the stack was never finalised for this year.

The handle's vertical clearance was off by a matter of millimetres too tall, and so conflicted with the arm support on the USAR-T's lid. Cutting the top section off the handle would reduce its structure, making the top section too thin to bear the weight of the stack effectively, and the handle itself couldn't be re-folded. The addition of a new support for the upgraded arm would further reduce the height of the chassis interior: The handle would have to undergo a change in its design in order to account for the reduced clearance. Unfortunately due to time constraints (the support for arm the arriving a week before the competition) and aforementioned constant modification of the stack new handles were not manufactured and hence never implemented into the design.

2.2.5.5 Recommendations

Once final modifications have been made to the electronic stack, implementation of the manufactured brackets should be attempted. The handles should be redesigned to meet the new special requirements; shortened to be specific, and yet maintain their ergonomic design.

2.2.6 Motor Clamp Sub-System

Previously, the flipper motors have been held in place using motor clamps. These are located towards the front and back of the robot. They are used to keep the motors in place against the applied forces of the chain drive and motor torque, while also helping maintain the structural rigidity in the chassis. The previous year's design for the clamps can be seen in Figure 38 below.

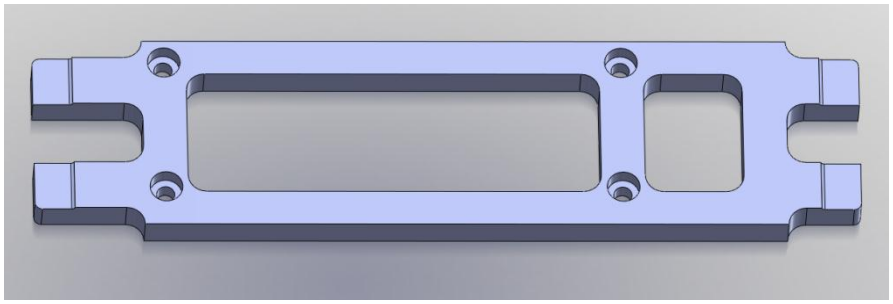


Figure 38: CAD image of old flipper motor clamp

2.2.6.1 Previous Year

One problem that has become obvious with the USAR-T robot's chassis was the slack in the flippers. This meant that the robot could not perform as well, with the slack in the flippers causing unwanted movement and uncertainty when the flippers are used. This slack was measured at roughly 30°, which was unacceptable as the operator of the USAR-T could not predict where the flippers were when tackling rough terrain.

When this was investigated further, the origin of the slack was precisely located to the flipper motor clamps. The deformation of the flipper clamp can clearly be seen below in Figure 39. The aluminium part has deformed under the constant stress and fatigue. This is because at points during the running of the robot, its full weight was on this flipper clamp as the stress is transferred through the chain that connects the motor and flipper axle.

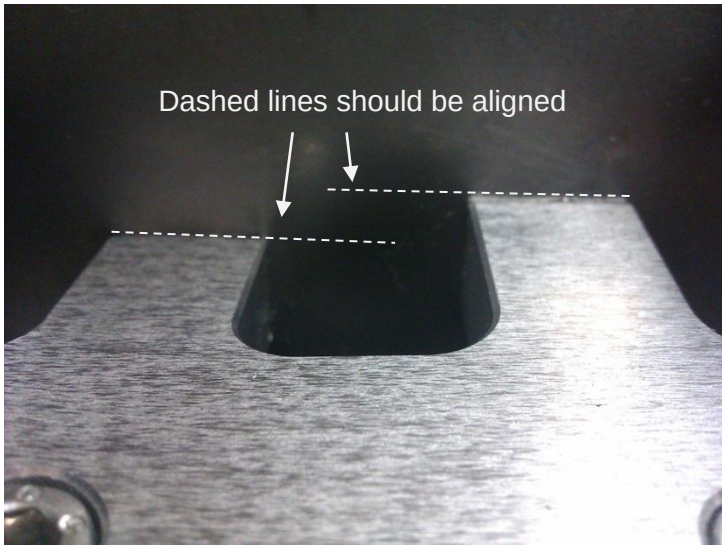


Figure 39: Deformation in original Flipper Motor Clamp

2.2.6.2 Specification

The new design for the flipper clamps need to be stronger in torsion than the older flipper motors using the same material. As the space inside the teleoperated robot is also a constraint, the size of the flipper clam must be the same size. This part also needed to be a simple design so it can be manufactured quickly and easily.

2.2.6.3 Design Method

From the results of the deformation, it was clear to see which parts in particular were failing. Solidworks was used to redesign the part, using the old flipper motor clamp as a reference, so as to keep good space efficiency. Solidworks Simulation (FEA software) was then conducted on the old part and the new part to show the difference between the designs, the results of which can be shown below in 2.2.6.5.

2.2.6.4 Final Design

The final design was chosen as it gave the best strength given the obvious space constraints inside the chassis of the robot. The flipper clamps are manufactured using the CNC machines inside the IMC by the technicians.

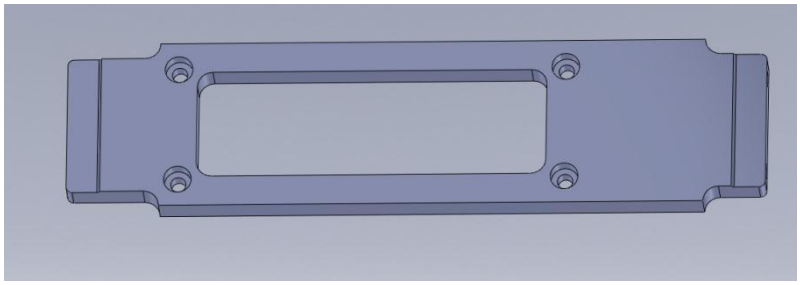


Figure 40: CAD model of new flipper motor clamp

Figure 41: Photograph of new flipper motor clamp

The notches that can be seen on the side of the part meant that some of the space could be saved without damaging the structural integrity of the part.

2.2.6.5 Evaluation and Testing

FEA was conducted on the part to show the deformation of it under stress. The FEA results can be seen below and show that the part is much stronger than the previous model and is therefore less likely to fail.

Figure 42 shows the FEA of the original flipper motor clamp. This shows that the maximum deflection for this part was over 5mm, shown by the scale on the right hand side of the figure. This seems about half of the actual deformation in the original part, however still shows the effect it had on the performance of the robot.

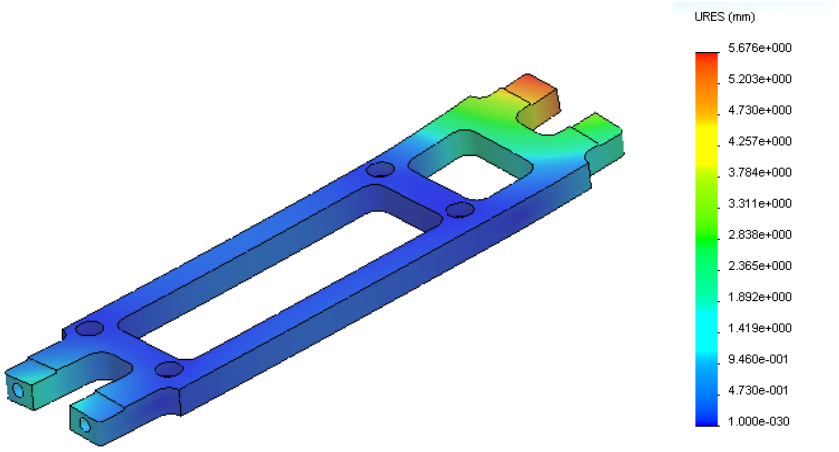


Figure 42: FEA on original Flipper Motor Clamp

Figure 43 shows the FEA conducted on the new flipper motor clamp. This figure shows that there is far less deformaition at around roughly 2mm. This is a significant reduction in the deformation observed previously.

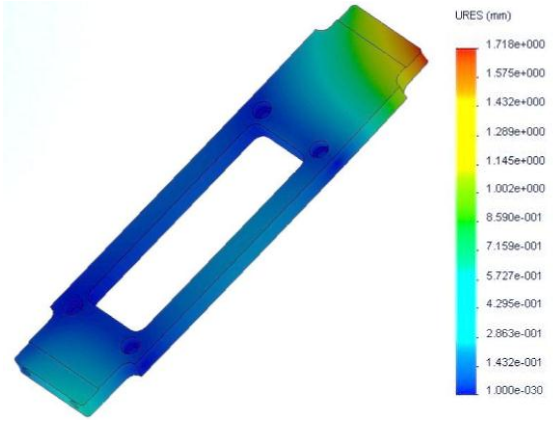


Figure 43: FEA on new Flipper Motor Clamp

The insertion of the new parts reduced the slack significantly to 5-10°. This will give more confidence to the operator to operator the USAR-T with more certainty and overall will improve the reliability of the robot.

2.3 Electronics System

2.3.1 Electronics Stack Sub-System

2.3.1.1 Overview

Figure 44 shows the current configuration of the robot's electronics stack, within the robot chassis, following the recent competition in Germany. The configuration is not vastly different compared to the previous year's configuration, the stack frames remain the same however several key additions, modifications, and removals have been made.

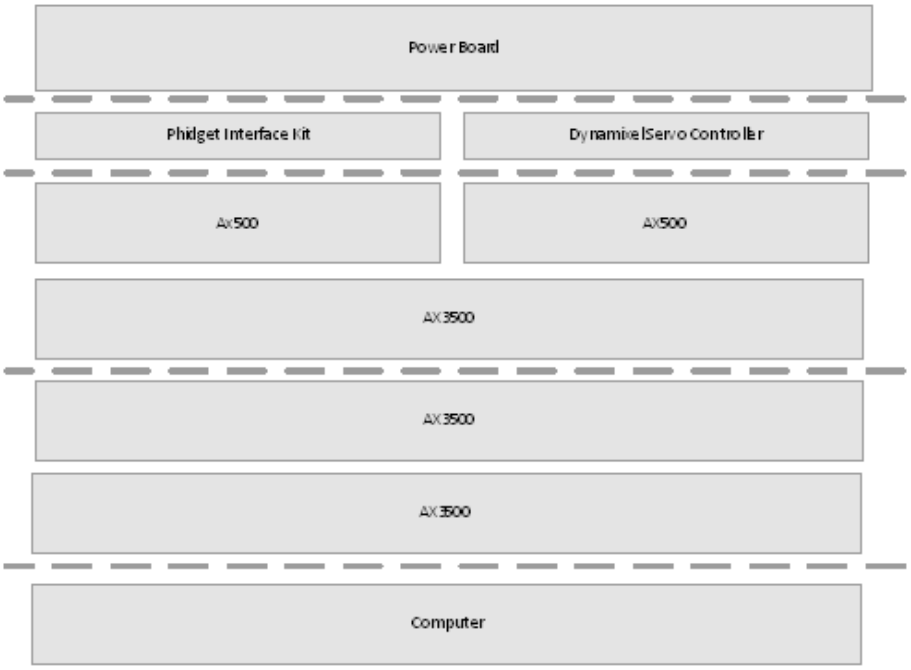


Figure 44: Electronics Stack Layout

Firstly, the computer and power board remain in similar positions within the stack, though the height reached by the power board has been reduced slightly.

The partition that was previously above the computer, where the Phidget interface kit and gimbal accelerometer were located, has been removed. This space has been replaced with the Ax3500 motor controllers used to control the tracks and flippers. Directly above these boards are additional motor controllers, one Ax3500 and one Ax500, used to control the robot's arm. The next layer contains four USB interface devices which had previously been elsewhere or loose within the stack; the Dynamixel servo controller, the Phidget Interface Kit, Battery Monitor, and EasyCap TV card.

Other devices that were previously within the stack that have been removed are now located within the head, such as the Parallax Servo Controller, or have been removed entirely, such as the gimbal accelerometer.

2.3.2 Battery Monitor System

2.3.2.1 Overview

The robot uses two LiPo (Lithium-ion Polymer) batteries; consisting of 6 LiPo cells each, in a parallel configuration as its portable energy source. When using these batteries, it is important to prevent the cells from discharging too far and becoming under-volted. An under-volted cell is likely to lose the ability to maintain a full charge or hold its voltage under load. Like most other LiPo batteries, the batteries used in the robot have a critical voltage level of 3V per cell.

While the charging stations we use are capable of monitoring the battery's voltages they can't be used while operating the robot and previous years teams have unnecessarily broken the batteries due to misuse. Therefore, a new battery monitoring system was required to reliably operate the robot while ensuring the cell voltages do not deplete below the critical level.

Likewise, a reliable monitoring system would prevent the robot from being driven or placed into vulnerable positions when the battery is critically low.

It was chosen to create a warning system for the monitor rather than use the system to terminate power to the robot. Cutting the power to the robot could risk physical harm to the robot, as above, and so it was decided that it would be better to leave such a decision to the robot's operators.

2.3.2.2 Implementation

The battery monitor, Figure 45 has been implemented to provide both a bespoke hardware and software solution, Appendix 7: Battery Monitor Schematic.

The principal component of the monitoring circuit is an AVR ATMEGA8 microcontroller, which is programmed using the C programming language. The ADC (Analogue-to-Digital Conversion) capabilities of the microcontroller allow for a fast measurement of the voltage input from the battery. The result of the conversion is a 10-bit value which is stored on the chip.

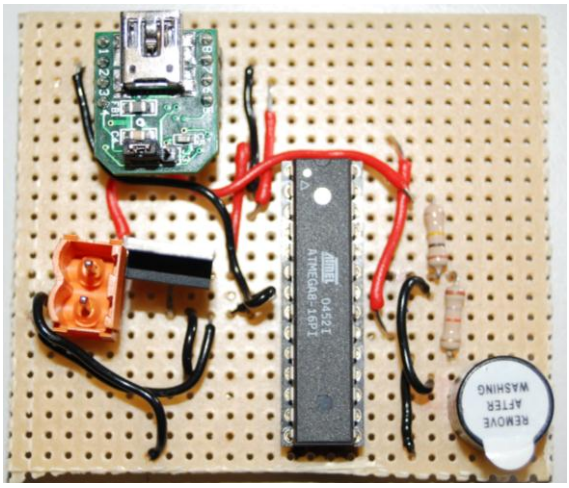


Figure 45: Battery Monitor

As the input to the ADC must be between 0 and 5V the input from the battery must first be shrunk through use of a simple potential divider circuit. Resistor values of 18k and 100k were chosen for the implementation providing an output ~0.15 times the input. This is very safely within the limits of the battery's input, which typically starts close to 26V.

Following a successful conversion, the data is processed to provide a double-precision float responding to the battery's voltage. If the voltage is too low (less than 18.5 volts) a port (PORT B1) is set high in order to sound a buzzer. The inclusion of the buzzer allows for battery monitoring without reliance on the robot's computer being switched on. The UML activity diagram for this section of the battery monitor is viewable in Appendix 7: .

The ATMEGA8 also allows for UART (Universal Asynchronous Receiver/Transmitter) communications between itself and other devices. Using this feature it was possible to implement serial communication between the robot's computer and the battery monitor microcontroller. As the voltage levels differ between the computer and microcontroller a UB232R USB-Serial UART interface was used to bridge the physical connections. The communication method works by polling the microcontroller at regular intervals (by sending a 'q' character). Upon receipt of the poll the microcontroller replies by sending the result of the most recent ADC. At the computer this reply is converted to a value representing the battery's voltage (as on the microcontroller). If this value has changed from the previously measured value then that value is sent to all connected server clients. Appendix 7: illustrates this behaviour using a UML activity diagram.

The source code used to control the microcontroller is available in Appendix 8: Battery Monitor Source Code.

2.3.2.3 Development and Testing

The monitoring hardware and software has been tested throughout its development process. As with other features of the robot unit testing has been used extensively.

After the initial design was made, the circuit was built using a breadboard to hold the majority of the circuit while the microcontroller was held in an AVR STK500 development board. The board provided a simple interface to all the pins on the microcontroller and LEDs are available to verify the operation of different pins as well as powering the microcontroller. Furthermore, the board is the device used to flash the microcontroller with the correct operating program. During the breadboard stage testing was first carried out to ensure that the device was capable of operating in isolation away from the robot and its computer. This implied getting the microcontroller to sound the buzzer when a voltage below the critical level was input. A variable voltage power supply was used as the input to simulate the varying voltage from the battery.

Following the isolated testing, emphasis was placed on the development and testing of the UART serial communications. In this case the monitor was connected to a PC through the UB232R and its USB connection. Using PuTTY (a terminal emulator) it was possible to test the low-level communication that would be used by the robot's computer. Using a keyboard 'q' could be sent via PuTTY to the microcontroller and then a reply sent back to the PC.

The final breadboard stages involved connecting the battery monitor straight to the robot's computer. Using this connection it was possible to test the monitor's serial connection in conjunction with the robot's computer. The java implementation on the computer was unit tested by running it separately from the rest of the program. During this stage the variable voltage power supply remained connected as the input to the battery monitor.

With the circuit design verified, the battery monitor hardware was rebuilt away from the breadboard and STK500 with Veroboard as the base. A single set of pins were used to both measure the battery voltage but also supply power to the monitor's hardware. The power supply line would be pass through a 5V regulator in order to provide the correct voltages for the microcontroller's operation. Alternatively, the UB232R could have supplied 5V from the computer but this would have limited the monitor to only function while the computer is initiated.

2.3.3 Arm Electronics Systems

As in the previous year, the robot's arm is controlled primarily by two Roboteq DC brushed-motor controller boards. Each board is capable of controlling the 2 motors and thus is used to control two joints. The arrangement of the system is such that a high power AX3500

board controls the motors of the joints under most stress (shoulder and elbow) while a lesser power AX500 board is used to control the joints under least stress (base and wrist).

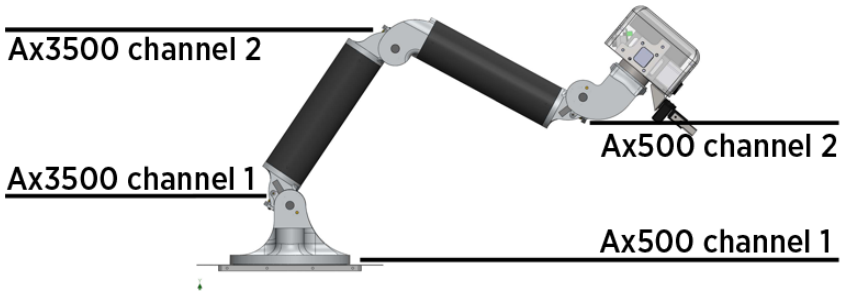


Figure 46: Motor Layout and Channel Allocations

The motor controllers are configured in position feedback mode and feedback is provided by a series of thin-pot rotary potentiometers. The potentiometers are located within the joints of the arm where the resistance (and therefore, sensed voltage) is expected to change based on the position of the joints. A plunger fixed to the "stationary" side of a joint glides along the surface of the potentiometer as a joint is moved causing the resistance and sensed voltage to change. The potentiometers are supplied with +5V supply and from this the sensed voltage is expected to vary between 0 and 5V depending on the joint angle.

Upon initialisation the motor controller boards move the joint to the default "home" position. The position is defined by all the joints returning 2.5V in feedback (the plunger is positioned in the centre of its path along the potentiometer).

In addition to the previously mentioned configuration a Dynamixel rx-64 servo is used to provide yawing of the head. Despite the redesigned head, the principles of operation for this servo have not differed from the previous year's implementation.

2.3.4 Sensor and Communications Systems

Besides the changes to the Arm control hardware, changes to the Manipulator hardware, and addition of the new Battery Monitor the following other additions or modifications were made to the electronic components used by the robot.

2.3.4.1 Xsens MTi

Primarily used in the autonomous robot but also integrated into the teleported robot, the Xsens MTi device provides very high accuracy 3D acceleration, orientation, and heading measurement using on-board micro electro-mechanical systems (MEMS) devices. The device

interfaces with the computers through a USB port and provides real-time streaming information to the computer.

With the inclusion of the Xsens, the digital compass on the autonomous robot has been replaced as the Xsens is more resistant to interference provided by the robot's motors. Additionally, the accelerometer previously used by the gimbal has been removed and the Xsens's orientation data is now used instead.

The MTi is be used as an integral part of the autonomous robot's SLAM and path planning as it provides all the features required for an INS (Inertial Navigation System).

In the case of both robots the Xsens is attached to the outside chassis of the robot.

2.3.4.2 D-Link DIR-825 Router

Upon receiving the pair of robots at the start of this year's project there was only one usable router for the operation of the robot; it became a large priority that a second router be purchased such that both robots could operate simultaneously and without related constraints. Each robot is designed to communicate to any clients wirelessly using the on-board router. Devices on the robot, such as the computer and IP cameras, connect to the router through its Ethernet ports. Due to competition requirements the router must be capable of operating in the 5GHz spectrum using the 802.11a specification.

An alternative to this arrangement was considered where the router was removed from the robot and operated remotely, likely at the client's location. To achieve this arrangement, wireless adapters would have been required for each networked device on the robot. For the computer this would have been achieved using a wireless card but the IP cameras would require Ethernet-WiFi adaptors. Doing so would greatly increase the number of wireless devices that the robot must depend on and it is possible that the devices would require greater storage space. With this in mind it was decided to remain with the standard arrangement.

The D-Link DIR-825 router was chosen to fill this gap and was integrated with the tele-operated robot. The router was positioned outside the robot, mitigating the effects of EMI from the robot's track and flipper motors, and attached between the should and base joints, where it could turn with the base joint but not interfere with any of the arm's possible movements.

The router proved to be very successful and the robot was capable of operating at distances previously not considered possible.

2.4 Software Systems

2.4.1 Restructuring and the development process

The codebase for the robot and its client has grown vastly since the first year of operation. As devices have been added and remove, legacy code remains in the software and adds bloat both in terms of runtime execution and for each year's team of programmer's workload. Previous teams have laid claim to implementing agile programming techniques or object orientation in an attempt to structure and clearly define their programming methodology but have failed in providing a legible, efficient and easily expanded software development methodology which can be maintained for the life of the project.

