Autonomous LEGO© Climber Project - Stingray

Luke Cole cole@lc.homedns.org

Adam Fran Nagy-Sochacki u2546772@anu.edu.au

Australian National University
http://www.anu.edu.au

September 25, 2007

1 Executive summary

The LEGO© robot outlined in this report was designed to climb randomly positioned holes on a steep inclined ramp. The robot is built from primarily standard LEGO© components and a LEGO© 1.0 Mindstorm kit. The computer (RCX 1.0) only has three inputs and three outputs thus limiting design choices. Two outputs are used to control three motors (two motors to extend the arm) the other to actuate a cable system for panning. One of the inputs of the control unit was used to detect holes via an IR sensor and the extremities of arm movement through the use of two touch sensors. The next input was used for the encoder on the tail motor which controls the cable mechanism that pans the arm. The final input was used for an encoder to measure the robots angle relative to gravity. As a whole the robot proved to be very stable, predictable, reliable, robust and extendable when given most climbing tasks.

2 Climbing Strategy

2.1 Ideal Algorithm

Ideally a climbing algorithm should be independent of a wall. Figure 1 outlines such a climbing strategy. The process is basically a simple loop of first searching for a hole, then hooking into that hole. However the routines for searching, refining the hole search and hooking into a hook are not trivial. Furthermore a gyro sensor is used which also adds some complexity to the algorithm.

The current search routine simply searches forward then at various increasing angles to the left and right. Searching left or right is based on the current angle of the gyro, that is, if the gyro is tilted to the left, the robot will search to the right. Clearly this is the most time consuming process and many different approaches could be researched to determine the fastest hole searching method, such as learning from previously found holes.

Once the IR sensor detects a hole the search routine stops and then begins a refining search routine to pin point the middle of the hole (i.e. the maximum sensor reading). This is done by taking a IR sensor reading then allowing the arm and pan axis to move in a random direction, then another IR sensor reading is taken. If the IR sensor reading is showing the IR sensor is moving away from the hole (e.g. reading reducing) then toggle the direction of the appropriate axis.

Hooking into the hole is also not the most trivial routine. To achieve this the arm must simply extend forward to hooking into the hole. However the base must pan relative to the arm to compensate for the changing centre of mass.

The gyro sensor is also monitored to realign the robot to keep stability and reduce the chance of a fall due to being of balance.

It should be noted that this algorithm finishes when no more holes can be found and all possible hole positions searched.

2.2 Competition Day Algorithm

The competition was to climb the wall to the highest point as fast as possible, so due this and our time constraints a trivial algorithm was implemented which had some knowledge of the hole positions on the fabricated wall used on competition day. Unfortunately the trivial algorithm was not complete so it was unable to complete the wall climb.

3 Mechanical design

3.1 Gripper design

The gripping design of the robot was initially aimed at being constructed entirely of lego pieces. Through rigorous testing of climbing the wall it was found that while a lego claw design could be successfully used they failed to offer the stability desired of the robot. This resulted in