2.4.1.1 Agile software development

Agile software development focuses on an iterative design process that makes effort to evolve alongside the ever-changing requirements of a software development project. One of the key principles of agile development that allowed this year's team to be so successful in their development is the continuous integration testing (CIT) in the form of deployment of working software and testing it in action and also unit testing.

Agile methodology tends to break down the software development process into small incremental updates. For each incremental update, a full development cycle is followed. This begins with planning which also includes analysis of the requirements or "use cases" when defined using the Universal modelling language (UML) specification. This gives the team both a specification to work towards but also a set of test cases that are useful in the final stages of development. The software is then designed and coded, perhaps being drawn up in a UML diagram before finally writing the functional software. After this step, unit tests are conducted; ensuring this element of software meets the aforementioned requirements. Finally, acceptance testing is performed, ensuring these incremental updates function well with the rest of the software as a whole. This is not the same as user acceptance testing which would test the software with the end-user.

2.4.1.2 Unit testing

The unit test is fundamental to ensuring the functionality of the code without testing every possible usage situation. Unit testing also allows a developer to check code written before

them is still functioning after their changes. Unit tests are supposed to test the smallest possible element of the software and are generally directly descended from the requirements analysis performed at the beginning of the software development cycle.

Another useful application of the unit test is in fixing problems identified with software. It can be considered good practice to develop a unit test for each problem (or "bug") fixed in the software to ensure the problem will not appear again with future development.

2.4.1.3 Object orientation

Object orientation is not just a feature of the language the robot is programmed in. Object orientation improves code readability and expandability, allowing developers to encapsulate the functionality required by one element of the software whilst hiding its exact implementation from other programmers. Decoupling also allows for the separation of objects into distinct layers within the software and is facilitated using object oriented principles.

Previous software developed by WMR attempted to include large amounts of code within single classes, leading to a large amount of repetition between classes. By separating out different elements of functionality into smaller, re-usable elements of code the program as a whole is much easier to follow and is far easier to extend.

2.4.1.4 Model View Controller

The client software makes efforts towards following a distinct model-view-controller (MVC) architecture. The model, in this context, refers to the underlying data structure in the software. This might be the data from the sensors from the robot or even the data messaging system required to send and receive messages to and from the robot. The view is the graphical representation of the model to the user and allows interaction. Finally, the controller negotiates the actions between the view and the model.

MVC is beneficial in debugging software as it allows for much quicker diagnostics. If the user interface is not being displayed correctly, the problem will lie in the View. If incorrect data is being displayed or transmitted, it is in the model or controller. It also means there is no ambiguity between what or how various user interface elements perform the same task such as messaging the robot. A prime example of this is the multiple methods through which a user may control the robot's arm. Originally, each user interface element would send the message to the robot, meaning multiple locations for errors in messaging and many places to change the program if message format should change. Instead, by centralising the arm control into one class and linking the various interface elements to this class.

2.4.1.5 Unified modelling language

Although unified modelling language (UML) is not used in the development of all aspects of the software, it proved invaluable in planning the structure of control systems and program flow in the more complex areas of operation. UML provides engineers with a standardised language with which to model systems and share ideas amongst other engineers. It also allows future WMR teams to more easily understand the workings of control systems within the software without having to read through all of the code. It was most useful in the development of new ideas such as the notification system in the robot and in modelling the flow of messages between the robot and the server.

2.4.2 Robot Server Sub-System

2.4.2.1 Architectural Changes

The general architecture of the robot software has been refactored significantly. Major goals of these changes have been removing code duplication, decoupling objects, and increasing the encapsulation of objects. If these goals are achieved the result should be a more stable program, which allows for easier development and refactoring at points later into the project's lifetime.

A major effort of this redesign has been placed in splitting the code required for low-level communication and direct control of devices from the public interface and any abstracted action that might determine the behaviour of device. This approach improves encapsulation between objects as changes to the communication methods will no longer affect, or break, the methods used by other objects; therefore, reducing the number of changes required to only the device's abstraction layer. Using this approach different physical devices performing similar actions, e.g. IMUs, could be interchanged with only the low-level communication class requiring modifications.

These features were implemented as a Profile-Driver system where each Profile class acts as the abstract layer and is intended for use as the public interface to all other objects. The Driver class performs all low-level operations such as connecting and communicating with the physical device. The Profile-Driver relationship is modelled as a "has-a" relationship where the Driver object is private member object within the Profile class. Currently, the system usually instantiates Driver objects during the construction of the Profile object, though the initialisation of both is usually delayed until later.

Figure 47 UML Diagram of the profile-drive relationship

Where possible it has been undertaken to remove unnecessary, repeated or deprecated code in order to reduce the size of the source code but also to reduce the confusion that might be caused when working with repetitious and erroneous code. Deprecated code usually appears as previously used devices and systems that should be removed from the source. For example, at the start of the project there were two class created for the purpose of controlling the arm: one for controlling the current model, and another for controller a previous servo model. Initially, the inclusion of the servo arm controller had caused confusion, especially due to the indecipherable methods it took and lack of comments, but was eventually removed. Other deprecated code included implementations for Sonar and IMU devices from the first year of the project. The implementations for the track, flipper and arm low-level communications, Drivers, were a source of very repetitious code. All three features used very similar controller devices only with minor configuration differences and using a typical object orientated programming "is a" model it was possible to implement a simple base class and derived class system as shown in Figure 48.

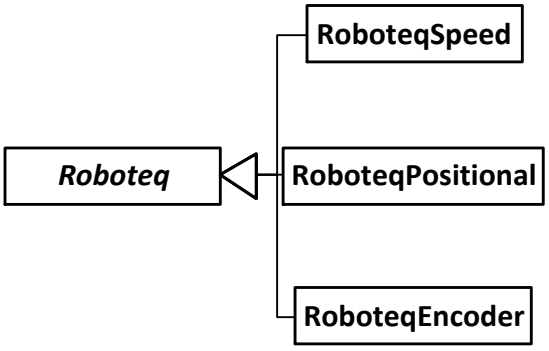


Figure 48 Inheritance diagram for Roboteq controllers

Originally, all devices were initialised at the start of the program in a queue along with the server and console, as part of the construction of a robot object. Should one of the objects not initialise quickly the rest of the program is delayed. To mitigate these effects, and ensure the server is initialised promptly, the devices were separated into related handling classes defined by subtypes, e.g. sensor, vehicle or health. The device handling classes run on separate threads and are designed using the Singleton design pattern, making their interfaces globally accessible but only allowing a unique instance of each class.

2.4.2.2 Improved Command and Messaging System

Messages and commands can now be passed directly from the keyboard to the robot program when it is run within a terminal window, either locally or remotely, using a SSH client. The system uses a console window to read keyboard input and then provide output based on the input's validity. The new console greatly aids development as it allows for areas of the robot to be tested and modified without reliance on the client software. Features can be improved and tested without having to redevelop sides of the client at the same time. This aides in development time as the developer will only have to modify the client once, when the robot side has been completed rather than continually.

The console is developed to process the same messages as used by the server's low-lag connection server (LLCS). For this to work efficiently the LLCS had to be refactored slightly. Previously, the LLCS operated by processing messages within its own class' methods. Should a different class' interface change, or the class was removed altogether, the LLCS processing management methods would need to be fixed to remove or update the messages that correspond to that class. The new system now relegates message processing to the actual class that the message or command attempts to control or interact with. Each class that needs to handle message processing, such as device Profile classes or the Server class, inherits from a java interface, Message Parser, defined to provide the processing method. Each inheriting class then overrides this method to provide their related functionality. Each message processing class registers itself in a list that is iterated through, and methods executed, every time a message is received, either through LLCS or the console.

Additional improvements have been made to improve the legibility of command messages by using messages more closely related to the process intended to be performed. For example, old flipper commands were usually sent in a form like "rfx:1000", which, without reading the source code, would not be relatable to that actual physical action, moving the rear flippers by 1000 steps. Improving on this the current messaging can now take messages of the form "flipper rear=1000" which performs the same action but provides a more clear picture of the action to be performed.

2.4.3 Arm Control System

2.4.3.1 Software Control System

2.4.3.1.1 Overview

The control software for the arm has been almost completely redeveloped this year to provide additional features and greater reliability for the use of the arm. Furthermore, the core framework for the system has been designed in such a way that changes can be made to the physical arm with only minor changes then required to the codebase.

2.4.3.1.2 Features

In addition to changes to its framework the new system also includes the following features:

- A reliable inverse kinematics solution
- Real-time positional feedback to the computer
- The possibility to queue movements between positions in space
- Translations of the head in Cartesian space
- Linear interpolation for arm movement

2.4.3.1.3 Framework

The framework has been designed to provide an abstracted version of the physical arm. The 'ArmProfile' class contains a number of 'Joint' objects designed to represent the physical joints. Each 'Joint' object attempts to mirror the physical joints by interfacing with its related controller board and motor channel. Each Joint object contains information on a joint's state, such as its current position, goal position and whether the joint is currently moving.

By using this design later redevelopments should be made simpler and time spent mitigated. Physical changes need only require changes to the types or number of instances of joint object used.

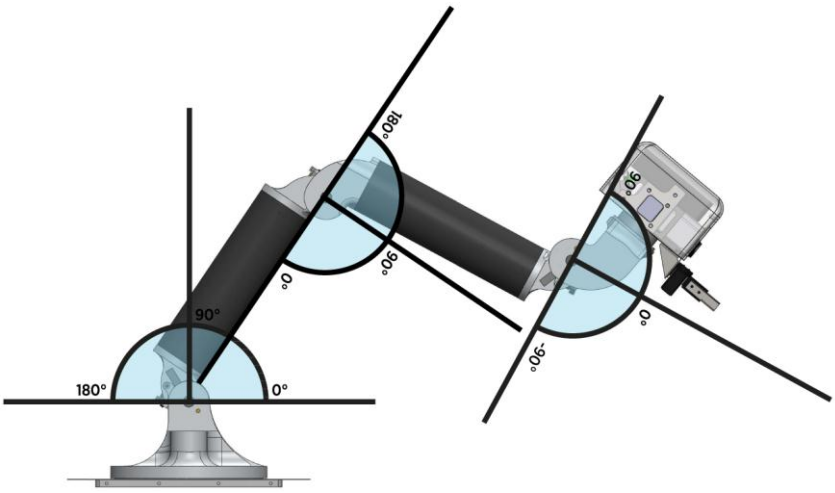


Figure 49: Permissible Angles, Side projection

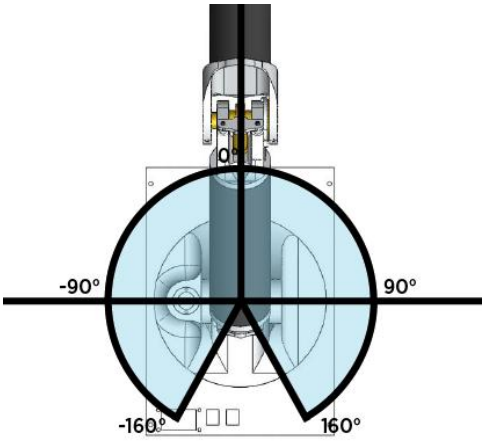


Figure 50: Permissible Angles, Top projection

Figure 49 and Figure 50 show the permissible angles that can be made by the arm's joints. To gain a simple representation of these angles within the software some formatting and offsets are required with the source.

Positions are sent to the motor controllers as 8 bit numbers representing a full 360 degrees of motion. Therefore, before sending a 90-degree position it must be first converted to its related 8 bit number.

$$position = 90 \times 256/360$$

Additionally, offsets must be introduced to ensure that the joints move to the correct position. Offsets are required due to the positions of the potentiometer and their plungers not being parallel or perpendicular to the base of the arm. For example, the shoulder joint requires a 75-degree offset due to the position of its plunger.

$$relative\ position = 75 - \ actual\ position$$

2.4.3.2 Kinematics

The previous iteration of the arm control software was only capable of providing joint angle kinematic control. Some attempts were made to introduce IK (Inverse Kinematics, where input Cartesian arguments will produce a valid set of joint angles) but behaviour was found to be unreliable. Therefore it became an important goal for this project to provide a valid and reliable solution to this problem.

Initial steps were taken and a solution was found using the previous year's arm. Conveniently, due to only minor changes in the arrangement of the arm (the degrees of freedom were unchanged) it was possible to transfer the solution into use on the new arm with only minor revisions necessary.

2.4.3.2.1 Inverse Kinematics

The inverse kinematics problem is solved using largely trigonometric methods to find the angles made by the three lower (base, shoulder and elbow) joints. The two upper joints (head, neck) do not need to be included in the calculations as they have no significant impact on the end position in space, only its view direction.

Parameters are passed to the method as a 3 dimensional vector object, of form $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$, or as individual arguments. The method is developed such that an input vector $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ will result in the arrangement seen in Figure 51. The lengths of the members of the arm, equal in size, are each treated to be of length 1.

Figure 51: Positional Result of a [0 1 1]' Input vector

Following the method, the first angle to be calculated is the base. The formula for finding the base angle in radians is written as:

$$\theta = \frac{x}{|x|}\left[\frac{\pi}{2} - \frac{z}{|z|}\left|\tan^{-1}\frac{z}{x}\right|\right]$$

The angle is defined by the vector made by x and z as illustrated in Figure 52. As the arm is capable of moving in negative z space using $\tan^{-1}\frac{z}{x}$ is not adequate to supply an adequate angle, when dealing with negative x, z, or both. The additional parameters shift the angle into the correct range.

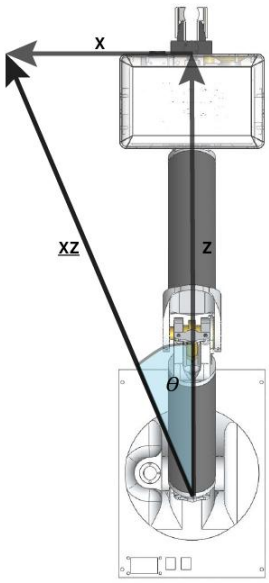


Figure 52: Arm XZ Vector

The final two angles, elbow and shoulder, are found using triangles constructed by the arm member lengths, l, and the length of the vector made by x, y, and z. Using the cosine method it is then possible to find correct angular positions for each point. Figure 53 shows the configuration for the arm through which these angles may be evaluated.

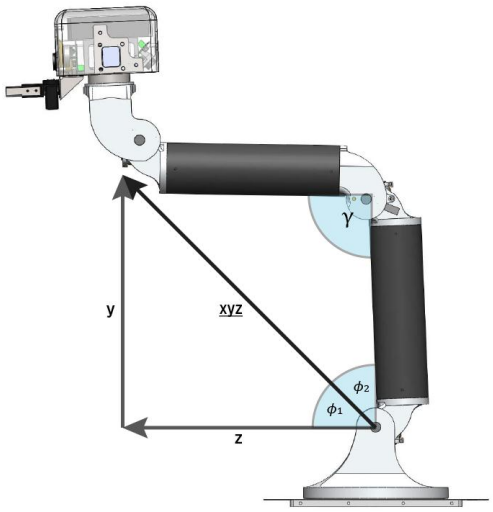


Figure 53: Evaluation of Arm Angles

For the shoulder joint the equation for evaluating its angle is:

$$\phi = \cos^{-1}\left[\frac{\left(\left|\begin{matrix}x\\y\\z\end{matrix}\right|^2 + l^2 - l^2\right)}{2l\left|\begin{matrix}x\\y\\z\end{matrix}\right|}\right] + \sin^{-1}\left[\frac{y}{\left|\begin{matrix}x\\y\\z\end{matrix}\right|}\right]$$

$$\phi = \cos^{-1}\left[\frac{\left|\begin{matrix}x\\y\\z\end{matrix}\right|}{2l}\right] + \sin^{-1}\left[\frac{y}{\left|\begin{matrix}x\\y\\z\end{matrix}\right|}\right]$$

Finding the elbow joint's angle is far simpler for which the equation is:

$$\gamma = \cos^{-1}\frac{l^2 + l^2 - \left|\begin{matrix}x\\y\\z\end{matrix}\right|^2}{2l^2}$$

Once all three joint angles are found they must be converted to degree form and are returned from the method.

2.4.3.2.2 Forward Kinematics

Forward Kinematics provides a Cartesian output when considering a set of joint angles. Such an operation is not only useful for the testing and validation of the IK but is also is used in conjunction with the positional feedback to provide the current position of the arm in Cartesian form. Such a feature is necessary for use in translational movements or for linear interpolation.

The implementation used in the program uses linear algebra methods to find the solution for the supplied joint angles. As with the IK solution, this method only focuses on the angles made by the three lowest joints in the system (base, shoulder and elbow). The equation representing the solution is definable as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \left(1 + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi-\beta) & -\sin(\pi-\beta) \\ 0 & \sin(\pi-\beta) & \cos(\pi-\beta) \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi-\alpha) & -\sin(\pi-\alpha) \\ 0 & \sin(\pi-\alpha) & \cos(\pi-\alpha) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Where α is the angle made by the shoulder joint, β is the angle made by the elbow joint and γ is the angle made by the base joint.

The method begins by constructing a vector of length $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ which is rotated along the y-axis by π minus the shoulder angle. The vector is duplicated and rotated a second time along the x-axis by π minus the elbow angle. The two vectors are summed and converted to a point in spaced. A final rotation is then performed on the point along the y-axis that is equal to the value of the base angle.

2.4.3.3 Real-time Positional Feedback

The new control system has been designed to provide positional feedback in real-time to the robot's on-board computer. This is achieved using the serial features of the Roboteq motor controllers where it is possible to query the value of their analogue voltage inputs (from the thin-pot rotary potentiometers). Reading the voltages it is possible to then derive a joint's angle.

2.4.3.3.1 Method

The voltage is polled by sending '?e' serially to a controller. The controller responds with the voltages supplied as signed Hexadecimal numbers where -127 represents a 0V reading and +127 represents a +5V reading.

$$voltage = 2.5 + \left(\frac{hex\ input}{128}\right) \times 2.5$$

Using the known parameters of the arm in respect to the potentiometers it is possible to derive a joint's angles. An angle-offset constant is included in the calculation as the position of the potentiometer plunger is not immediately beneath a limb when the arm is homed.

$$position = angle\ offset + \frac{voltage \times 360}{5}$$

2.4.3.4 Other Features

2.4.3.4.1 Queue System

The queue system allows for the arm to move through successive arm positions. The system reads positions from a list where the current goal position is the first element in the list. When the arm reaches its goal (as verified through feedback) the target is popped from the front of the list and the system moves on to the next element as its goal.

Single or series of positions may be added at a time or the queue can be overridden with new sets of positions.

2.4.3.4.2 Linear Interpolation

Linear interpolation through use of several of the features already detailed. Linear interpolation causes the arm to move along a straight path from one position to the next.

First, positional feedback provides the arm's current position. From this information forward kinematics is performed to provide a Cartesian representation of the arm's current position. Using this Cartesian position and the supplied target position a unit direction vector is computed. Using this vector intermediate positions are computed between the current and target positions, passed through the IK method and queued (using the Queue system) so that the arm can move along a straightened path.

2.4.3.4.3 Translation

From the arm's current position and after forward kinematics are processed, translation can be easily be performed on the output with supplied x, y and z parameters. The translated outcome is then subjected to the IK method to provide the required output of joint angles.

2.4.3.4.4 Testing

The systems has undergone extensive unit testing where possible largely with respect to the kinematic methods. It was very important when developing the inverse kinematic method to perform rigid unit testing before using the solution on the physical arm. Should the method find a non-valid solution while using the arm there is a strong risk that the arm might try to move a joint past its critical angle limit. Alternatively, it might cause the arm to collide with the rest of the robot's body.

2.4.4 Manipulator Control System

2.4.4.1 Overview

The manipulator added to the robot is controlled by a HS-422 servo which in turn is handled by the Parallax Servo Controller already included within the robot. The Parallax Servo Controller allows support for up to 16 different servos. However, the code written to operate the Controller through the computer was written specifically for the LiDAR gimbal, the 'Gimbal' class, and only made reference to the servos used by it. Merely patching additional code into this class would not benefit the robot in the long-term; should other servos be required at a later date the control class would not be very fit to include them. As such the control class was rewritten and code specifically for the gimbal and manipulator were separated to other classes.

The new 'Parallax' class was designed to hold a series of 'Servo' objects, which would be registered to the class by different 'Servo' handling objects ('Gimbal' and 'Gripper'). Each 'Servo' object is given a specific servo id, related to the ports on the controller, and contains information on the physical servo's goal position. By regularly iterating through registered 'Servo' objects the 'Parallax' detects whether the goal position for the servo has changed since the previous iteration; if this is the case then the Parallax will send an updated position command to the physical controller.

For the manipulator, activating buttons on the client software, which sends a relevant command to the robot, can perform gripping and un-gripping actions. Upon receipt at the robot, the 'Gripper' object inputs the related position to the 'Servo' object, and at the next 'Servo' list iteration by the 'Parallax' the new position is sent to the controller.

2.4.5 Client Software Sub-System

2.4.5.1 Human-Robot interface improvements

The client software graphical user interface and robot control system has been redesigned from scratch after finding the old software to lack continuity and run slowly at times. Redesigning the interface would give the chance to develop a sleek, unified solution to controlling the robot rather than broken, multi-part affair that had been used previously.

2.4.5.1.1 Requirements analysis

The main requirement of the user interface is to display all the information collected by the robot to the operator in such a fashion that is clear and easy to understand. The control systems in place on the client software should give the operator a natural method by which to control the robot that they are used to and requires minimal training to use. By studying the previous client software and the capabilities of the robot, the key pieces of data and elements of control the operator required were identified.

Data	Control
CO2 Sensor data	Independent tracks
Webcam data (x2)	Arm joints
IR camera data	Arm kinematics
The current "pose" of the robot	Two-way communication
The average "heat" from the IR camera	Gripper
Text feedback from the server	

Table 3: Identified data and control required to by the operator of the robot

The previous client software lacked a function to display webcam data or the average heat read by the IR camera. It also did not have arm kinematic control, independent track control or any supporting methods for the gripper as the previous robot did not have one. It was decided that some of the old code should be reused for continuity in communication between the robot server and the client but that processing of the data and displaying it could be changed.

2.4.5.1.2 Implementation

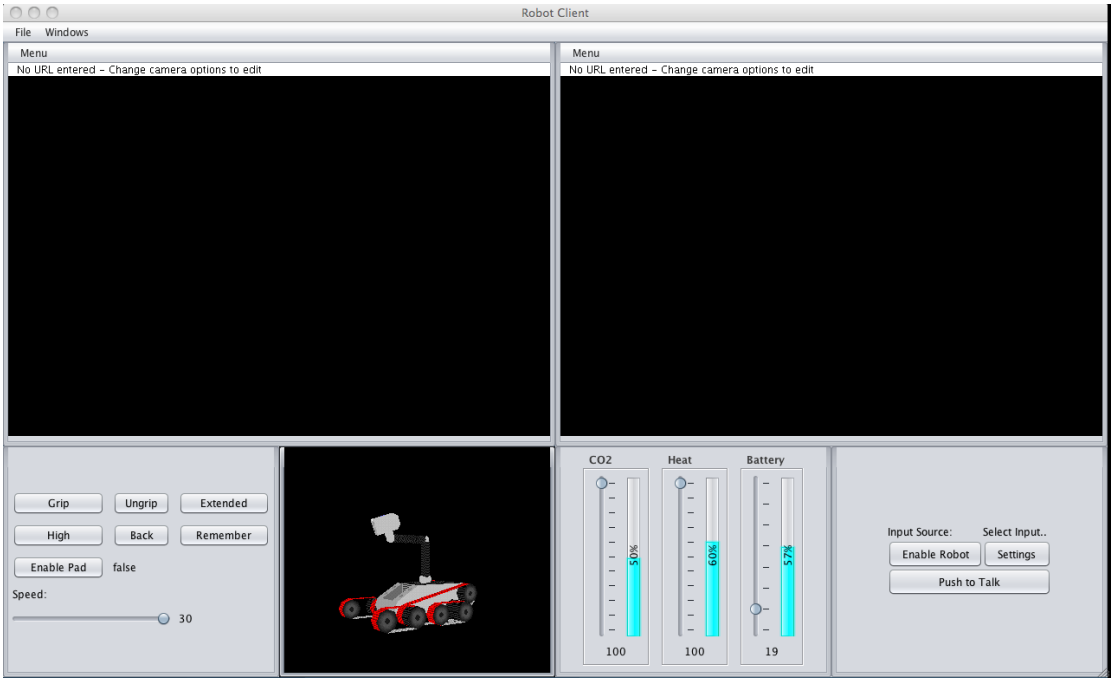


Figure 54: The new human-robot interface as seen by the operator

2.4.5.1.2.1 Webcams

The general layout of the display was quickly settled upon, allowing two-thirds of the vertical height of the screen to display the webcam data. This is a logical choice as the webcam data is the most important element of the user interface when controlling the robot. The two webcam images could comfortably fit on a 15 inch display side-by-side with just enough height to fit the entire image. Previously, two web browser windows had to be open alongside the client software, resulting in a poor user experience and a much longer set-up time. It was also decided that the IR image data was not necessary to view all the time and as such it could be switched with a webcam image panel when it was necessary to check the IR data.

2.4.5.1.2.2 Robot pose

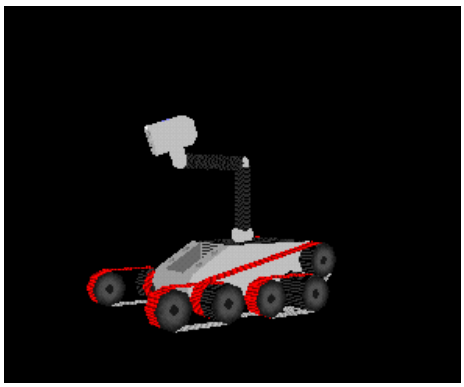


Figure 55 The 3D pose representation presented to the operator

The robot "pose" describes the current joint and flipper orientations, improving the remote perception of the operator who is unable to see these angles through the webcams. Previous years had completed a 3D model of the robot that could be manipulated in software, providing an excellent method to visualise this data. The same code was used this year but a few changes were made. The initial view of the robot was at an incorrect angle and so a transformation matrix was established to provide a correct initial viewing angle. The background was also stripped from the scene, allowing the operator to concentrate solely on the position of the robot.

2.4.5.1.2.3 Gripper, arm presets and speed control

Figure 56: Gripper, positional and speed controls

The gripper control is a simple two-button system. Arm presets are also tied to buttons on the user interface and are labelled with a helpful name to remind the operator of their action. Speed control was added in later revisions of the software after it was found that fine-grained control and speed were necessary in different areas of the competition arena. This quick-fix is a simple slider which can be moved to change the scalar value that is applied to the track control discussed later.

2.4.5.1.2.4 Signs of life panel

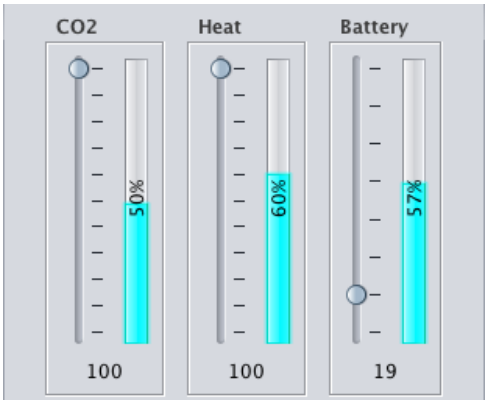


Figure 57: The Signs of life window with CO2, Heat and Battery levels

The “signs of life” panel provides quick feedback as to the status of the robot and its sensor readings. It is a collection of bars that fill up depending on the reading of the sensor. The code was later extended to include a slider on the left hand side that allowed the operator to select a “limit”. When the reading crossed this limit, the software would notify the operator using the notification system. This allows for dynamic readjustment of the sensor triggering levels depending on ambient temperature or CO2 readings. The battery status was also included in this panel and has a slider normally set at 19 volts. The battery status is a critical factor in determining how long the robot can run for and this element is one of the most noteworthy improvements on the system from last year.

2.4.5.1.2.5 Two way communications

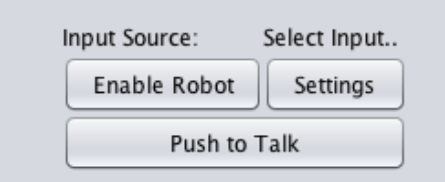


Figure 58: The communications panel

Explained in more detail later, the communication panel allows the operator the start and stop recording their voice for transmission.

2.4.5.1.2.6 Robot console

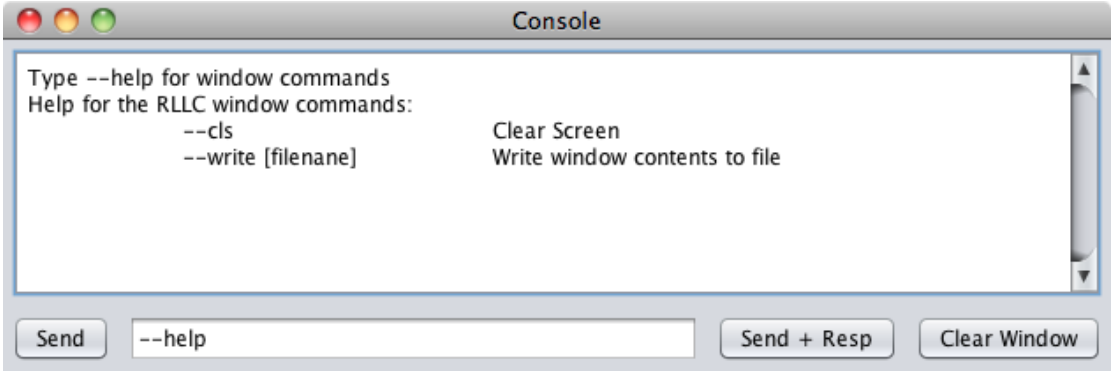


Figure 59: Improved console window with help functionality

The console maintains the functionality of the previously utilised “Robot low-lag connection” (RLLC) console from previous years. Functionality was added to include a command history as well as interactive help. This help functionality is easily expandable to allow future years to write down all RLLC commands they will implement so that any operator may have a quick reference to the commands whilst driving.

2.4.5.1.2.7 Game controller

The game controller code was re-written from scratch. The new code provided a much simpler way of programming new commands into the controller depending on what button was pressed when the game controller was last polled. This should enable future years to quickly adapt the code to suit their needs rather than using the old control structure that required an intimate knowledge of the controller hardware to correctly program. These layers of abstraction introduced by this year's development team also introduced great robustness into the system and led to a more predictable behaviour from the controller. Each analogue stick was assigned to each track on the robot, providing control of the individual track speeds and, as such, enabling a gentler turning radius and smoother driving experience to be achieved. The new system also responded to the amount of movement the operator gave to the analogue stick. This resulted in a fine grain control possible on the controller and a harsher control by using the speed slider on the user interface. The individual, variable track speed control was a vast improvement over the previous control system.

2.4.5.1.3 Testing

Unfortunately, some of the elements of the user interface were unable to be tested, as the functionality was not ready on the robot. The infra-red webcam was broken in the weeks leading up to the competition, meaning that the IR stream could not be tested in the webcam windows and an average heat reading could not be calculated and shown in the signs of life panel. The arm kinematics could not be implemented as the physical arm was not manufactured until relatively late the therefore the corresponding interface program on the robot server was not completed either. The client still functioned flawlessly, however, providing effective visual feedback from the battery monitor and CO2 sensor, complimented by the notification system, discussed in further detail later in the report.

The new control code that works with the game controller also allowed for much greater control over the movement of the robot. Thanks to an adjustable speed scalar and independent track control it was much easier to manoeuvre the robot in situations where it was not possible previously. The new system of control enabled a more elegant method of traversing the more extreme terrain by individually controlling the tracks and gradually edging over obstacles. This gave the operator much more control over the location of the centre of gravity of the robot relative to the competition arena.

2.4.5.2 Notification System

The notification system is an entirely new concept for the USAR-T3.5. In a discussion with PhD students that were present at the 2010 competition in Hannover it was understood that

information was not presented to the user in a timely fashion or required them to try and process vast quantities of data displayed to them on the screen at once. Interest was expressed in some form of notification system so that the operator of the robot could focus on the mission critical control of the robots movement through the course. There has been extensive research into Human-Robot interfaces, attempting to minimise the mental strain on the operator of the robot so they do not become fatigued too quickly. According to (Chen, et al. 2006) the problem that faces most tele-operators of robots is the "remote perception" that is provided to them by the software interface.

2.4.5.2.1 Requirement analysis

The operator needs a clear indication of whatever error is being reported and, as such, the notification system should display a short, titled message with some sort of visual indicator of event severity. Colour is the most obvious choice as an indicator of urgency. The notification system should be able to display multiple notifications on the screen at one time should many things go wrong. As this system could be seen as vital to the operation of the robot if it is reporting faults with the robot itself or the connection, it should run in a separate thread to the other robot software, ensuring it will run if something else fails.

2.4.5.2.2 Implementation

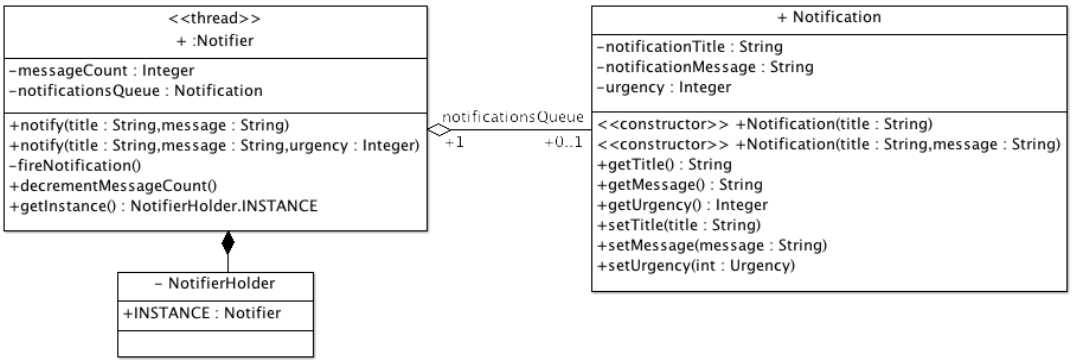


Figure 60: UML class diagram of the notification system

The notification system consists of a notification dispatcher, a notification model and then the display views for the notifications. The idea is that anything can produce a notification on the screen simply by passing the notification manager a string containing its message as seen in Figure 60. The upshot of this is that, once the code is written, nothing needs to be added to it, other programs just need to then send the correct data to the notification manager through its public methods.

The implementation is graphically somewhat unattractive at present but it performs the task to an adequate standard. The most interesting application of the notification will be the integration with sensor data in future systems. Rather than just notifying the operator of sudden changes, a more intelligent notification system could, instead, monitor the environment over time and report if it 'thinks' a victim is near rather than the minutia of the situation (such as individual sensor readings). The immediate benefits of a notification system include increased awareness of the operator and an increased reaction time should something need attention.

2.4.5.2.3 Testing

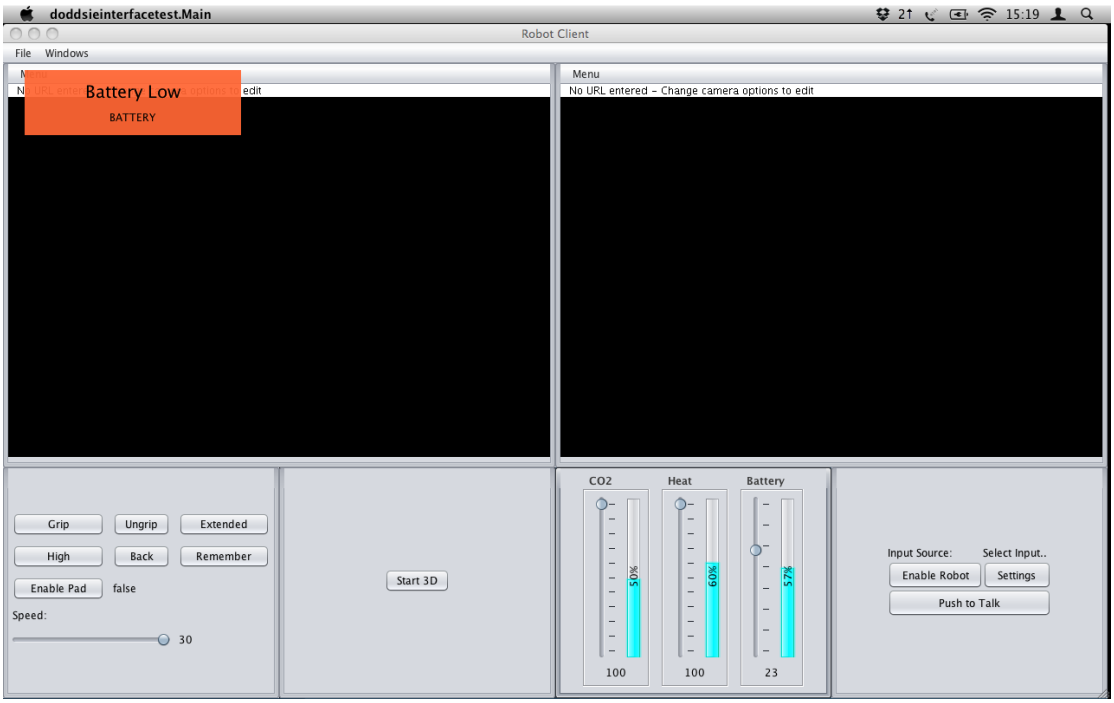


Figure 61: The user interface with red notification popup in the top left corner.

The battery monitoring code implemented on the robot notifies the client of the battery voltage at regular intervals. On the client side, this information is presented graphically to the user and can be set to use the notification system to provide a warning when the battery voltage falls dangerously low. The notification system worked flawlessly presenting this data for the first time during the competition in Magdeburg. While operating the robot, the operator neglected to note the rapidly decreasing battery voltage presented in the user interface. It was only when the notification popped up that the operator was made aware and the battery could be changed, avoiding dangerous under-volting of the batteries.

Unit tests were also created and ran to check that notification methods ran smoothly and to ensure the "Notifier" itself correctly incremented and decremented its notification count when it was displaying and destroying notifications.

2.4.5.3 Webcam integration

Previous years have used a web browser to view the live webcam streams from the robot, requiring extra programs to be open with limited control in terms of image processing. By integrating the live webcam feeds into the client software, a more unified control methodology has been created. Writing software that only works for one specific webcam is not useful and would be a poor starting point for future groups should they upgrade the imaging equipment. It is with this in mind that an extendable set of code has been put together for future teams to utilise. Complimenting the code is a set of unit tests that ensure that new code changes will ensure the basic functionality of the viewer is still in place.

2.4.5.3.1 Requirements analysis

The webcam view should display the stream of images generated by the webcams mounted on the robot. It should be easy to extend the view to allow different webcams or video devices to be incorporated into the same window, allowing for a more unified experience. The view should provide feedback to the user while the software is running so the user may know if the webcams have successfully connected or if the stream has been lost. There must be a simple method of reconnecting to the webcam in the event of a broken connection. The webcam view must be easily configurable to choose different addresses for different webcams on the robot.

2.4.5.3.2 Implementation

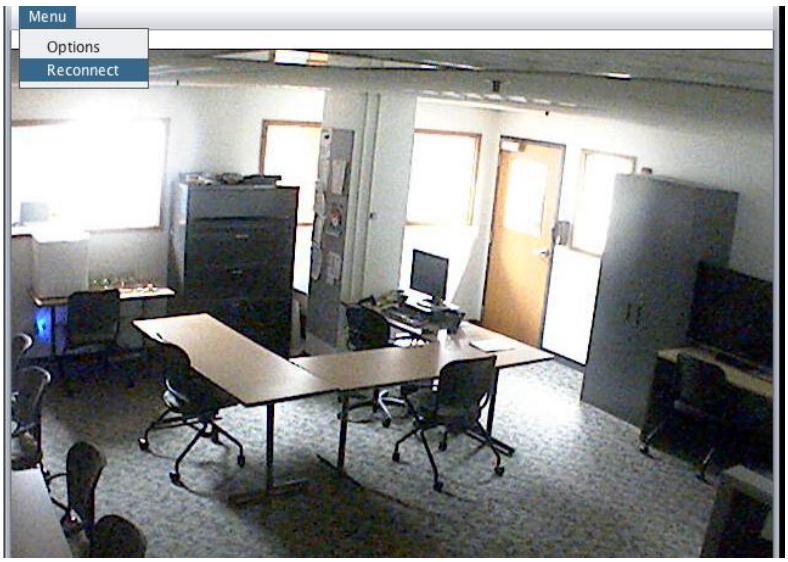


Figure 62: Screenshot of the Webcam interface and a test webcam stream

The webcam view renders a continuous image stream provided by the IP cameras. Clearly, using the processing power of the camera to generate and stream the images is advantageous to performing the task using the on-board computer as it does not use any of its processor cycles. There is also the advantage of only accessing the data from the cameras when necessary rather than having the camera broadcast which takes up significantly less bandwidth when the cameras are not being viewed.

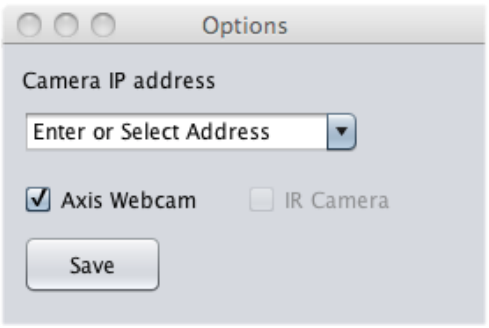


Figure 63: Screenshot of the Camera Selector

Users are also provided with a simple screen (Figure 63) to choose between the different cameras available from programmatically determined addresses. It is the intention that users will then be able to create their own addresses for cameras and add them to a stored database.

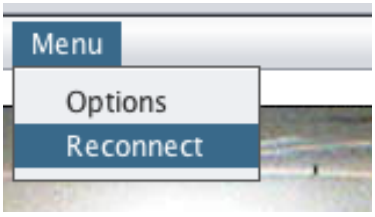


Figure 64: The reconnection option in the webcam view menu

There is also functionality to allow for reconnections to be made to the camera in the case of connection dropouts, a frequent occurrence at the competition. The problem with connection dropouts is that the stream information becomes disjointed and it is hard for the program to realise where the images begin and end. For this reason, a reconnection option was also built into the program structure. Allowing the current connection to the server to be abandoned and then re-established when the wireless link becomes available again. The MPEG headers could be monitored for in the incoming stream when a connection is dropped but a simple reconnection scheme allows for a much simpler implementation.

2.4.5.3.3 Testing

Prior to the competition, this implementation was tested both using the 802.11a protocol used at the competition and 802.11n. 802.11a tends to have a lower range than 802.11n as it is more readily attenuated by walls but does not have the same spectral interference as it uses the 5Ghz band as opposed to the relatively crowded 2.4Ghz. 802.11n can use both of these bands and represents the latest series of routers and performance using this protocol can be seen as a best-case scenario. Previous WMR teams had mentioned problems with low connection speed and noticeable time delay in the webcam streams but this was not the case with the new system. There was also the added benefit of having all of the necessary information presented to the operator within the one client application. There was a notably lower set-up time in getting the software running and there was a reduction in overhead on the client machine as fewer programs were running.

The reconnection was also attempted multiple times both on the Axis webcams mounted on the robot as well as a multitude of different, publicly available IP cameras to ensure the reconnect function worked flawlessly with both the current cameras as well as others available on the market.