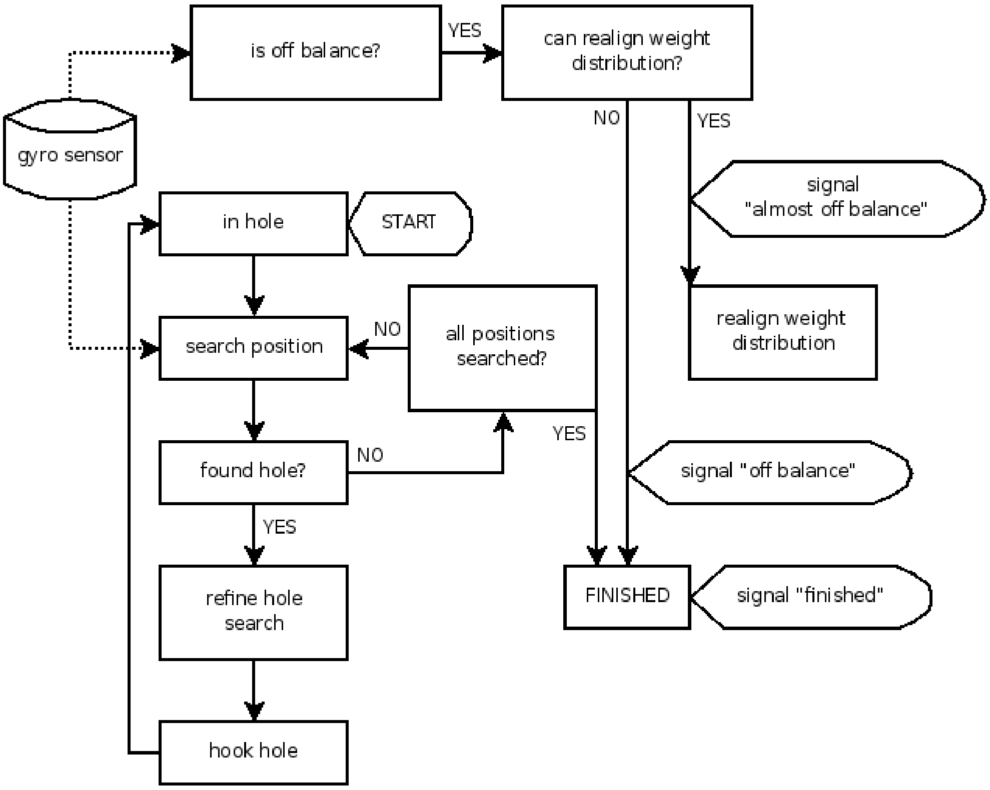

Figure 1: Flow Chart of Ideal Climbing Strategy Algorithm

the construction of aluminium replacement claws for the head of the robot. These new claws are similar in design to lego pieces, however offer a lower profile and added length. This new design is more durable and reliable. Increasing the life span and functionality of the robot.

The mechanism itself is a three claw design, the head of the robot contains the two aluminium claws while the third is attached to the main base frame of the robot. The claws on the head of the robot are surrounded by a guidance system to aid the positioning of the third claw, which upon engagement positions itself between the two claws on the robot head. The dual claw mechanism on the head of the robot allow help to balance the robot and restrict a pivoting motion when initially engaged, this is aimed at minimising possible disparities of knowing where the robot is. This is illustrated in Figure 2. The singular claw attached to the base module operates in a similar manor as to those on the head. A detailed picture of the mechanism and its attachment to the robot base can be found in Figure 7.

Each claw is actuated by a spring, these springs are used to force the claws out into a position to support the mass of the robot. When the robot is climbing however the weight of the robot and absence of holes forces the claw to lie flat until such time it is able to spring free.

3.2 Arm design and actuation

Initially a scissor mechanism was proposed for the robots reaching mechanism, this initial design seemed to offer many benefits. Once the scissor mechanism was implemented on a robot, as seen in Figure 3, it was found to have several disadvantages that rendered its implementation not viable. Primarily, the excessive free play in the mechanism was of greatest concern. But more importantly the scissor mechanism was unable to pull is base up when placed on a wall.

Figure 4 shows the final arm gearbox that supports a geared arm as showen in Figure 8. The motion of the robot in terms of translation uses a prismatic joint of a rack and gear arrangement. In order to balance the weight, increase torque and run time to move the robot at a consistent constant velocity the prismatic joint is actuated by two motors. The motors are geared down to allow for greater torque and smother motion of the arm mechanism. This is useful due to the size of the robot. Whilst weight is added to the robot to allows better accuracy when using a timing based approaches to calculate the prismatic position.