Another useful function is the text-based feedback seen in the white bar at the top of Figure 62. This allowed for the diagnostics of problems during operation in the competition. It allowed the operator to ascertain if the webcam was still online but suffering a software fault or if the network had disconnect. It provided an early warning sign for a larger communications failure between the client and the robot server.

2.4.5.4 Two-way communications

Previous years' attempts had not included two-way communications between client and robot. This was a key milestone achieved this year as it allows a more immersive driving experience for the operator. Sound clues are given off by many of the victims including crying or even the sound of their animatronics whirring.

2.4.5.4.1 Requirements Analysis

Two-way communications should provide the clear transmission and reception of speech data between the robot and the client. It should place minimal load on the network to avoid congestion and blocking the mission-critical control commands from being sent. The communication system should allow for changing audio quality settings to ensure optimum transmission speed. To avoid feedback, the system should operate with "push-to-talk"

functionality as with walkie-talkies. It would be nice if the system could dynamically adjust the compression settings to minimise network load but this is not a vital requirement.

2.4.5.4.2 UDP

Real-time information that also requires large amounts of bandwidth to send, such as sound, does not function well using the Transmission Control Protocol (TCP). In TCP, packets sent must be collected and then placed in order when they arrive at the client computer. Packets containing audio data can grow to be extremely large even using high compression techniques and having to store and re-order them can lead to large delays while waiting for packets to arrive. The User Datagram Protocol (UDP) is advantageous in this situation because there is no requirement for packets to be ordered and lost packets are just dropped. In audio transmission, you can cope with losses or slightly "choppy" signals from disordered packets because you don't need all the audio data to comprehend what is happening.

2.4.5.4.3 Implementation

The two-way communication system was implemented rapidly and is still in its infancy. Clearly there is much effort to be had in separating out the different classes to provide separate audio capture, transmission and playback functions as well as improving the method by which the code operates. Currently, the addresses of the receiver and transmitter computers are hard-coded into the software. The receiver will play anything it receives from the transmitter address and will continually be "listening" for such information. The Transmitter starts and stops the "MicrophoneCaptureThread" depending on whether or not the user has clicked the transmission button. At which point the thread is created and the information is streamed to the receiving computer that subsequently plays the audio data.

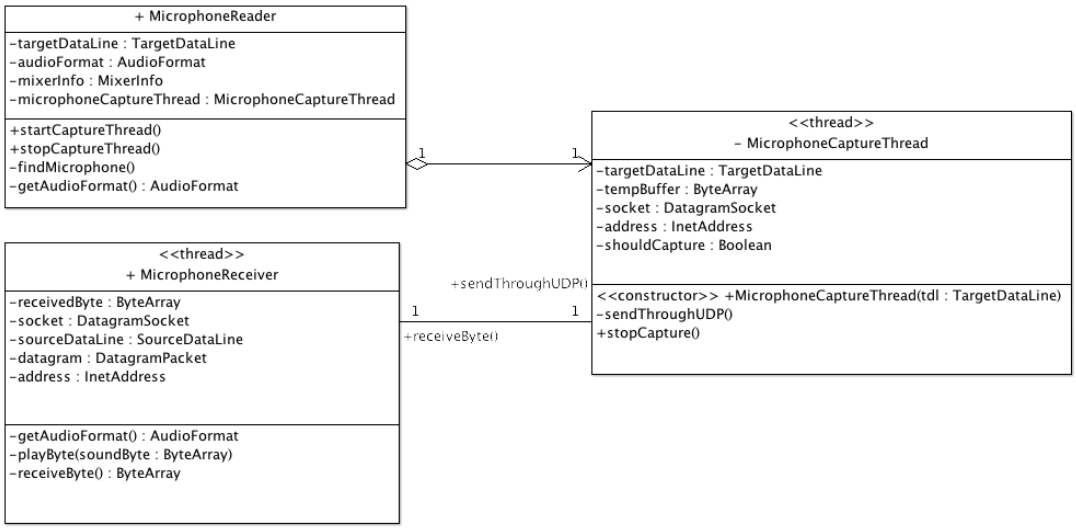


Figure 65: UML class diagram for the audio capture, transmission, reception and playback

Audio data is sampled at either the client or server end with a user-defined set of audio format properties using standard Java libraries. These audio format properties affect the quality, size and therefore transmission speed of audio data across the network and the balancing of the data is essential to the rapid transfer of data. These audio format features include:

- Sample rate
- Sample size in bits
- Number of channels

The signal is captured and transmitted as a pulse code modulated signal. On the receiving end, the software needs to know the audio format properties to correctly capture and replay the transmitted audio data. This poses a problem for dynamically readjusting the audio format during operation. This would require a separate transmission with this information before beginning the stream again. This is also a downside of using the UDP protocol, as it requires a fixed packet size that isn't possible using different audio format properties as they change the packet size.

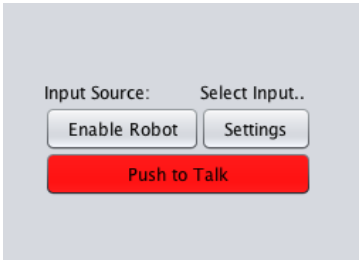


Figure 66: The two-way communication user interface (microphone active state)

The user interface for the two-way communications is very simple as seen in Figure 66. The push button highlights red when the microphone is active and is grey when inactive, providing the operator with instant visual feedback as to the system status.

2.4.5.4.4 Testing audio transmission

Initial testing was conducted using local-loopback, that is to say, the audio was transmitted and received using the same computer. Although this did not allow the transmission speed and losses to be accurately analysed, it did enable the testing of the functionality of the software. Delay was tested using the sharp sound of a clap, timed with a stopwatch before the computer played the sound back. The delay was originally around 4 seconds but with some adjustment of the sample size, rate and buffer size that was being sent it was possible to reduce this to two seconds. Time delay is not really important as the competition rules do not dictate the ability to hold a conversation but rather demonstration of the ability to both hear and speak through the robot. There is much further work that may be undertaken in this element of the client software, this is detailed in the recommendations for further work section.

2.4.5.5 A temporary audio amplifier

A small speaker was purchased to sit in the head of the robot but it was clear after initially plugging it in to the robot's computer, the speaker required an amplifier between it and the computer to ensure the replayed signal will be loud enough. As this was discovered rather late in the development process, it was too late to order special audio amplifier circuitry and so a temporary replacement was attempted.

2.4.5.5.1 Implementation

A simple 741 operational amplifier was used as a quick solution to the problem as there were already some in the WMR store cupboard and there was no time to experiment with different audio amplification circuitry. The 741 has characteristics that are poor for high-fidelity audio applications but was assumed to be sufficient for the reproduction of voice data.

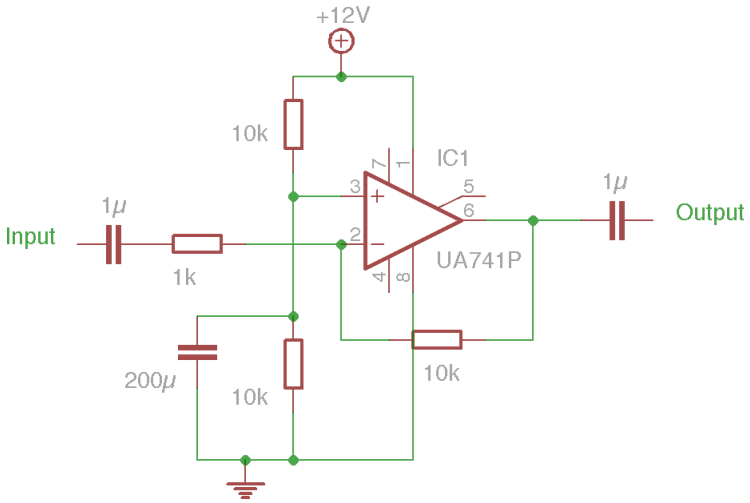


Figure 67: The quick audio amplifier design using a standard 741 operational amplifier

As can be seen in Figure 67, the amplifier is configured in a negative feedback arrangement, with the 10k and 1k resistors determining the gain, equal to 10k/1k or a gain, A_v, of 10. The capacitors ensure any DC is blocked either in or out of the circuit. To run the 741 from a single rail supply (+12V) it was necessary to provide an offset to the non-inverting terminal of the op-amp through means of a potential divider. By using same-valued resistors it is possible to offset the amplifier by half of the supply rail, at 6 volts. The DC is grounded through the 200μF capacitor.

2.4.5.5.2 Testing

The circuit was first prototyped on a breadboard and tested using the sound output from a computer. The device supplied an adequate amplification to a sample piece of music and so the circuit was soldered to strip board. The device was then tested again and amplification was again satisfactory. Further simple observations were made regarding the quality of the amplifier. It was clear at the competition that the level of amplification was simply not enough given the background noise. It was also obvious that there was a great deal of distortion when playing back recorded vocal clips which may be due to the output swing reaching the supply rail voltage. For this reason, the amplifier was deemed unsuitable for the competition.

2.5 Future System Improvements

2.5.1 Mechanical System Improvements

2.5.1.1 New, Bespoke Manipulator

More manufacturing time and effort should be invested into building a bespoke electromechanical manipulator such as that described in section 2.2.3.3.3. This design incorporates many simple machining processes and cheap components (such as stock length spacers) and also allows for overall standardisation through holding common spares for multiple sections of the robot. An additional improvement suggestion would be pressure feedback allowing operator to read and control gripper pressure.

2.5.1.2 Arm and Head Changes

If possible, the arm base reinforcement, which is currently quite heavy, should be further optimised to reduce its weight while maintaining the majority of its strength. Also, a new attachment method could be devised for quickly attaching and removing the arm and its connections.

The use of anti-backlash gears for the worm gears in the arm joints would help further reduce free vertical movement of the end of the arm and also prevent some of the shake seen during PID loop testing.

The router is too vulnerable to impact in the head, as was discovered during testing at the competition. It should be moved to a more secure, protected location, either by internalising it within the body and extending the aerials along the arm, or by moving it to, for example the arm base joint (Figure 68).

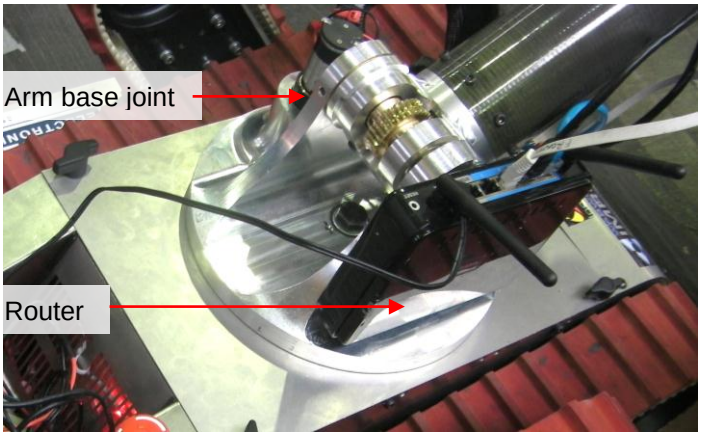


Figure 68: Positioning Router on Arm Base Joint

2.5.1.3 Flipper Drive Improvements

To further remove slack, a gear system could be implemented for the flipper motor clamps, using anti-backlash gears. The current system uses a chain and sprocket to transmit the power from the motor to the flipper. This is a poor transmission system as it takes up too much space and could deform with the new flipper motor clamp. A gear system would totally remove all slack from the flippers, increasing the reliability of the robot.

2.5.2 Electronic System Improvements

2.5.2.1 Arm Electronics

As identified in section 2.2.1.6.5, real world testing of the arm showed that more powerful motors and motor control boards would be required. Additional testing on the arm will be required to determine the specification for the replacement motors.

The current computer used by the USAR-T robot has not been upgraded since the first iteration of the robot, and therefore now four years old. Revising this hardware could increase the versatility of the robot.

2.5.2.2 Improved Computer Hardware

Optionally, a similar structured but higher performance computer could be chosen for the robot, resulting in no general change to the size of the system but greater ability to execute operations. This is likely to be very useful for any additional video transcoding that needs to be performed by the robot, currently used by the thermal imager.

Alternatively, an approach can be taken to integrate lesser power/mobile chipset based computers into the robot. Such a computer would most certainly aid in battery life but also occupy a smaller space when compared to the current computer. Intel Atom or VIA based chipsets, like those used in netbooks, would provide this functionality, and with at least comparable performance to the current computer, but it may be useful to look at boards built using the ARM chipset, such as a BeagleBoard (BeagleBoard.org 2011), which typically have a very small volume but feature all necessary components and ports. Besides performance requirements, the computer must also be capable of running Ubuntu Linux.

2.5.2.3 Battery Monitor Improvements

A few steps may be taken to improve the battery monitor's function: add current and charge measurement, monitor the voltage state of individual cells and integration with the existing power board.

By monitoring the current drawn from the battery it should be possible to measure the decrease in charge of the battery over time. Knowing the remaining charge in the battery and current it would be possible to estimate the time until depletion. This is not currently possible using the voltage measurement method of monitoring as the voltage does not decrease linearly.

The current system only measures the voltage across the two batteries. As each battery is composed of six cells there is a possibility that a cell may become critically low sooner than other cells if the battery is not balanced correctly. If this occurs it would not be detectable by the monitor, though a high enough voltage limit should largely mitigate this risk, and the battery would possibly be rendered unusable.

Integration with the power board would likely bring many advantages and with proper foresight not require too large a change to the board's design but of course re-fabrication would definitely be required. After integration, the monitor would be capable of e-stopping or disabling specific parts of the robot if required. Furthermore, integration with the power board would likely reduce the number of USB connections to the computer.

2.5.2.4 Motor Controller Board Upgrades

Besides the computer, the Roboteq motor controller boards could also benefit from an upgrade. The next generation of similar Roboteq boards would be a logical choice for this as they provide similar functionality but also provide advantages in communications protocol, USB rather than serial. Due to the limited number of serial ports available on the robot and the greater volume of occupancy taken by serial connectors it would be very useful to integrate USB-based boards in their stead. The newer boards also occupy lesser volumes of

space but can also provide greater powers to the motors without fault. Table 4 shows the comparison between two comparable Roboteq motor controllers, the next generation SDC2130 and the AX500 currently used by the robot (Roboteq 2011).

	SDC2130	AX500
Voltage Range	7V to 30V or 50V	8 to 24V
Number of Motor Channels	2	2
Max Current	2x20A	2x15A
USB Interface	Yes	No
Dimensions	70mm x 70mm x 19mm	106mm x 50mm x 38 mm

Table 4: Motor controller specifications

2.5.3 Software System Improvements

2.5.3.1 Improved Joint Class for Arm Control

There are several possibilities for expanding the arm design outside of physical alterations. Improvements to the Joint class could be made to provide a more dynamic design. For example, each joint could provide references to the proceeding or succeeding joints, in effect a doubly linked list. In addition the class could include details on the related physical joint's degrees of freedom and following member length. With these changes it would be feasible to develop methods that can dynamically find the forward and inverse kinematics such that additional degrees of freedom (joint or actuators) can be added to the robot without having to develop new solutions manually.

2.5.3.2 Improved Manipulator Control

As stated in 2.4.4, the current manipulator system only uses binary controls, grip and un-grip. Expanding the system to include variable positions for the gripper should be relatively simple to implement. Holding a button would increase a count, and possibly causing the grip to reduce the distance between the gripper's fingers, while holding a different button would decrease the count, and possibly causing the grip to increase the distance between the gripper's fingers.

2.5.3.3 Possible Integration with ROS

The Robot Operating System (ROS) (ROS (Stanford, Willow Garage) 2011) provides tools and libraries for developing robots very similar to those developed by WMR and for which the available support could greatly aid long-term development. Integration with ROS would not require any new hardware. In addition, it was designed for use with our current operating system, Ubuntu Linux, and can be easily installed through its package manager.

Drivers and libraries already exist for the vast majority of our sensors and controller boards such as the Xsens, Roboteq AX series, LiDar, Dynamixel and Kinect and others can be written with ease. Other libraries also exist to aid with autonomous actions, such as SLAM, or to aid in the kinematic control of the robot's arm.

Furthermore, the developer community for the project is also very large and new drivers or other packages are released at regular intervals by contributors while support is easily found through tutorials, wikis or forums.

However, Integrating ROS does introduce several major caveats. The current Java code used by the robot would be almost entirely deprecated as ROS requires use of either C++ or Python as programming languages. The current code base and documentation should provide an adequate guide for the implementation of the robot's current features but additional time would be required developing and testing the new system.

2.5.3.4 Future Changes to Robot Client Software

The temptation for each year to start the client software from scratch is inescapable as it provides a new team a fresh start on the problem. This should be unnecessary thanks to the extendible fashion in which the client has been rebuilt. Work should primarily be made in ensuring that the IR camera stream integrates correctly with the user interface. A simple averaging algorithm should be constructed to estimate the heat over the area viewed by the camera as well to provide the operator with simple feedback via the signs of life panel.

Neural networks also should be investigated for processing the sensor data collected by the robot. Given the array of information being collected (CO_2, heat, sound, visual) it should be possible to feed this information into a system that could estimate the probability of a victim being located within the vicinity of the robot. This "assisted control" is of far more use in the tele-operation of the robot than autonomous methods presented elsewhere. The best method of control may be whereby the operator of the robot is presented with these probabilities from which they may decide to investigate further to find a victim.

Incorporation of mapping from any source would also be useful, not only in collecting points but for navigating the arena. Although there is visual feedback from the webcams, it is easy

to become disorientated controlling the robot and a map generated of the arena would provide both points and a useful method of planning a route around the arena.

Two-way communications need to be implemented fully and tested both on the client and the robot over different wireless standards (Both 802.11a and 802.11n) with varying levels of background noise to determine the most acceptable level of compression and delay for the given application.

An improved audio amplifier using a specialist audio amplification chip and more rigorous analysis of the amplifier requirements should not prove too much of a challenge and can provide multiple points for every victim discovered.

A method of controlling the robot arm using the control pad using the inverse kinematic system would also help greatly at the competition, enabling a much smoother method of approaching and manipulating the different objects in the arena.

3 USAR-A, Autonomous Robot

The Autonomous Urban Search and Rescue (USAR-A) robot is the autonomy testing platform for the USAR robots. It is required to navigate much less complex terrain and does not have to complete tasks such as lifting and delivering payloads. As a result the platform can be much simpler in design, not requiring the flippers or arm used in the USAR-T. This year's USAR-A has been redesigned to have a core body similar to that of the USAR-T as part of the next stage of merging autonomy into the teleoperated robot.

Students in the Computer Science Department completed the programming and control of this year's USAR-A so the main engineering task this year was developing the platform itself.

3.1 Evaluation of the 2009/10 Robot

The 2009/10 USAR-A platform was the first stage development platform for autonomous robot navigation and victim finding. The platform performed effectively on flat ground but had difficulty with more sloped terrain. A SWOT analysis was carried out on the robot to identify the areas that most needed improvement this year:

3.1.1 SWOT analysis

3.1.1.1 Strengths

The main chassis' simple design was quick to manufacture and easy to assemble. The incorporated adjustable track tensioning features were also useful. A lot of parts from the old teleoperated robot had been recycled into this robot which helped keep the component costs down and the motors provided easily sufficient power to move the robot. The two degree of freedom head was positioned at a good height and had a functioning array of sufficiently accurate sensors used by the robot autonomy.

3.1.1.2 Weaknesses

The chassis had several major weaknesses that made the computer control of the robot more difficult. The centre of mass was located in the rear which resulted in off-centre turning and sudden falls over the edge of some slopes. A low ground clearance meant that the robot was vulnerable to getting stuck on ridges between opposing slopes. The chassis was also quite

flimsy compared to that of the USAR-T, requiring additional internal support from a torsion bar.

The sensors, although at an appropriate height for some victims, were too low for detecting others. The sensors in the head were also poorly protected from impact and also prone to substantial vibration during the operation of the robot. The internal board layout of the autonomous robot was also quite poor.

Although the motor power was sufficient in the robot's drive system, the belts sometimes were unable to provide sufficient grip on smooth surfaces. The belts also would occasionally loosen, adding to the problem.

3.1.1.3 Opportunities

The extensive manufacturing capabilities of WMG will mean that manufacture of a new chassis should be relatively quick. Rebuilding the USAR-A chassis will also provide opportunity to make it more similar to the successful USAR-T chassis, continuing in the progression towards merging the two platforms. The same merging of designs can be said for the drive train, using a design standardised with the USAR-T.

New technological developments have meant that 3D depth perception sensors are available at significantly lower prices through hardware such as the Xbox Kinect. This new avenue of technology may be investigated further by the computer Science Department.

3.1.1.4 Threats

Potential changes to the operating environment such as sand or other small debris may cause problems with the autonomous robot's tracks, particularly as the autonomous robot may not be aware that something is going wrong.

In standardising the chassis, there will be a large amount of parts needing to be manufactured and bought in (such as the customised tracks) which may be subject to delays.

3.1.2 Identified Areas for Improvement

Having completed the SWOT analysis on the 2009/10 USAR-A, the following areas were identified for improvement this year:

- Redevelopment of chassis and drive train to be more similar to that of the USAR-T.

- Design a new head to better protect the robot's sensor array and incorporate any new sensors such as the Xbox Kinect.
- Redesign the internal electronics stack

3.2 Mechanical System Improvements

3.2.1 Chassis Sub-System

The chassis functions as the structural basis for the entire robot and all parts are ultimately secured against it. The chassis houses the electronic components and is a contributing factor to the robots mobility.

3.2.1.1 Evaluation of the 2009/2010 Chassis

The 2009/2010 autonomous robot chassis design is a simple (490mm Length x 280mm Width x 150mm Height) box made from 0.9mm stainless steel. The chassis is separated into two compartments, a motor housing (180L x 280W x 150H) and an electronic components housing (310L x 280W x 150H). The plates are manufactured from of 0.9mm laser cut stainless steel panels which are folded into shape. In order to improve torsion stiffness the chassis also has a 6mm diameter bar across its width during operation as shown in Figure 71 Torsion Bar View 1 and Figure 72 Torsion Bar view 2.

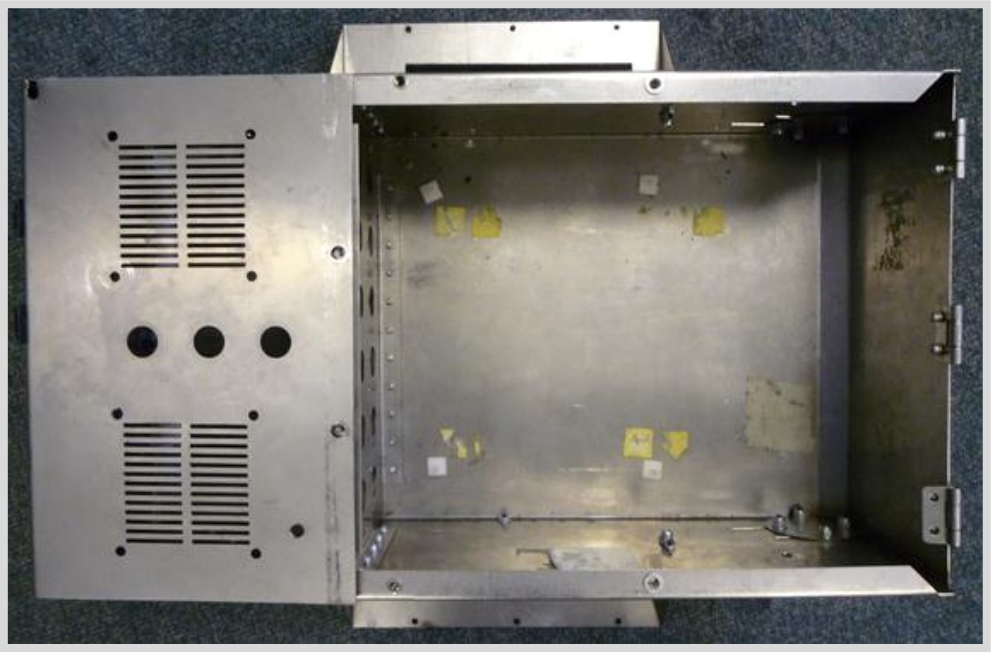


Figure 69(2009-2010) Autonomous Chassis Top View

Figure 70(2009-2010) Autonomous Chassis

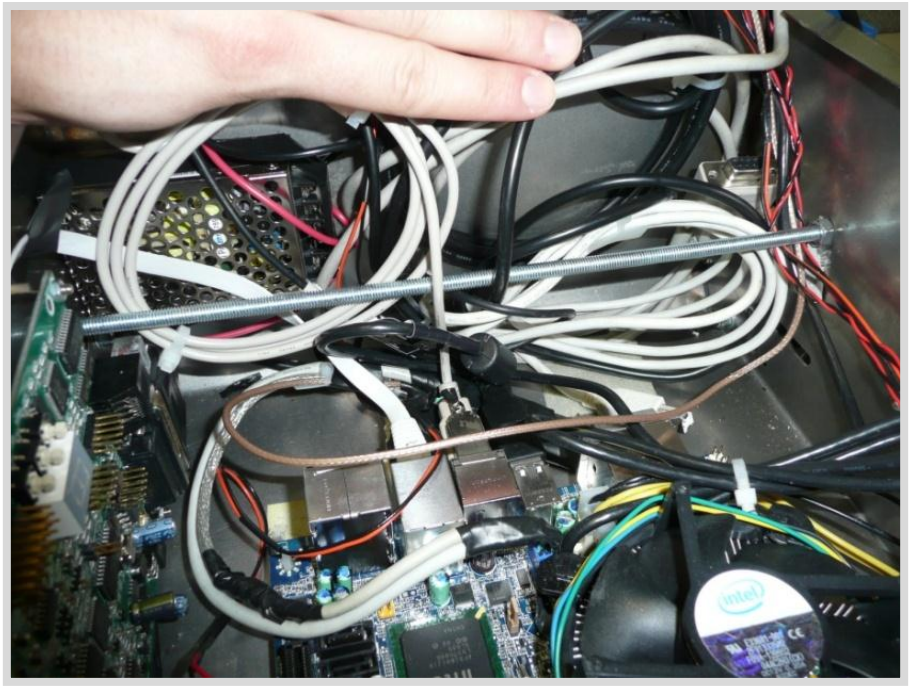


Figure 71 Torsion Bar View 1

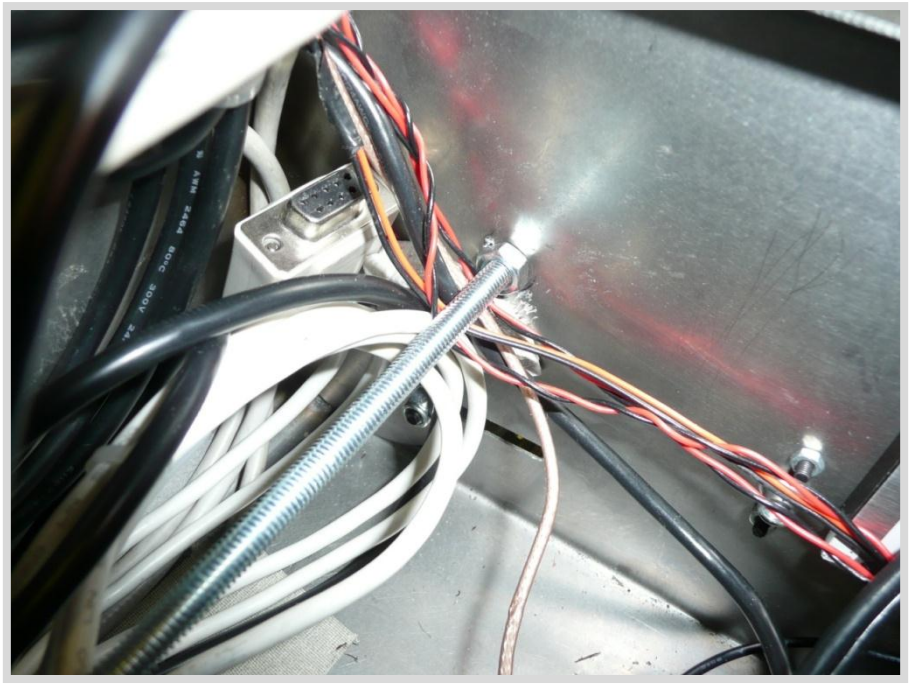


Figure 72 Torsion Bar view 2

3.2.1.2 Specification

From the analysis of the current chassis it is possible to outline specifications for the autonomous chassis design:

- The Autonomous chassis should be as similar to the tele-operated robot as possible, in order to bring WMR products closer to uniformity.
- The chassis must be capable of housing all the necessary electronic components, A volume of at least 0.013m², as well as protect them from the impact should the robot potential fall from a distance of 200mm.
- The chassis and drive chain combined must also give a ground clearance of at least 20mm; double that of the 2009-2010 design.
- The cumulative height of the head, chassis and drive train must place the sensors in the head at least 500mm above the ground. Therefore the chassis should be at least 180mm in height
- The chassis must also include battery holders with dimensions at least L: 190mm W: 60mm H: 80mm. The battery holders should not be significantly larger than this to avoid excess battery movement.

An additional design suggestion is the addition of handles to the chassis. This will not aid the performance of USAR-A, however it will allow the WMR team to lift it much easier, due to the weight and limited places which to grip USAR-A it is very difficult to lift. It would

therefore be beneficial to design and manufacture handles that could be used on the robot either permanently or temporarily.

3.2.1.3 Development and Justification of Designs

It is possible to address all the weaknesses previously identified in the evaluation with a chassis redesign. This makes up a large proportion of all the weaknesses identified with the entire autonomous robot. These are weaknesses which must be addressed in order to achieve WMR's 2010-2011 aims and objectives of improving the reliability of both platforms and securing the European championship.

The autonomous robot chassis is based on the teleoperated chassis design for many reasons. Firstly the teleoperated robot is a proven platform, previously winning the best in mobility award in the RoboCup Rescue competition. This chassis design would vastly improve on many of the weaknesses identified with the current autonomous chassis as it offers improved ground clearance and a much stiffer design. Another advantage offered by the teleoperated design is ease of assembly and disassembly.

Another advantage with basing the chassis on the teleoperated design is increased interchangeability between the two robot platforms. This also brings WMR a step closer to its aim of producing a standardised product that can be operated both by a user and autonomously. The design will allow for parts to be moved from one platform to another in the event of a component failure. Having a single platform would also make design and maintenance simpler and would be a more effective use of resources.

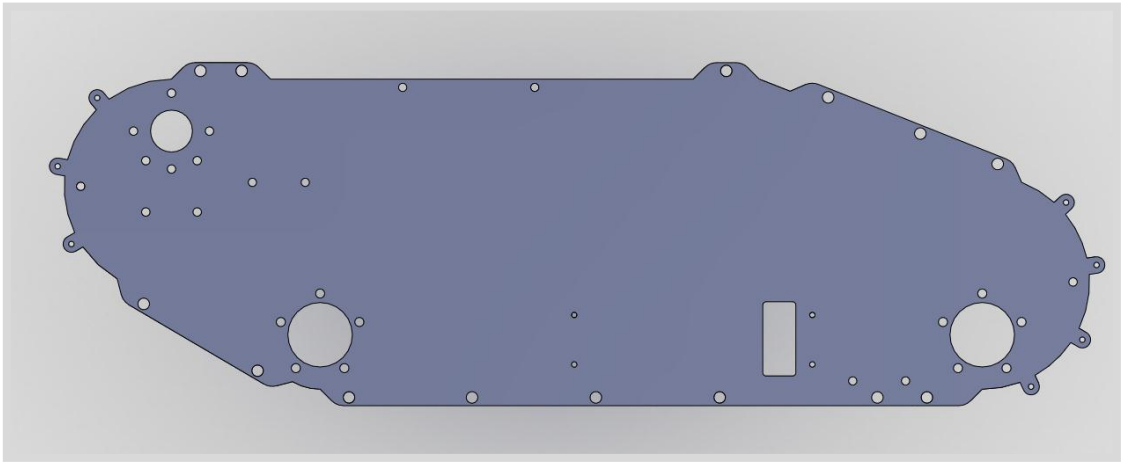


Figure 73: 2009/2010 Teleoperated Side Plate Design

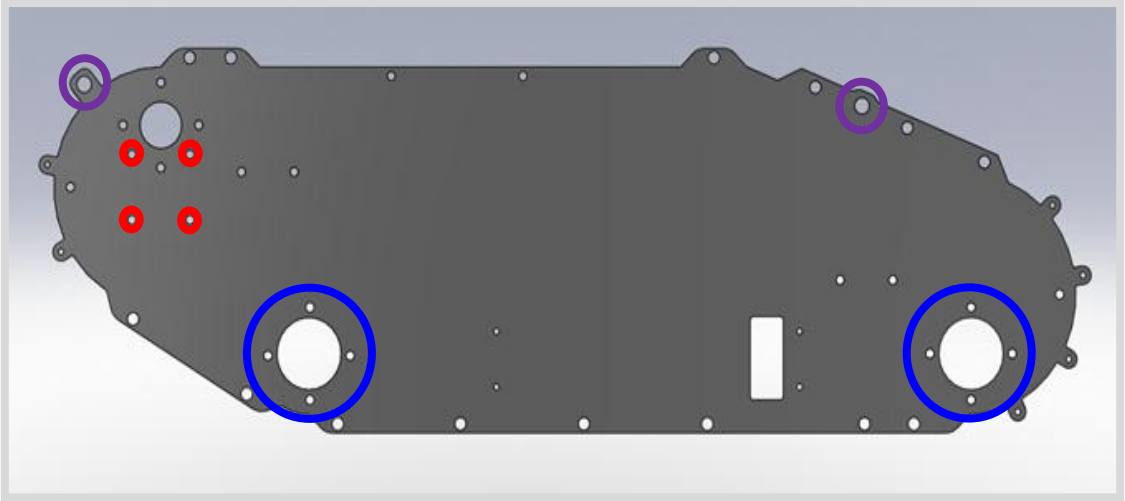


Figure 74: 2010/2011 Autonomous Side Plate Design

The overall design shape and geometry is identical however certain modifications had to be made as shown to the side plates. The current teleoperated chassis design for the side plate is shown in Figure 73: 2009/2010 Teleoperated Side Plate Design, the changes which have been made are highlighted in Figure 74: 2010/2011 Autonomous Side Plate Design. These changes Black Circles – Larger holes (9mm) were added at the highlighted points in order to house M7were made due to the following factors:

- Purple circles - threaded rivet nuts which will secure the handles. These threaded rivet nuts have to be large in order to support the weight of the robot when lifted. The addition of handles gives no performance improvement but reduces the risk of accidental impact from mishandling.
- Red Circles – These holes are spaced at a 22mm radius as opposed to a 25mm radius in order to attach to the motors which in the autonomous robot are slightly smaller in diameter.
- Blue Circles – The number of outer holes which secure the stub shafts to the chassis has been reduce from 5 to four in order for ease of assembly and disassembly and to bring the stub shafts into uniformity with each other.

The chassis material was chosen to be 0.9mm stainless steel sheeting, as with the previous autonomous design and the current teleoperated design. Stainless steel is affordable and strong enough for the requirements, as demonstrated by the USAR-T chassis.

The manufacturing method to be used is laser cutting and bending this is the method used for the previous autonomous and teleoperated chassis. It is well within the capabilities of the WMG and is appropriate for 0.9mm stainless steel.

Threaded rivet nuts will be placed in all the holes situated on the outer tabs as seen in Figure 74: 2010/2011 Autonomous Side Plate Design, these are cheap and make assembly and disassembly much easier than a standard nut and bolt design.

3.2.1.4 Final Chassis Design

Images of the CAD design of the final chassis and its assembly can be seen in Figure 75 and Figure 76 below:

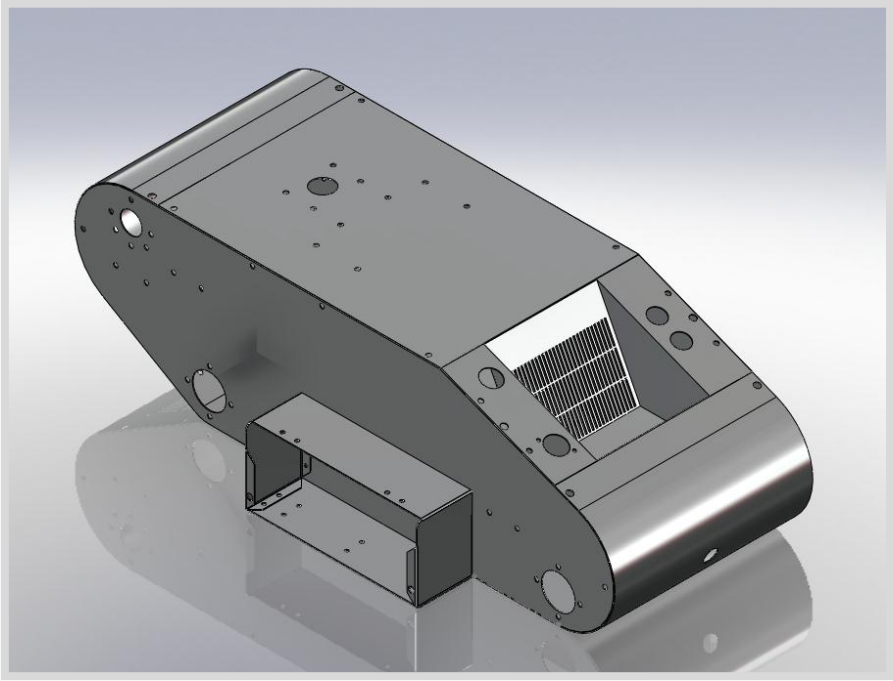


Figure 75: Image of Assembled Chassis

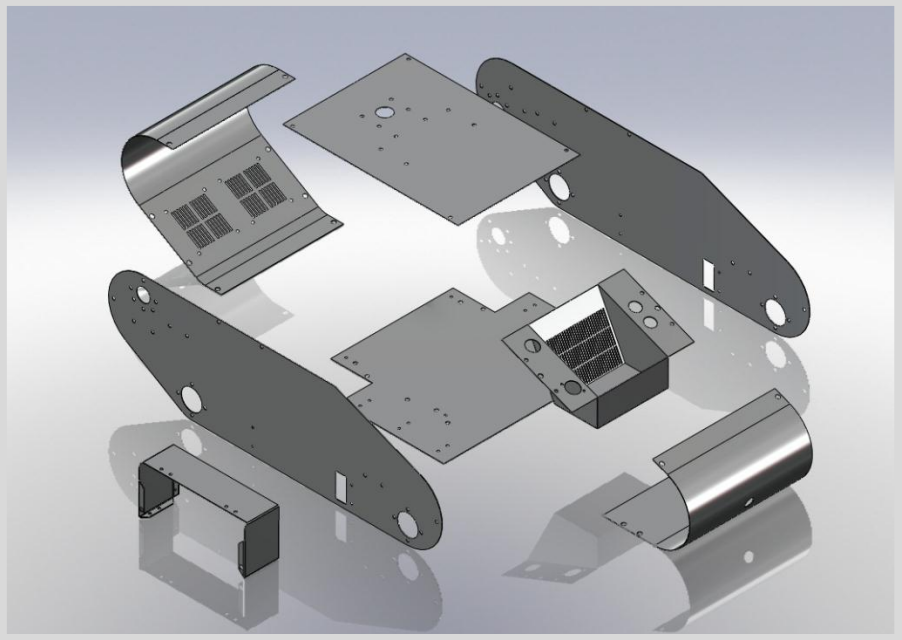


Figure 76: Exploded View of Chassis Assembly

Figure 77 below shows the fully assembled USAR-A with its new chassis design:

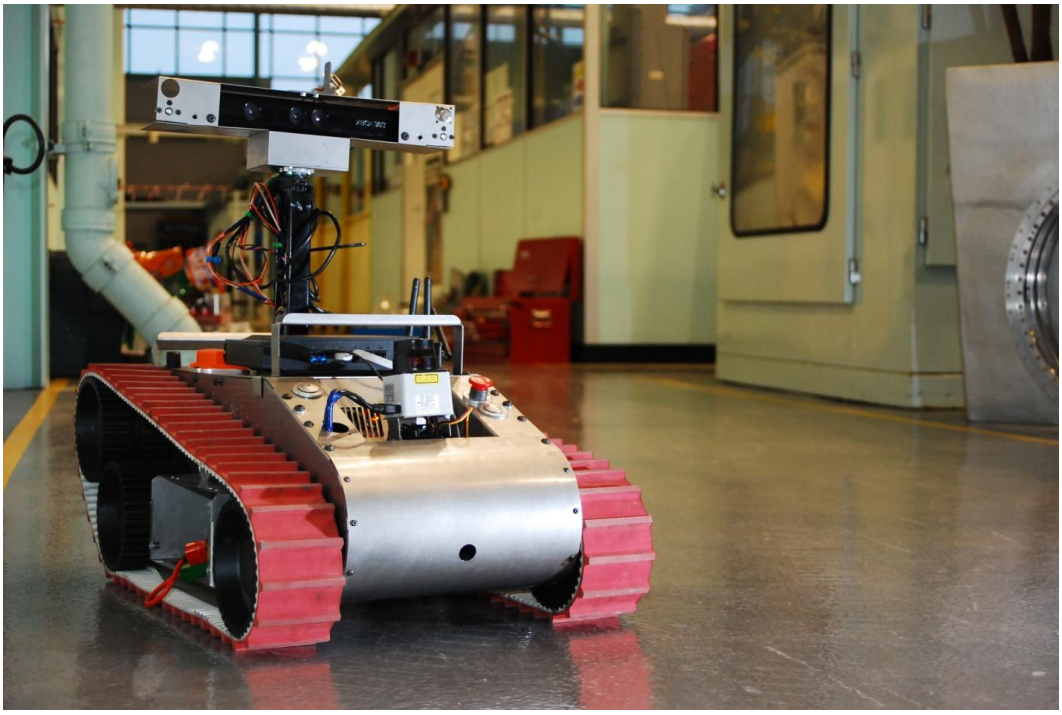


Figure 77: Final, assembled USAR-A Chassis

3.2.1.5 Evaluation of new Chassis

3.2.1.5.1 Evaluation against specification

All of the specifications outlined were achieved. A comparison against the specification was as follows:

- The overall shape of the autonomous chassis is identical to that of the teleoperated. There are subtle alterations that are highlighted and justified in Development and Justification of Designs.
- The chassis volume allocated to the stack is 0.01746cm^3 well above the minimum volume of 0.013m^3
- The manufactured Battery holders are L:196mm W:63.5mm H:82.6mm. These dimensions are within the specification sizes.
- The ground clearance is 40mm four times that of previous years design and twice that of the minimum specification
- The chassis height is 188mm high, exceeding the specification minimum chassis height.

3.2.1.5.2 Performance at RoboCup competition

USAR-A operation in the RoboCup Rescue competition allowed WMR to analyse the performance of the new chassis.

Initially USAR-A was operated autonomously as intended. Whilst autonomously operated, USAR-A was only required to navigate yellow grade terrain, this includes. 15° degree slopes. Due to problems with mapping and motor control USAR-A did not negotiate the yellow terrain.

Due to an electronic failure with USAR-T, USAR-A was switched to teleoperated control for the later rounds of the competition, as this was deemed to be more effective. This then required USAR-A to navigate much tougher Orange and Red grade terrain. This was well beyond the initial considerations made in USAR-A's chassis design obstacles included, alternating 15° slopes as shown in Figure 78 - Alternating Slopes, a 45° slope, a 200 mm step, a 45° flight of stairs and a random maze of 500mm to 100mm steps, known as step fields (Jacoff 2009). A robots ability to physically negotiate these terrains is dependent on a combination of the chassis and drivetrain.