Rotation of the arm is achieved through a unique system of pulleys as shown in Figure 5, that is actuated via fishing wire from a single motor in the tail of the

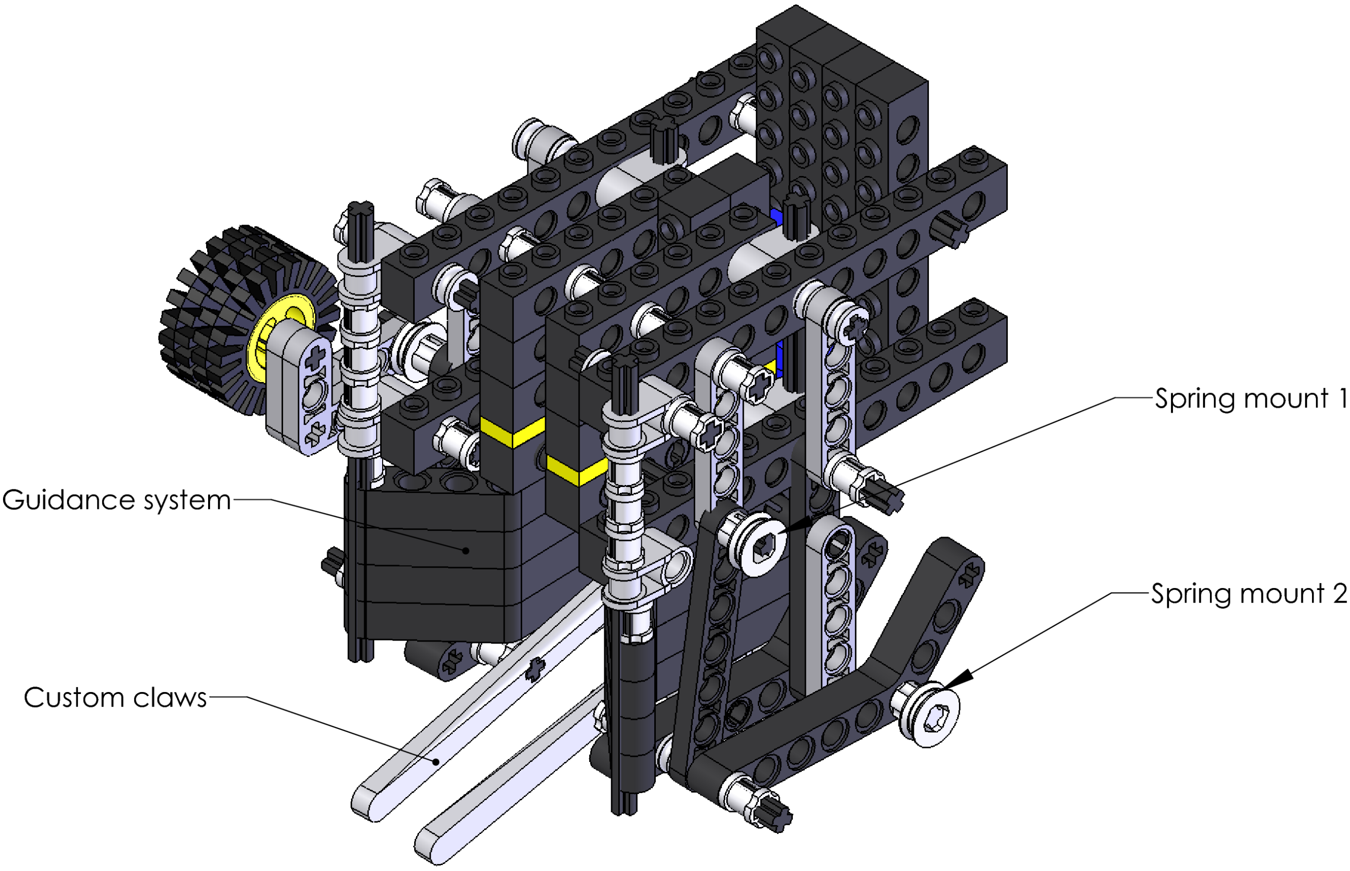

Figure 2: Head construction

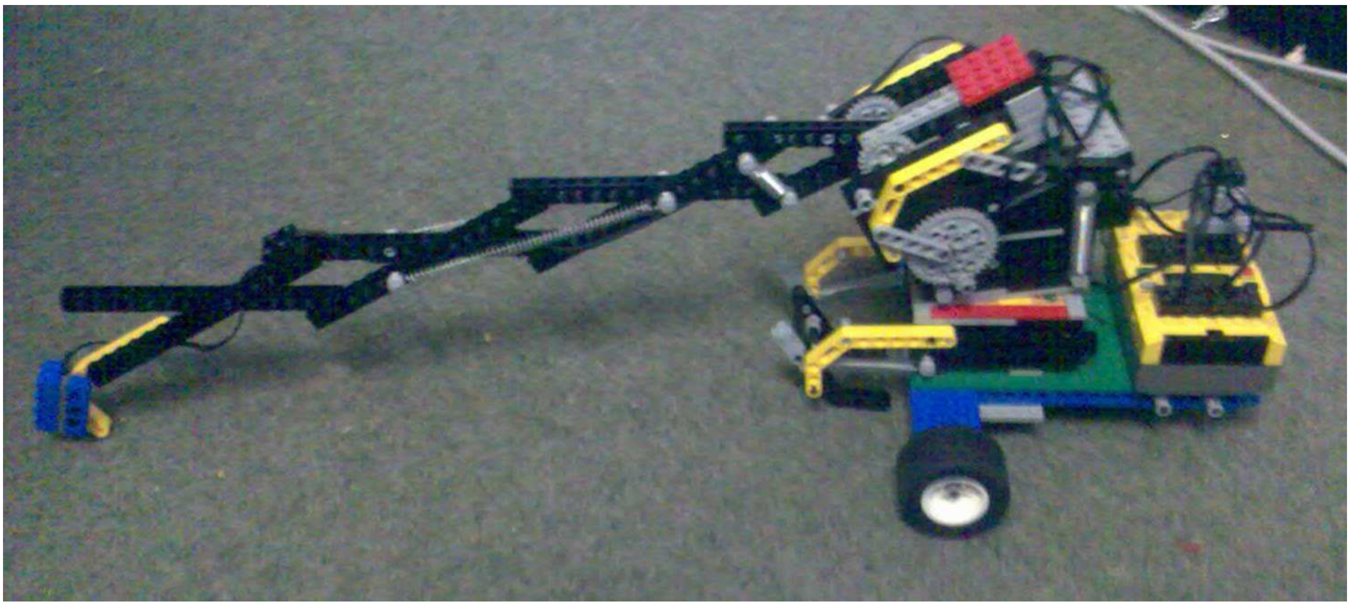

Figure 3: Initial design using a scissor mechanism

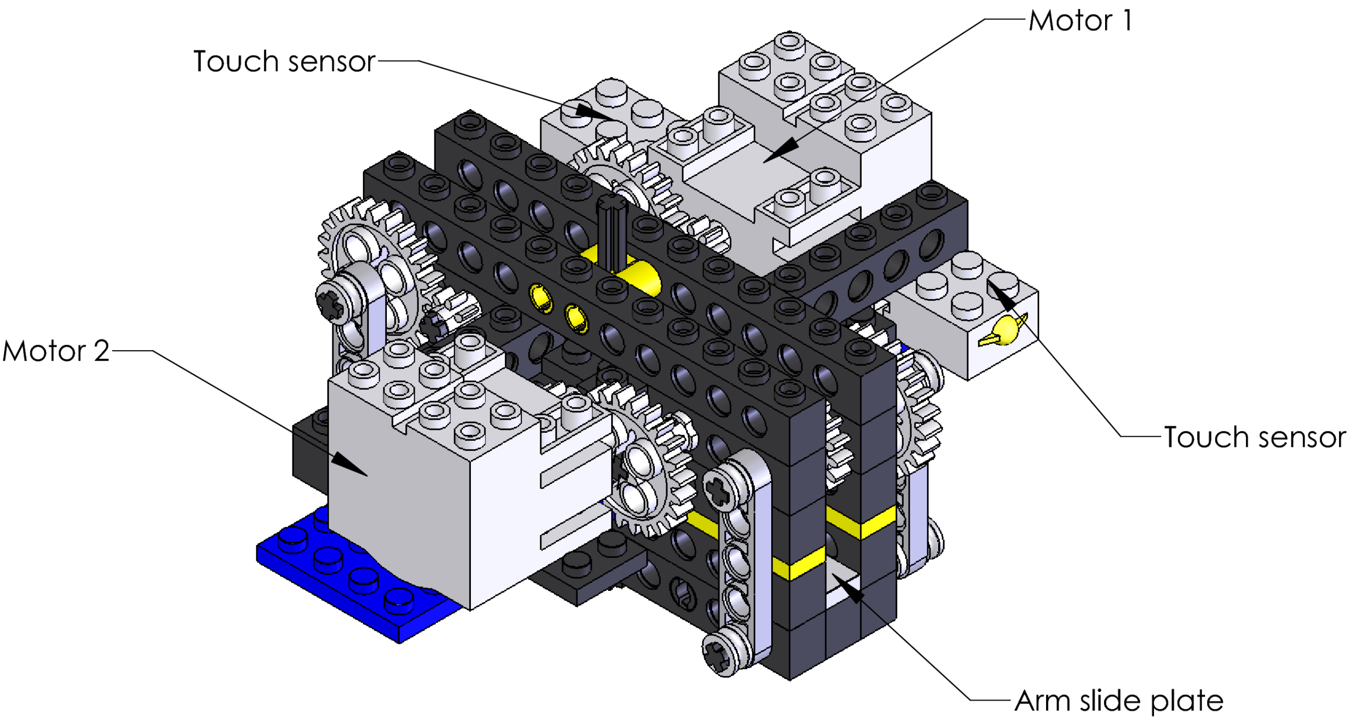

Figure 4: Arm gearbox construction

robot as shown in Figure 6. This design not only drives the panning motion of the robot but helps to support the structure. The mechanism links the head and tail to each end of the base. The actuating motor in the tail of the robot operates a series of gears to increase the torque at the main pulley. Consequently, the speed of rotation of the robot is also decreased to a more controllable and predictable motion. The cable pulley system allow precise control of the pan motion of the robot but is also unfortunately reasonably complicated. Given this complexity, the cable mechanism also limits modifications to the arm