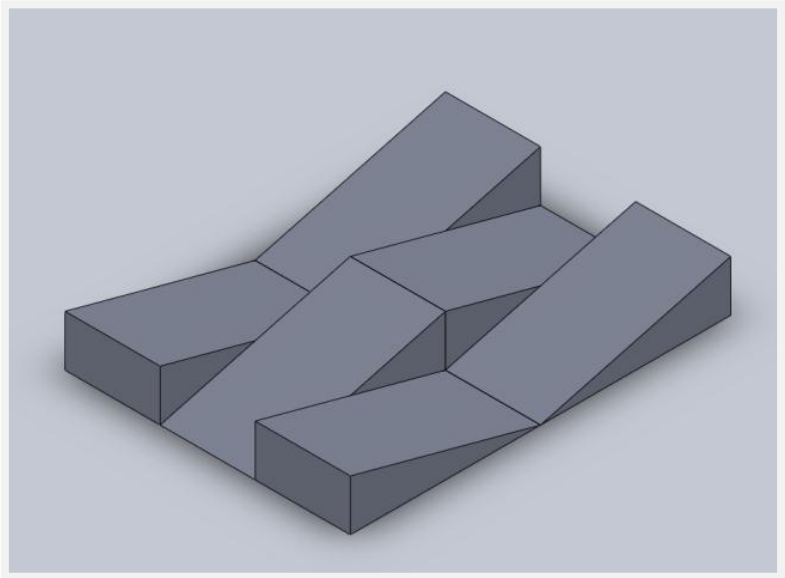


Figure 78 - Alternating Slopes

The USAR-A robot successfully; Ascended and descended the 45-degree slope, dropped down the 200 mm steps and negotiated the alternating 15-degree slopes. However USAR-A failed in an attempt to ascend and descend the 45°-degree flight of stairs. The step fields were not attempted.

Although USAR-A failed in some of the more difficult terrain challenges, the overall mobility of USAR-A platform, as demonstrated in the robocup competition, can perform above and beyond its normal operating requirements.

3.2.2 Drive Train Sub-System

The drive train for the autonomous robot transfers the power from the motors inside the robot to the tracks, where the tracks will pull the robot along the ground. The autonomous robot drive train has been redesigned and manufactured in order to help achieve specific aims which were outlined in the beginning of the project and to address some of the weaknesses which were highlighted in the SWOT analysis.

3.2.2.1 Evaluation of 2009/10 Drive Train

The drive train for the autonomous robot can been seen below in Figure 79 This shows the pulleys, manufactured from aluminium, meaning the pulleys were heavy. As the chassis is being redesigned to have more uniformity with the teleoperated robot, the autonomous drive train needs to be modified to fit this uniformity. The drive chain from the teleoperated robot is more than adequate for the requirements. It would give the autonomous chassis enough power from the motors and grip to manage the terrain needed.

Figure 79: Old Autonomous Drive Chain

3.2.2.2 Specification

The specification for this modification is that the motors need to be transferred from the old autonomous chassis to the new autonomous chassis.

Using the SWOT Analysis, it shows that the WMR team need to design a drive train with a larger amount of ground clearance, more suitable belts and more suitable materials for the pulleys.

3.2.2.3 Design Method

The design of the power train needed to take into consideration the "MagMotors" from the old autonomous robot. These motors were larger than the motors used in the USAR-T and also had imperial measurements. The drive train was designed in conjunction with the chassis so the parts would mesh with one another when manufactured. This is because the WMR team in the past has had problems with simultaneous engineering.

3.2.2.4 Final Design

The drive train uses several different components. These different components can be seen below in Figure 80. This figure shows a cut-through of one of the pulleys and the different components involved in the design.

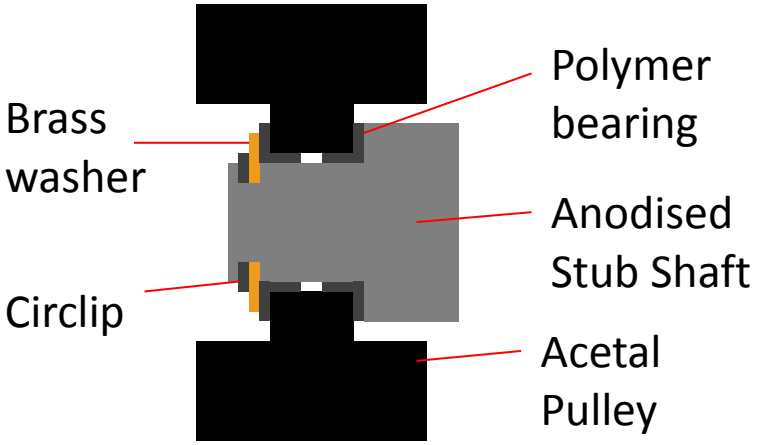


Figure 80: Cut-through of a pulley

The figure below shows the two different designs of the pulleys, the drive pulley and slave pulley respectively. The pulleys were manufactured from acetal, a lightweight polymer that has comparable mechanical properties. The drive pulley has an annulus gear attached to it which is where the motor spur gear will mesh with the drive pulley. The annulus gear used was from the previous autonomous robot.

Figure 81: Drive pulley (left) and Slave Pulley (right)

Figure 82 shows the stub shafts attached to the chassis. The stub shafts designed for drive train were anodised to give the surface of the aluminium used an increased corrosion and wear resistance.

Figure 82: Anodised Stub Shaft

The tracks used were the same tracks used on the USAR-T. The tracks purchased were 75TK10K13C/1410V. They were custom built to fit the drive train designed. With 75 teeth around the outside, this gave enough grip to the USAR-T so will be suitable for the USAR-A.

As the other USAR product uses polymer bearings to give the drive train a low friction surface for the shaft to revolve within, it is obvious why they were purchased again for the new drive train.

The final design can be seen below in Figure 83, showing the drive train constructed in its entirety.

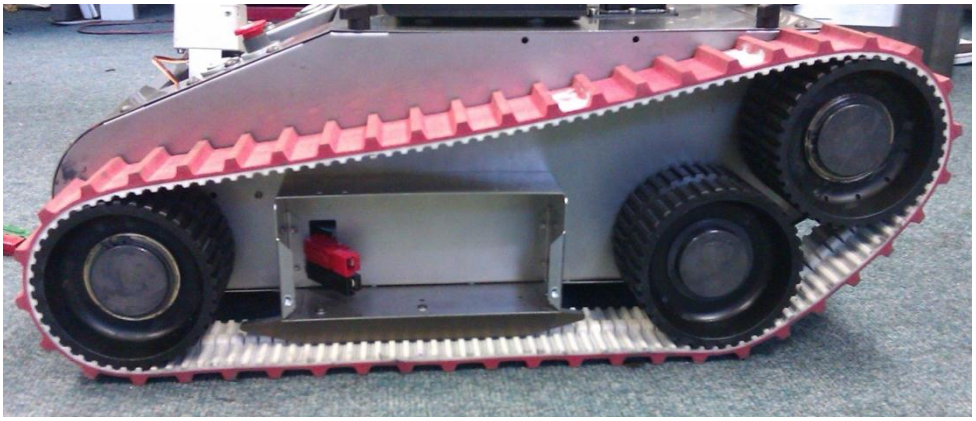


Figure 83: New USAR-A power train

3.2.2.4.1 Evaluation and Testing

The performance of the tracks and the drive train during the competition demonstrates the drastic improvement in mobility of the robot as a direct result from these modifications as the USAR-A tackled some of the most difficult terrain. This can be seen in Figure 84 where the USAR-A robot travels up a 45° slope, with little difficulty. This shows that the motors deliver enough torque to the track to keep the robot stationary on an inclined plane.

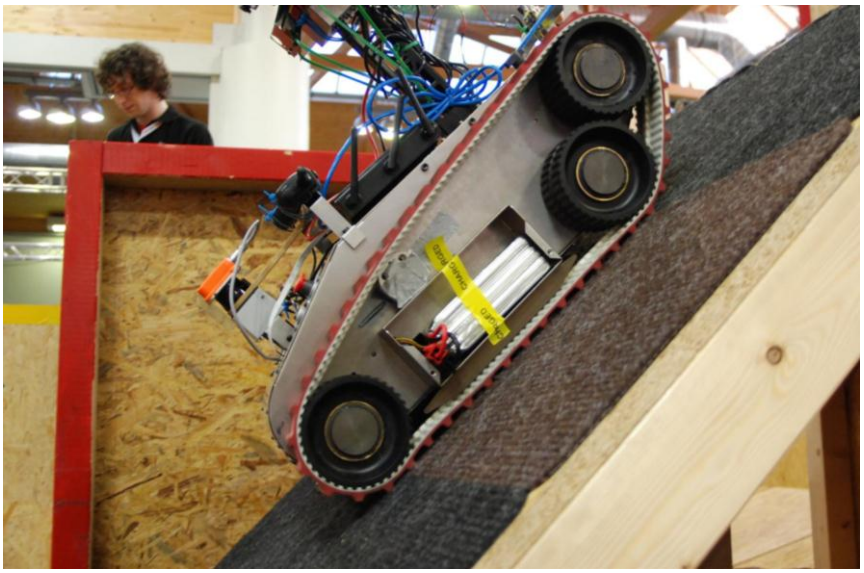


Figure 84: USAR-A traversing difficult terrain

The new custom made tracks meant that the USAR-A had excellent grip on most surfaces and could tackle all obstacles in the Orange Zone of the competition and a two of the obstacles in the Red Zone.

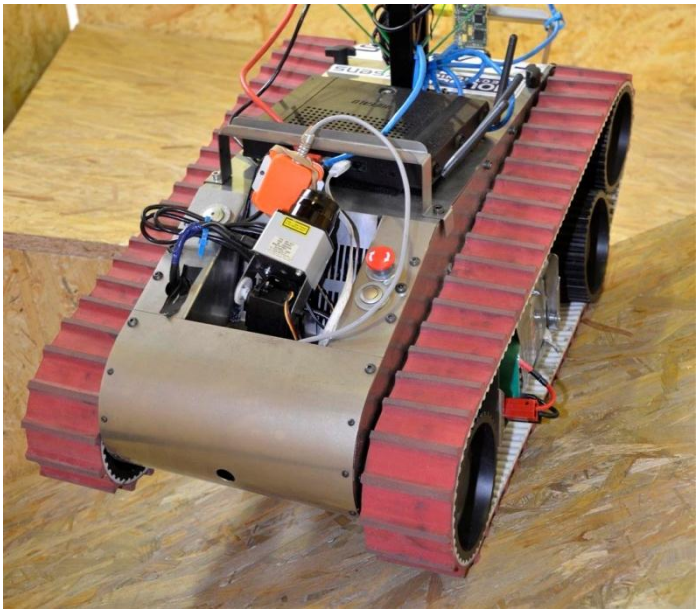


Figure 85: USAR-A tackling inverted ramps

However, due to some damage to the left hand motor of the drive train, there was significant vibration to the whole of the chassis. This was damped at the competition using redundant parts from the previous chassis.

3.2.3 Head Sub-System

The head on the autonomous robot houses a variety of sensors; these are positioned away from the main body of the robot in order to gain a better "view" of the surrounding area. To a limited extent this position also reduces electrical noise in the sensors generated from the electric motors. The head also functions as a method of altering the orientation of the sensors with respect to the robot chassis through panning left to right and up and down.

3.2.3.1 Evaluation of 2009/10 Head

As last year was the year the autonomous robot was created there has been one previous design of an autonomous head.

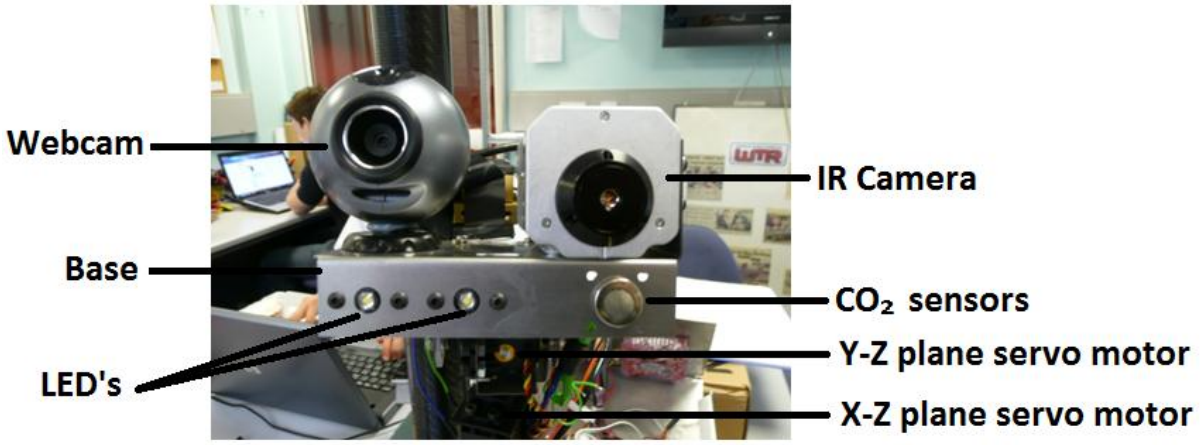


Figure 86 Autonomous Head Front

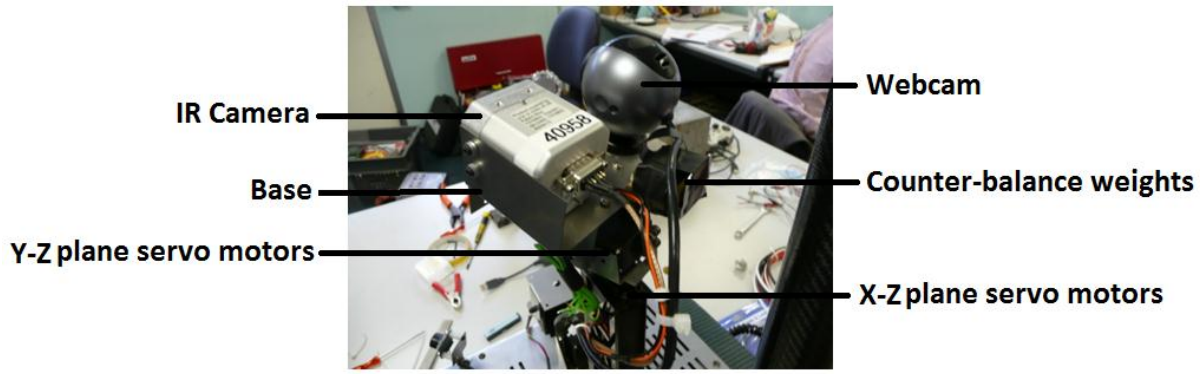


Figure 87 - Autonomous Head Rear

3.2.3.2 Specification

The head must house all the necessary sensors in an appropriate position with two degrees of freedom in order for the sensors to pan from left to right and up and down. The head should also protect the sensors, which are considerably valuable, from potential impact during operation and transportation.

The head must be capable of containing the following sensors

- Microsoft Kinect
- Light Emitting Diodes
- Infrared camera
- Microphones
- CO2 sensors
- Loud Speaker

The head should be positioned at least 500mm above the ground and also 500mm away from the front of the chassis, therefore the stand should be sufficiently high and placed at a

sufficient distance away from the front of the chassis. The 2 microphones should be placed as far apart as practical.

3.2.3.3 Final Design

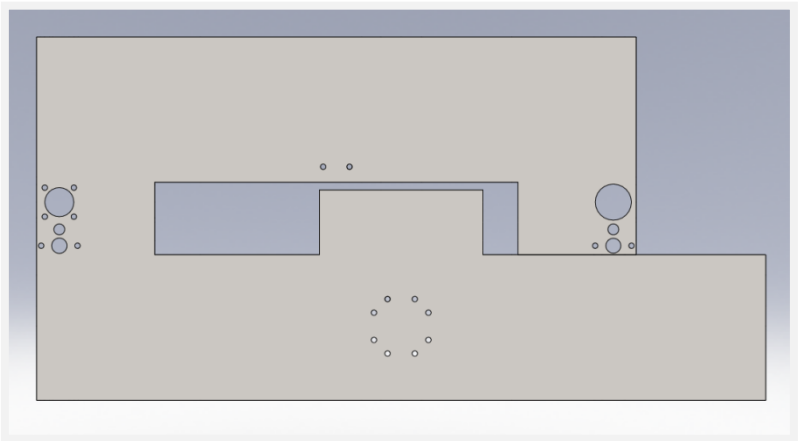


Figure 88 - USAR-A Unfolded Head Plate

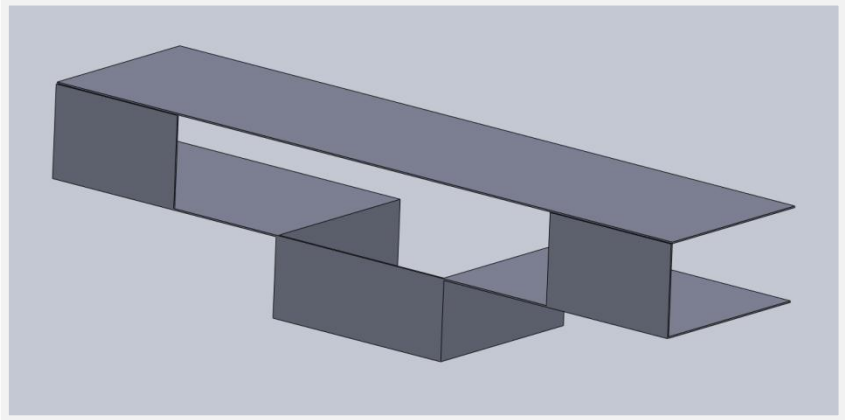


Figure 89 – USAR-A Head Folding Geometry

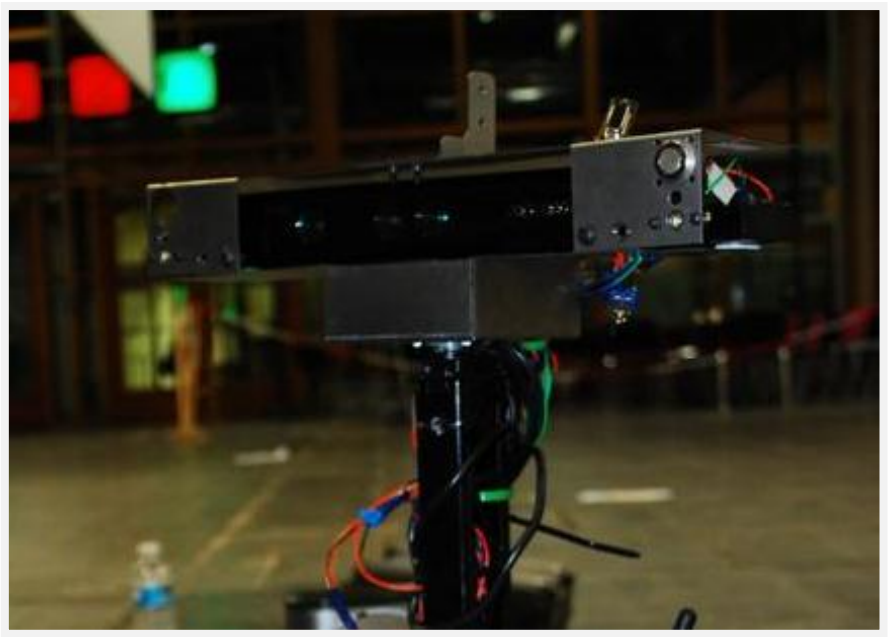


Figure 90 - Manufactured USAR-A Head

3.2.3.4 Justification of Design and purchases

The head should be positioned at least 500mm above the ground in order to identify victims placed at medium and low heights at the competition. The highest “victims” are at heights of 900mm however there are very few of these and having a head at this height means that the robot would not be able to get under certain obstacles. The only way to avoid this is to have a telescopic arm, this was deemed to be too much work for very limited return and therefore was not pursued.

It was necessary to redesign the head in order to house the Microsoft Kinect, which is a new sensor for 2010/2011 and a key sensor in the autonomous robots operation.

The Microsoft Kinect sensor must be placed 500mm away from the front of the chassis. This is because the Kinect cannot detect objects within 500mm of its lens. Placing the head at this distance gives the advantage that chassis will not interfere with the Kinect depth sensor which in turn means that the autonomous robot will not detect itself and mistake it for an obstacle. Placing the Kinect there also means that it does not have a “blind” spot in the 500mm directly in front of the chassis should it be placed on the front of USAR-A. This would be problematic during operation should USAR-A come in close proximity of an obstacle.

The design is made from 0.9mm thick stainless steel sheet which is laser cut and folded into shape. The structure of the head is subject to very little load and thus negligible stress. Stainless steel will be used due to its manufacturability, availability and low cost. It is

therefore unnecessary to do a finite element analysis as the steel will be far stronger than needed.

Protection is considered a secondary concern as the autonomous does not have to negotiate terrain which is likely to cause it to suffer large impacts. Therefore the design offers limited impact protection in order to keep the design simple and costs low.

3.2.3.5 Evaluation

Certain specifications were met in the autonomous head design. The height of the sensors are 500mm and the distance from the chassis front 420mm.

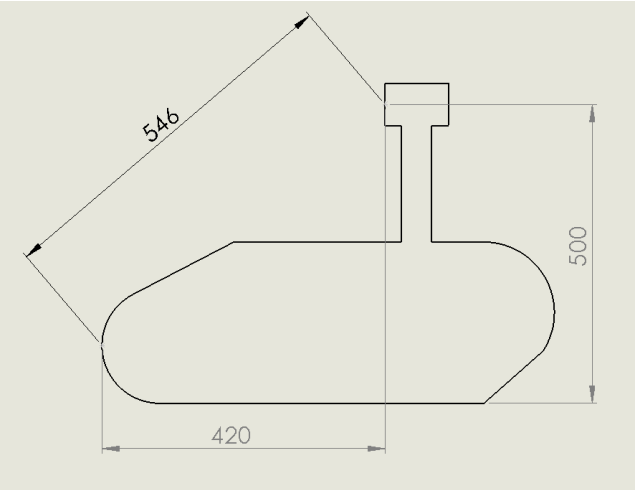


Figure 91 - USAR-A Head Position

In other aspects the head design fell short of the specification required of it. Although this was not detrimental to WMR performance at the RoboCup competition, in order to perform in future competitions this is an area that needs a complete redesign. The following shortcomings were identified.

Due to an oversight in the size of the LED's, the microphones did not fit in the space allocated for them in the head. This did not affect the operation of USAR-A in the competition as USAR-A currently lacks the software capabilities to use microphones. A loud speaker was also excluded from the head as again USAR-A lacked the software to use the function, it was seen as an inefficient use of time and resources to produce a loud speaker and amplifier. In the designs the IR camera was mounted on top of the head offering no protection.

When operational it was discovered that the head was too heavy for the Y-Z servo to lift. In order to keep the sensors at an operational angle the Y-Z servo was removed. This meant losing a degree of freedom in the head, but again USAR-A lacks the software to use the Y-Z

plane servo autonomously, so this should not have affect the performance of USAR-A at the competition. Removing this servo also meant that a thrust bearing purchased in order to limit vibrations also had to be removed this could not be incorporated into the design without the servo in question. This caused the head to vibrate significantly during operation.

At the RoboCup rescue competition the X-Z servo was broken in an accident. The part of the servo broken was unfixable and no replacements were available, which meant that this servo also had to be removed again losing a degree of freedom.

An unforeseen problem with having the Kinect horizontally mounted was that on slopes the head overhung the tracks (Figure 92) this made the head susceptible to hitting walls and edges. During autonomous operation at the competition this caused damage to the head. Although the sensors were kept intact by the structure, the joint between the head and the stem was damaged.

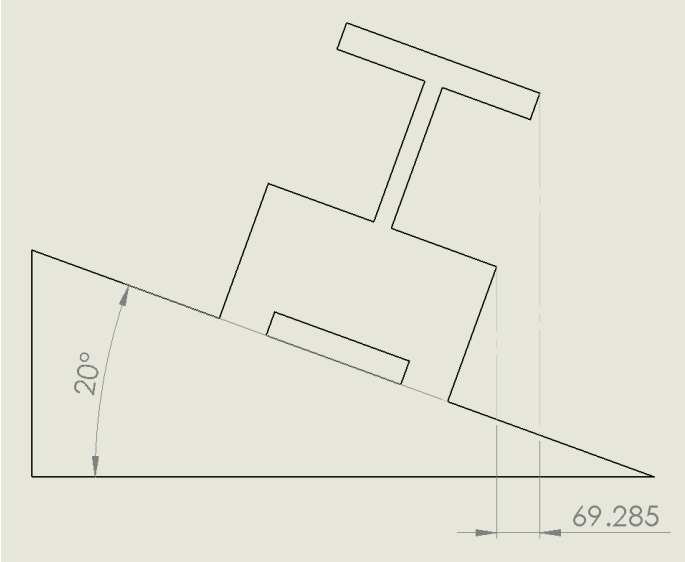


Figure 92: Head Vulnerability while Tipping

Due to an electronic failure in USAR-T, USAR-A was teleoperated for some rounds of the competition. As a result of losing the two degrees of freedom in the head some victims could not be identified.

3.3 Electronic System Improvements

3.3.1 New System Sensors

Several new sensors were added to the robot at the request of the computer science team. These included the Xsens MTi and Microsoft Kinect.

3.3.2 Internal Electronics System Redesign

3.3.2.1 Introduction

With the upgrading of the chassis of the USAR -A to match the design of the USAR-T the rearrangement of the electronic and electrical components were required. The newly implemented stack was hoped to bring similar advantages observed in the USAR-T to the new USAR-A; the accessibility of connections, the ease of modification of components on the stack itself and finally the vibration-reducing properties on the less robust electronic components.

3.3.2.2 Investigation into Connectors

Harwin connectors are specialiased data and power connectors purchased by previous year's teams (Harwin Plc 2011). The datamate cable-to-cable (or wire-to-wire) range are shown in Figure 93 which clearly demonstrates their utilitiy in keeping each connection electrically isolated from the other and yet compacted into a tightly organised layout of secure connections.

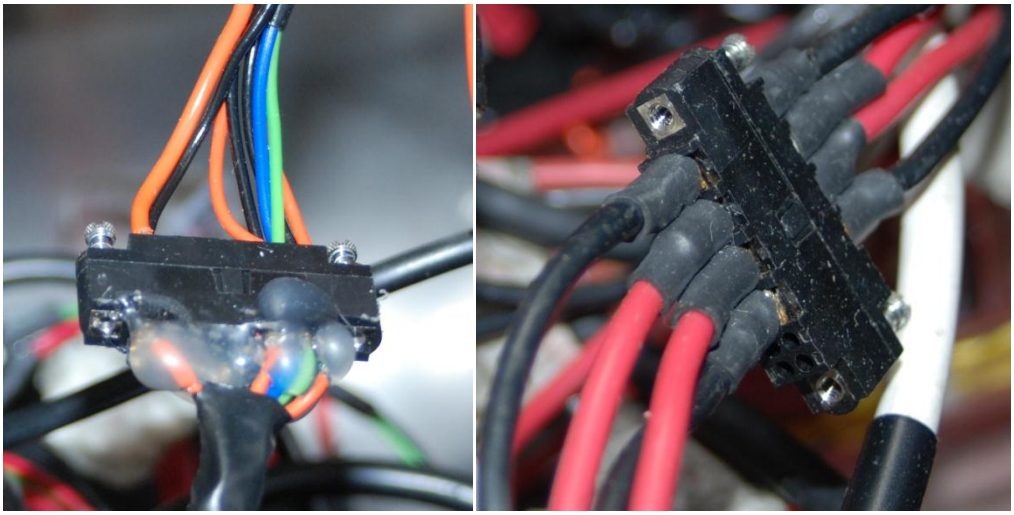


Figure 93: The use of Harwin connectors in the USAR-A

3.3.2.3 Evaluation of 2009/10 Internal Electronics

In the 2009/10 design, wires are highly disorganised and some wires are unused, some connections are damaged and more have yet to be uncovered, some components are semi-permanently connected to others making their removal from the chassis difficult, and finally most components are electrically isolated through improvised means (e.g. electrical tape on the underside of PCBS). Figure 94 below shows the ad-hoc arrangement of components in the USAR-A.

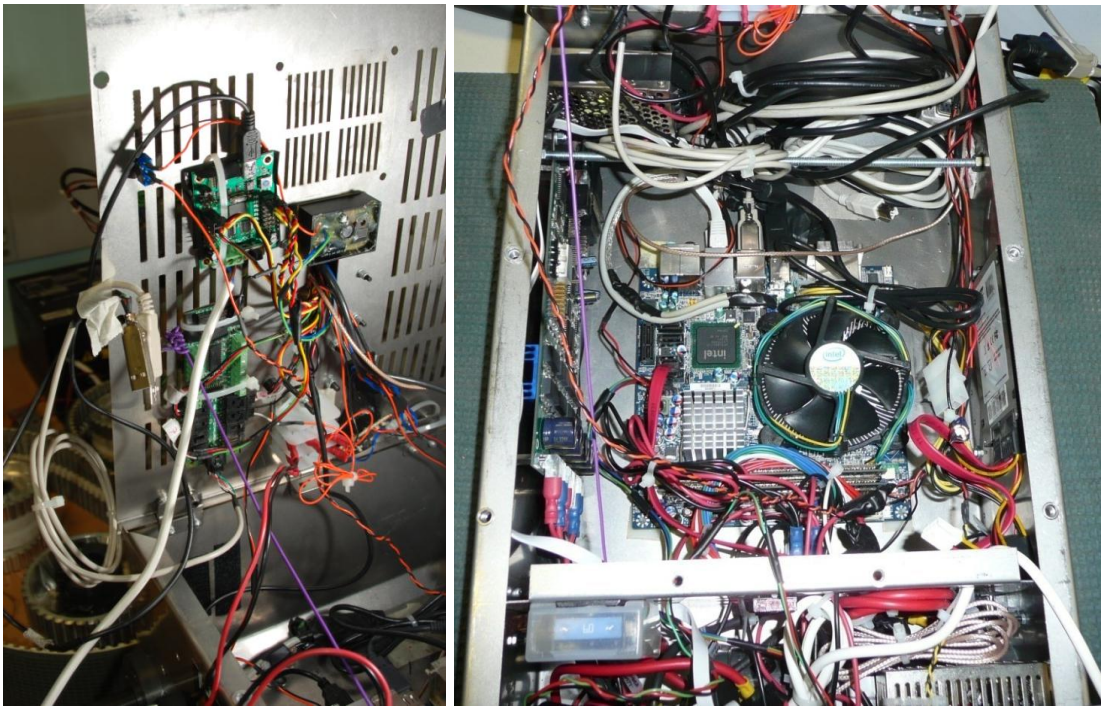


Figure 94: The original USAR-A1 electronic component layout.

The ad-hoc design allows for adequate ventilation due to its open layout and it also allows for the visibility of all components. The incorporation of components directly into the folding lid of the USAR-A lid makes this possible.

The incorporation of a new stack would allow for the reorganisation of wires and reparation of damaged connections, a vibration resistant design, implementation of Harwin connectors for both data (low current) and power (high current) wires, to allow the stack to be easily removed from heavier components. It would also allow for the exposure of important connectors for the purposes of accessibility and incorporation of new devices such as the Xbox Kinect adaptor and Xsens.

If the stack is improperly designed; vibration and jostling of connectors may cause them to come loose. Lack of ventilation in the new design may cause components to overheat during use. Some important connectors may risk becoming inaccessible due to the design of the stack, or shorting of components could occur due to exposed wires and connectors in addition to poor grounding. Also, inadvertent grounding of specific components through the chassis may be an issue.

3.3.2.4 Specification

From the evaluation of the 200/10 platform it was determined that the stack was required to meet the following specification:

- The new stack must allow for adequate ventilation to the motherboard and the motor controller to prevent overheating.
- Components must remain as visible as possible with regards to the stack design.
- Long and defunct wires should be shortened, reorganised or removed where applicable during construction.
- The stack should be designed with vibration reduction in mind.
- Harwin connectors should be implemented to ensure that components external to the stack can be connected or disconnected with ease.
- Necessary connections such as USB ports, servomotor control pins and phidget board ports should remain easily accessible.
- Components should/should not be grounded through the stack as appropriate.
- Appropriate safety measures present in the USAR-A design should be transferred to the new stack of the USAR-A.

3.3.2.5 Design Methodology

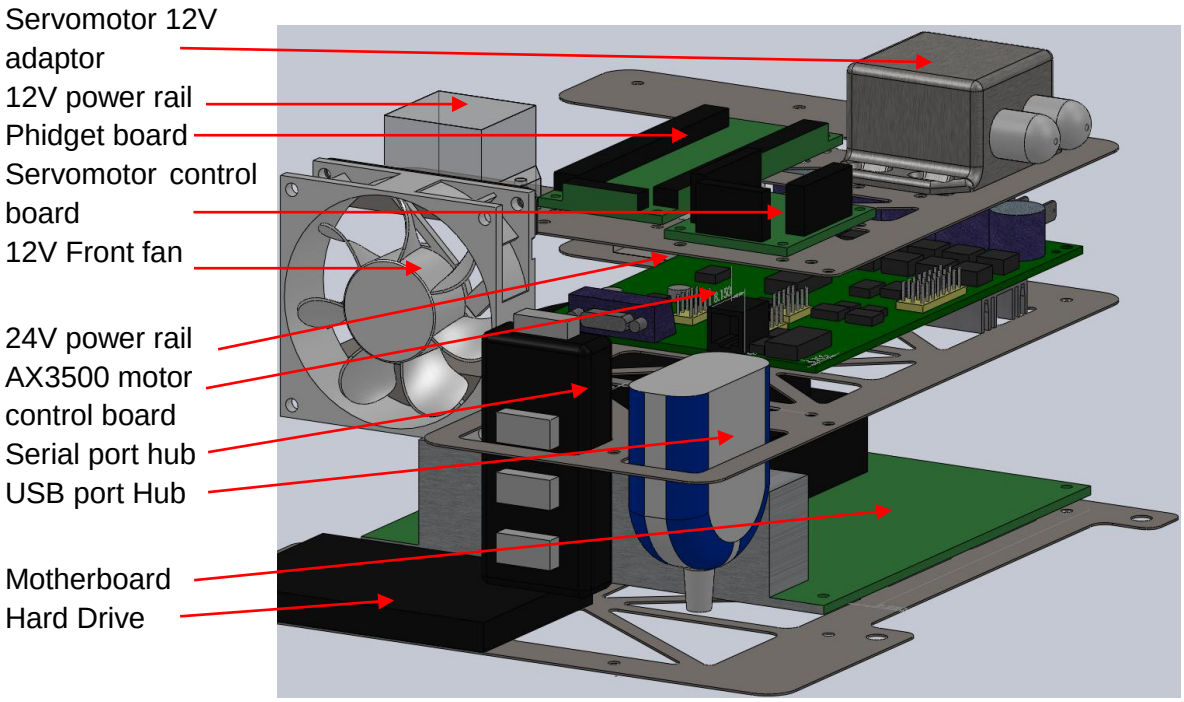


Figure 95: The Solidworks drawing of the USAR-A2 stack

From the above specification the following design features were formulated and a Solidworks drawing was created to visualise the design. The Solidworks drawings of the stacks' plate net plans can be found in Appendix 9: USAR-A Chassis (CAD)

Appendix 10: USAR-A Drive Chain (CAD)

Appendix 11: USAR-A Head (CAD)

Appendix 12.

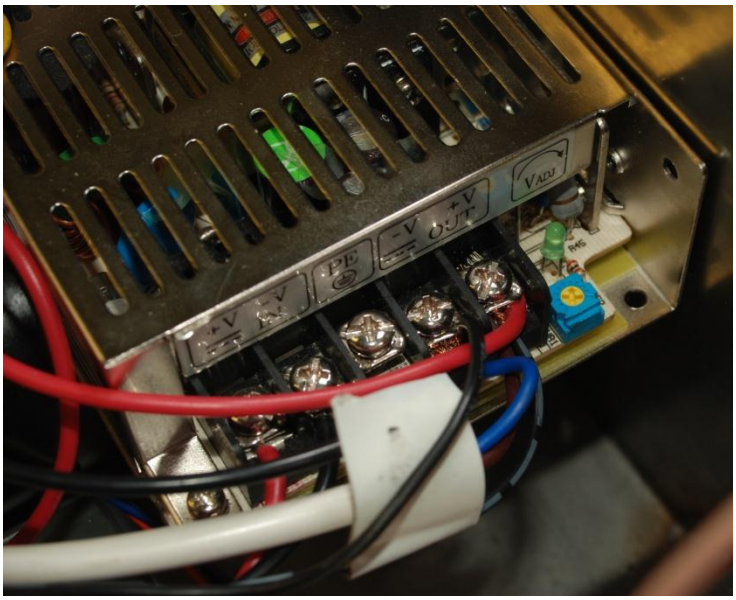


Figure 96: The 12V power converter

The top plate also contained the 5V power converter which is not shown in Figure 96. The 12V power converter (Figure 96) was to remain attached to the inside of the chassis due to its size and lack of space on this stack design.

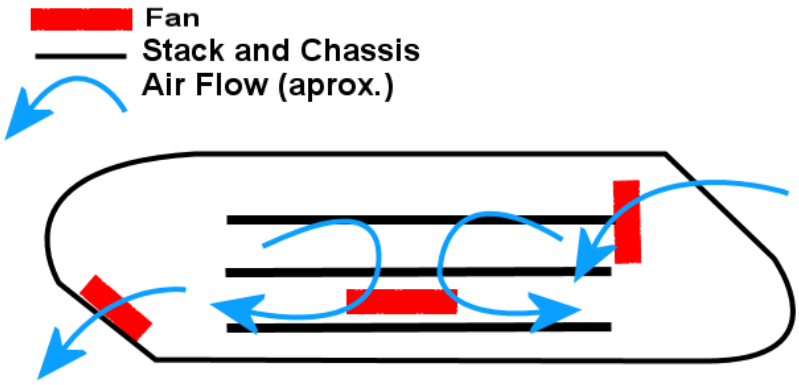


Figure 97: The predicted circulation of air inside the USAR-A2 chassis

The overall stack design was tall and open to allow for air circulation around the motherboard's built-in fan and the AX3500 motor controller just above. Vents were also added to each of the layers for the purpose of ventilation. The front fan was positioned to align with the front vent in the USAR-A's new chassis. The height of the stack, coupled with the similarly long length of the spacers, would mean that the upper levels of the stack would be less vulnerable to horizontal vibration. The addition to two foam blocks on the base of the stack would further dampen the effects of vibration.

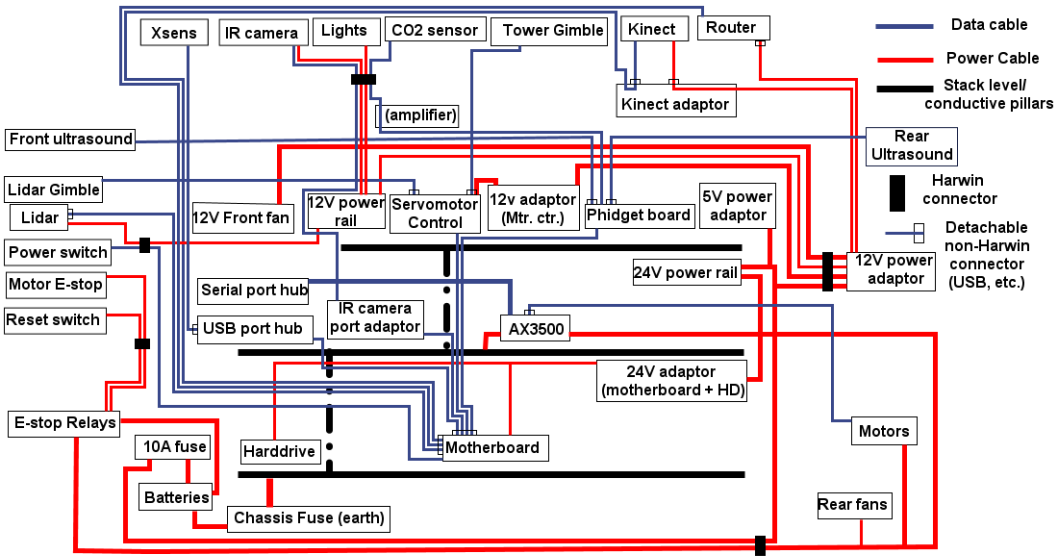


Figure 98: Block diagram of the USAR-A systems.

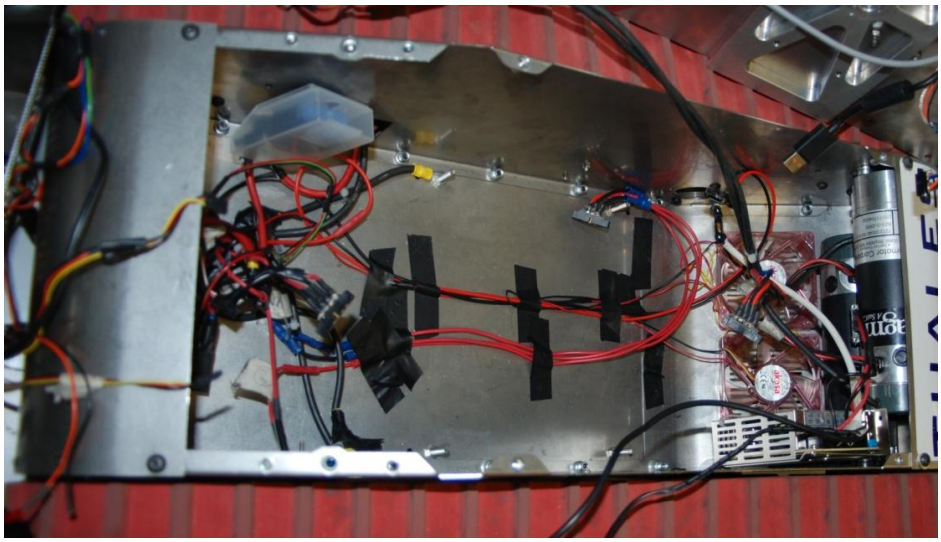


Figure 99: The arrangement of components inside the USAR-A chassis

Figure 98 demonstrates the use of Harwin connectors to separate components secured to the chassis of the USAR-A from the stack, provided they do not already have an easily detachable connector like a USB port or Ethernet port (the Xsens positional sensor and AX3500 motor speed control board respectively). The Harwin connectors are used for their secure connection, ease of connection and removal and appropriate electrical isolation between incorporated wires in their casing. Also demonstrated is the use of fuses to isolate high currents travelling through the chassis of the USAR-A. Conductive spacers were used to provide a common ground to the AX3500, and respectively nonconductive nylon spacers were used to isolate the motherboard and other components that would be at risk from this grounding of the stack. Finally, the AX3500's motor power connectors were positioned at the rear of the stack, closest to the motors, along with the 24V power rail. The latter component

would require specific wires to run from the front of the chassis to the rear, below the stack as demonstrated in Figure 99.

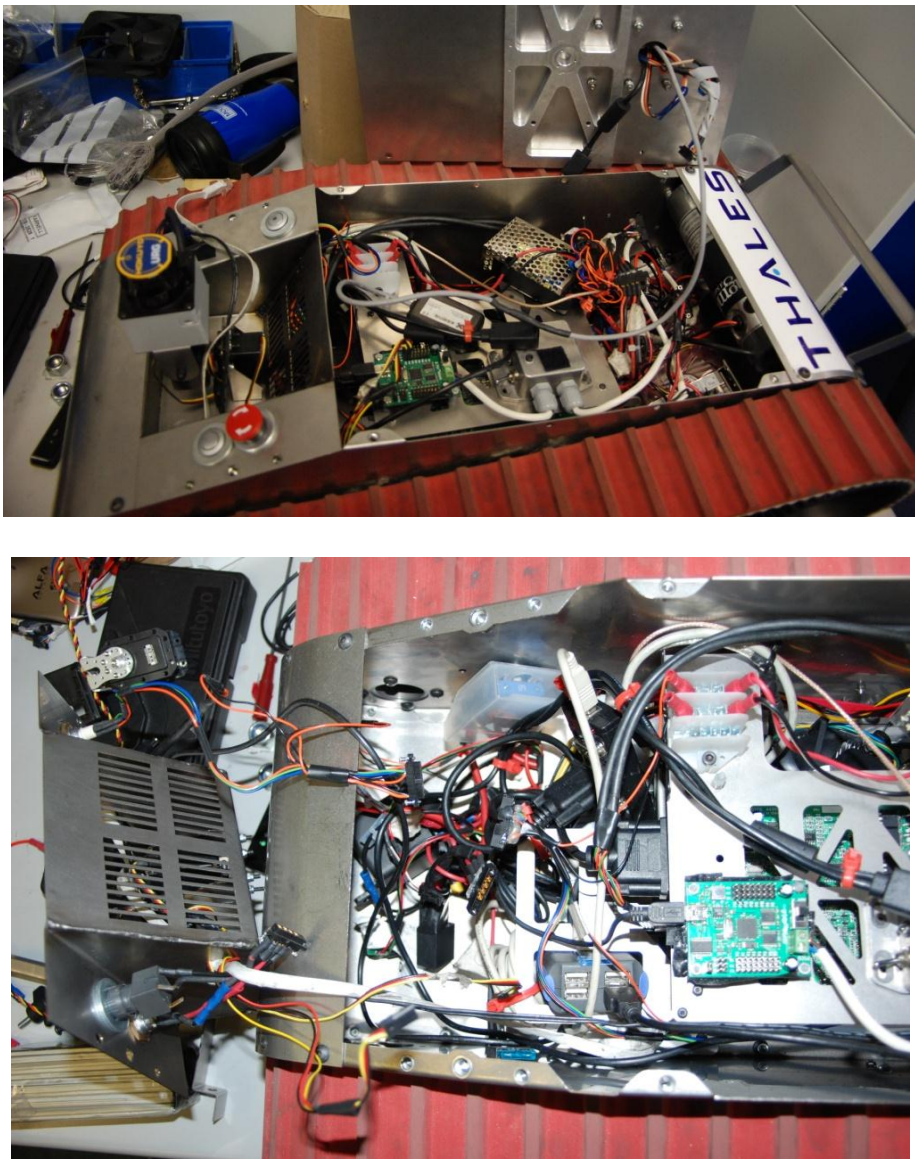


Figure 100: Demonstrating the accessibility to components on the (a) top stack and (b) USB hub under the (a) lid and (b) front plate respectively.

The USB hub, USB motherboard ports, serial port hub and the Ethernet port used for the motor encoders on the front of the AX3500 motor control board were positioned at the front of the stack for accessibility once the front vent was removed. The phidget board and servomotor controller were positioned on the top of the stack to be accessible when the lid was removed.

3.3.2.6 Evaluation

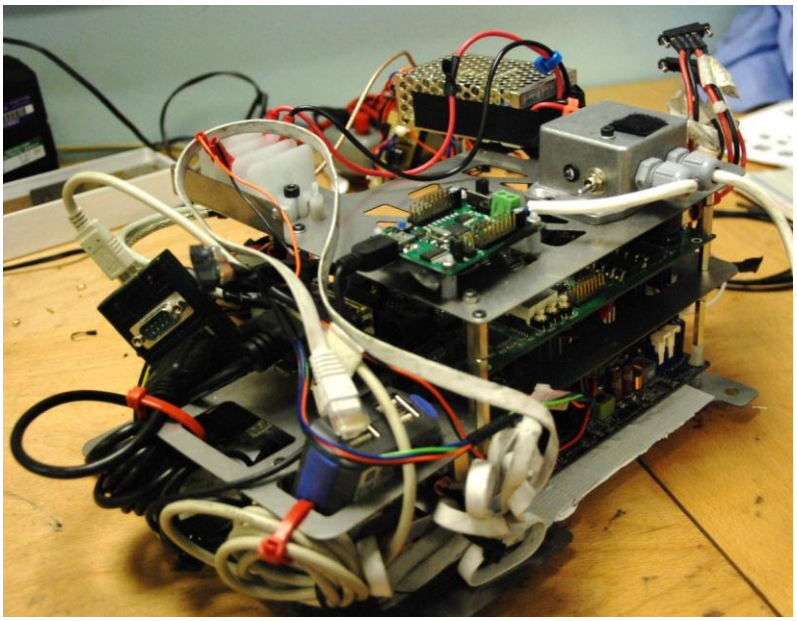


Figure 101: The USAR-A stack, after the RoboCup Rescue competition

The overall design of the stack was resistant to vibration, appropriately ventilated and structurally secure as intended. Aside from the exceptions below the electronic function of the USAR-A2 were not affected after being implemented into the stack, and there were no cases of connectors coming loose due to vibration. Lengths of wires that could not be shortened were appropriately secured with cable ties, both inside and outside the casing.

One design problem existed regarding the positioning of the battery connectors and their housing that was to be copied from the USAR-T. If the housing for the internal battery connectors was implemented the stack would not be able to physically fit inside of the chassis. It was decided to remove the unnecessary housing and lengthen the cables to the battery connectors so that they extended outside of the chassis and into the battery compartments. This allowed the stack to fit inside the chassis.

An early technical problem was the access to the phidget board's ultrasound and CO_2 sensor data. However, after testing the connection and operability on the USAR-T the problem was determined to be purely software based, and so the responsibility was left with the computer science team to solve. The phidget board was eventually removed and incorporated into the USAR-T during the RoboCup Rescue competition.

Figure 102: The connection of the 12V power converter to the chassis via velcro

The method of securing the wires to and from the 12V power converter (Figure 96 and Figure 102) via screw threads proved difficult and time consuming, especially when attaching Kinect's adaptor. A solution would have been to utilise crimped adaptors with circular holes, (similar to those implemented on the USAR-T arm motors) for the securing screw thread to slide through in parallel. This would make the addition of new connections to the 12V power converter as simple as attaching the crimped connector to the new wire. It was not implemented due to time and availability constraints at both the computer science's testing period leading up to and during the competition.

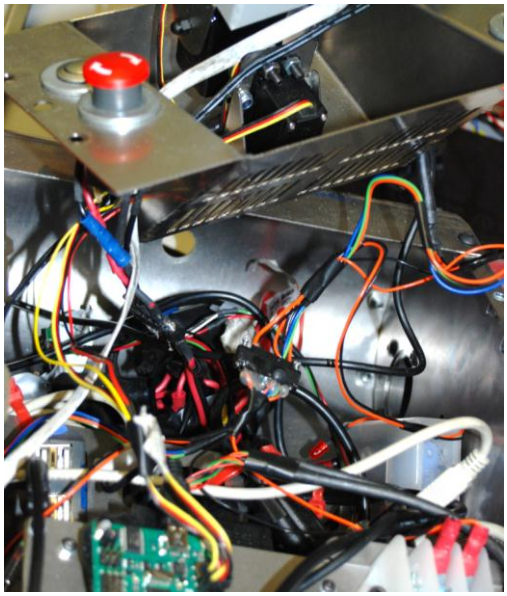


Figure 103: The lead to the E-stop switch and reset button are particularly tight.

Similar, minor, technical issues would include: The length of the leads to the emergency stop switch being difficult to access when the front vent was attached Figure 103. And the

attachment of the 12V power converter to the side of the chassis via Velcro (Figure 102) where a more suitable

method, such as nuts and bolts in the chassis casing, would have been preferable. These were also not corrected due to the time constraints mentioned earlier. The latter would require the drilling of new holes in the chassis itself after disassembly.

The front fan was no longer required as the ventilation provided by the motherboard's fan was adequate; complimentary to the stack's spacious design. The rear fans would operate when the power to the motor and AX3500 was applied and provided additional cooling to the stack during operation (in the previous chassis these two rear fans could only supply ventilation to the motors due to a separating metal plate).

During testing at the RoboCup Rescue competition it was discovered that the motherboard and hard drive's 24V power adapter's fuse had blown. The stack's connections were checked and the causes of the problem remained uncertain, but after replacing the micro fuse (by soldering another atop it) the USAR-A continued to function normally.

3.4 Future System Improvements

3.4.1 Mechanical System Improvements

The chassis structure now works very well so there is currently no reason for change; however the battery holders are incomplete. A hinged door should be added to the battery holders in order to hold the batteries in place whilst allowing easy access.

There are a lot of areas that need improving with the autonomous head. The head should be made lighter and all sensors should fit within it. A more substantial and stable stand should be manufactured. A method for getting two degrees of freedom whilst keeping the head stable should be designed.

3.4.2 Electronic System Improvements

There are only a few recommendations that are associated with this section of the robot. For further standardisation between the models, the Mag motors should be replaced with Maxon brand motors. This will give more power, replace the damaged motor and give uniformity between platforms. This modification will not need many parts to be redesigned and manufactured as the parts have been standardised enough already.

The stack's overall design (and height inside the chassis) can accommodate additional layers of components provided the initial specification of important connectors remaining accessible on the top of the stack is fulfilled (the servomotor controller and phidget board). There is also space for a replacement phidget board on the existing top stack and also space available for the front fan to be incorporated for any future need for additional cooling of the stack.

The problem of securing the 12V power converter to the chassis (identified in section3.3.2.6) still needs to be addressed in addition to finding a simple method of integrating new components to the mechanically overburdened output connectors. This could be achieved via the method using crimped connectors and spacers on lengthened screw threads as discussed in section3.3.2.6 or by the purchase of an additional power rail to be connected 12V converter output. Another minor issue requires the lengthening of the wires leading to the emergency stop button.

4 References

Active Robots. *Robot Kits, Gripper Kits*. 1 Feb 2011. http://www.active-robots.com/products/robots/lynx-a-gripper.shtml (accessed Feb 1, 2011).

BeagleBoard.org. *beagleboard.org*. April 2011. http://beagleboard.org/.

Chen, Y J, E C Haas, K Pillalmarri, and N C Jacobson. "Human-Robot Interface: Issues in Operator Performance, Interface Design and Technologies." 2006. http://handle.dtic.mil/100.2/ADA451379 (accessed 11 20, 2010).

Geartronics. *Pneumatic Paddle Shift System*. 3 March 2011. http://www.geartronics.co.uk/paddleshift.htm (accessed April 6, 2011).

Harwin Plc. *Harwin PLC - Electrical Connectors and SMT PCB hardware*. 2011. http://www.harwin.com/ (accessed October 2010).

Jacoff, Adam. *RoboCupRescue Robot League Progress Update and Future Directions*. NIST, 2009.

Massy-Westropp, N., W. Rankin, M. Ahern, J. Krishnan, and T.C. Hearn. "Measuring grip strength in normal adults: Reference ranges and a comparison of electronic and hydraulic instruments." *The Journal of Hand Surgery* 29, no. 3 (2004): 514-519.

Roboteq. *Brushed DC Motor Controllers*. April 2011. http://www.roboteq.com/brushed-dc-motor-controllers/brushed-dc-motor-controllers-selector.

ROS (Stanford, Willow Garage). *ROS.org*. April 2011. http://www.ros.org.

RS Online. *Finger Type Mechanical Gripper*. n.d. http://uk.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=0478040#header (accessed Feb 1, 2011).

—. *Finger-Type Mechanical Gripper*. n.d. http://uk.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=3663626#header (accessed Feb 1, 2011).

Warwick Mobile Robotics. *2009/2010 Technical Report*. University of Warwick, 2010.

5 Appendices

Appendix 1: Arm System (CAD)

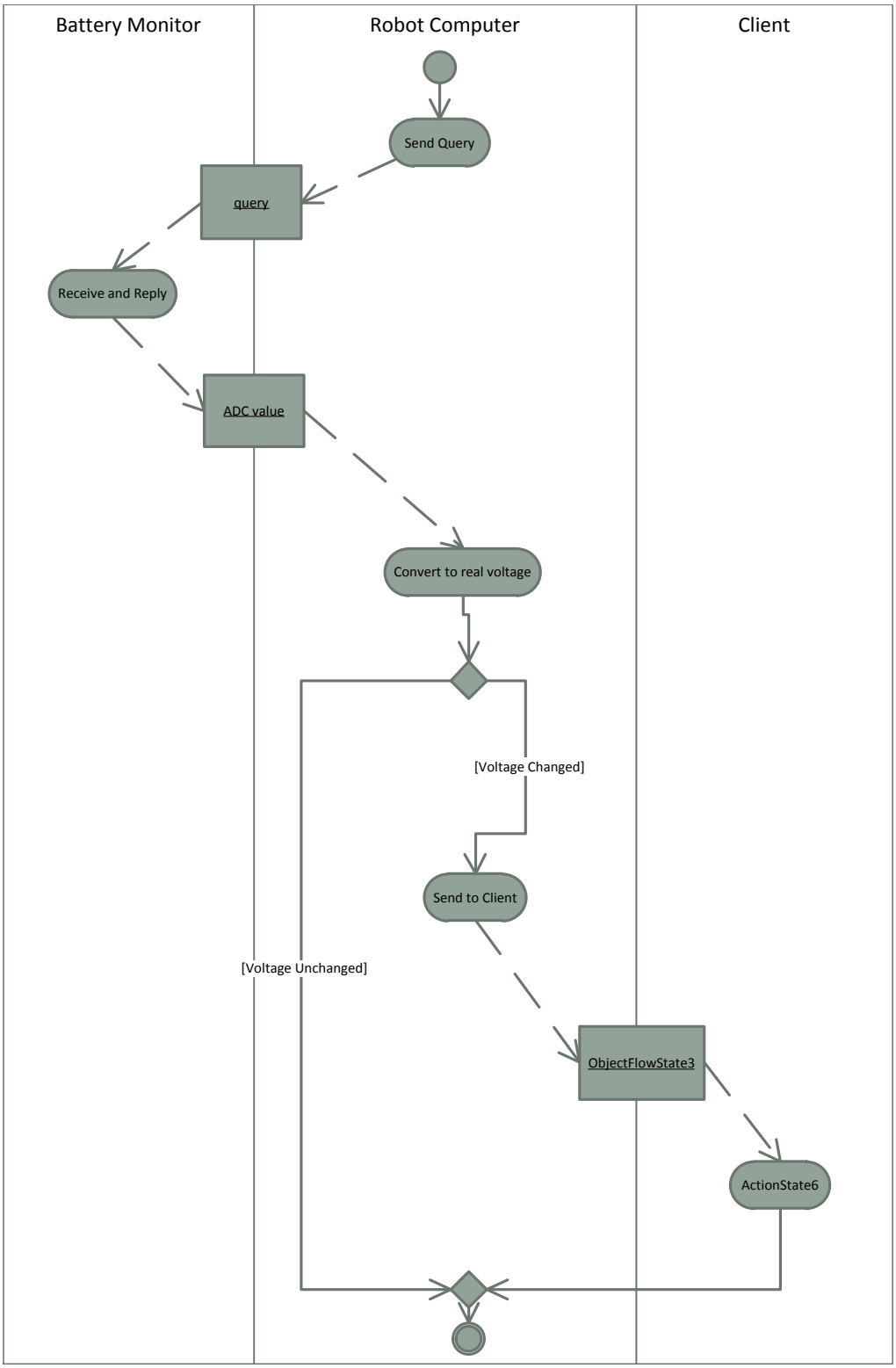
Appendix 2: Head System (CAD)

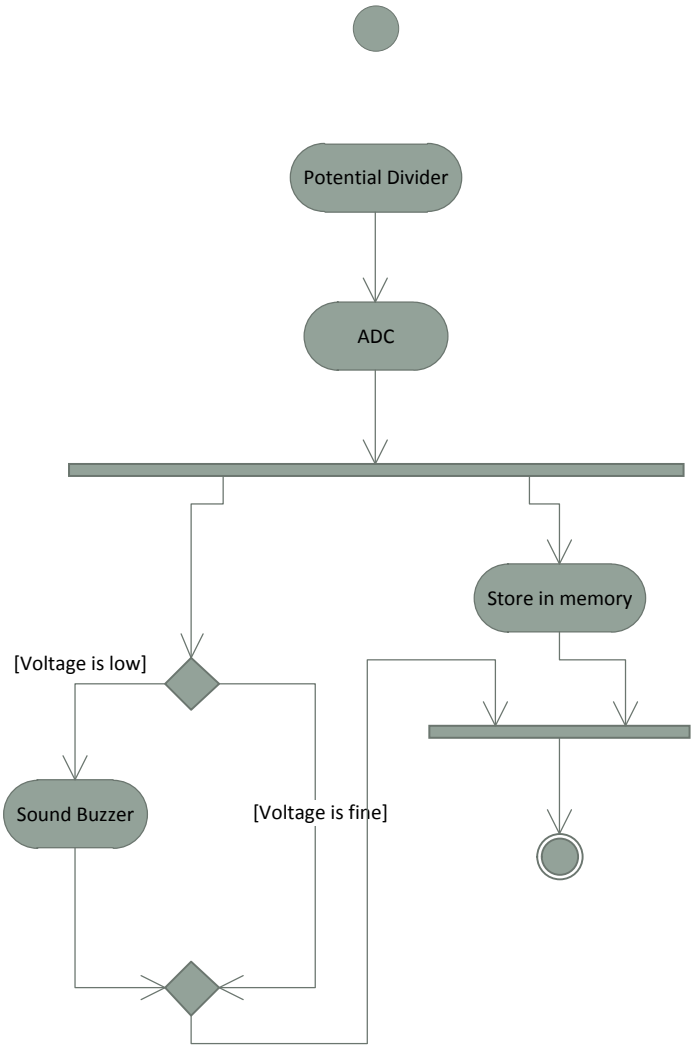
Appendix 3: USAR-T Chassis (CAD)

Appendix 4: Stack Handle (CAD)

Appendix 5: Stack Casings (CAD)

Appendix 6: Battery monitor activity diagram




Potential Divider
ADC
Store in memory
[Voltage is low]
Sound Buzzer
[Voltage is fine]

Appendix 7: Battery Monitor Schematic

Appendix 8: Battery Monitor Source Code

```
#include <avr/interrupt.h>
#include <util/delay.h>
//#include <uart.h>

//#define FOSC 8000000 //clock speed
//#define BAUD 9800
//#define MYBURR 6

//int buzzer = 0;
char msg = 0;
char msg1= 0;
char msg2 = 0;
int checkHigh(double voltage);
int checkLow(double voltage);
void USART_init(unsigned int ubrr);
void transmit();
void newMessage();
short int i = 0;
short int transmitTrue = 0;
char HexToChar(int hex);


ISR(ADC_vect)
{

    //get reading
    int reading = ADCH<<2;
    int lsb = ADCL/64;
    reading += lsb;
    double voltage = reading * 5 /1024.0;
    voltage = voltage * 118 / 18;
    PORTB ^= (1<<PORTB3);
    //buzzer control
    if(checkLow(voltage)==1)
    {
           PORTB |= 2;
    }
    else
    {
           PORTB &= ~(1<<PORTB1);
    }

           msg = ADCH<<1;
           lsb /= 2;
           msg += lsb;
```

```
    newMessage();

}

int checkHigh(double voltage){
    if(voltage > 25.5)
    {
           return 1;
    }
    return 0;
}

char HexToChar(int hex)
{
    char hexChar = 0;

    if((hex >= 0) && (hex <= 9))
    {
           hexChar = 48 + hex;
    }
    else if((hex >= 0x0A)&&(hex <= 0x0F))
    {
           hexChar = 55 + hex;
    }
    return hexChar;
}

void newMessage()
{
    int temp = msg>>4;
    msg1 = HexToChar(temp);
    temp = msg%16;
    msg2 = HexToChar(temp);

}


int checkLow(double voltage){
    if(voltage < 18.5)
    {
           return 1;
    }
    return 0;
}

ISR(USART_UDRE_vect)
{
```

```
// transmit();
}


ISR(USART_RXC_vect)
{

    char temp = UDR;

    if(temp == 63)
    {
           PORTB ^= (1<<PORTB1);
           i = 0;
           transmitTrue = 1;
           transmit();
    }
}



ISR(USART_TXC_vect)
{
    transmit();
}

void transmit()
{
    switch(transmitTrue)
    {
           case 1:
                  switch(i)
                  {
                         case 3:
                                UDR = 10;
                                i = 0;
                                transmitTrue = 0;

                                break;
                         case 2:
                                UDR = 13     ;
                                i++;
                                //transmitTrue = 0;

                                break;
                         case 1:

                                UDR = msg2;
                                i++;
```

```
                                break;
                        case 0:
                                UDR = msg1;
                                i++;
                                break;

                }
                break;
        case 0:
                break;
    }
}

int main()
{

    USART_init(51);

    DDRB = 0xFF;

    sei();

    PORTB = 2;
    ADCSRA = (1<<ADEN)|(1<<ADFR)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

    ADMUX = (1<<ADLAR)|(1<<REFS0);
    ADCSRA |= (1<<ADSC);

    for(;;);
}


void USART_init(unsigned int ubrr)
{
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;
     //UBRRH = 0;
    //UBRRL= 51;

    UCSRB = (1<<RXEN)|(1<<TXEN)|(1<<RXCIE)|(1<<TXCIE);

    UCSRC = (1<<URSEL)|(0<<USBS)|(1<<UCSZ1)|(1<<UCSZ0);

}
```

Appendix 9: USAR-A Chassis (CAD)

Appendix 10: USAR-A Drive Chain (CAD)

Appendix 11: USAR-A Head (CAD)

Appendix 12: USAR-A Stack (CAD)