mechanism. Any major changes to the head, tail, arm or base would result in the cable system requiring re-cabling, which is a time consuming task (approximately 3 hours).

Throughout the design process adjustability of the robot was kept in mind, this resulted in the placement of gear systems that can be easily interchanged. Both the translation and panning of the arm can be modified via simple changes to the easily accessible gears. Such a feature allows the design to modify torque or speed in either direction.

3.3 Base

Figure 7 shows the base of the robot cradles the arm of the robot, the base is then free to rotate about a central axis. The base is not only designed to support the arm, but increases stability to the robot. This is clearly visible when the robot attempts complicated maneuvers at large angles, at these times the reasoning for the vehicle large track becomes most apparent.

3.4 Final design

Overall the robot is a modulated design as shown by the assemble view of the modules in Figure 8 that allows for somewhat easy assembly and disassembly. The main complexity of the robot is a consequence of the pulley mechanism used to actuate the panning motion. The robot is also quite large and weighs 1154g.

Given the nature of the task there was no need to have any more then 2 degrees of freedom for the robots motion. The rotational and prismatic joint are set up in such away to allow a two dimensional scanning over the wall as the robot climbs. This configuration is demonstrated in the schematic shound in Figure 9.

Various calculations are outlined in Appendix A and B which show the robots kinematics and forces present.

4 Sensing Strategy

The robot is equipped with five sensors: one IR, two encoders and two touch sensors. The IR sensor is used to detect holes. The two touch sensors are used to detect when the arm has reached its extensions. One encoder is used to measure the movement of the cable mechanism and hence the pan angle. The final encoder is used to measure the angle to the vertical and hence provides a

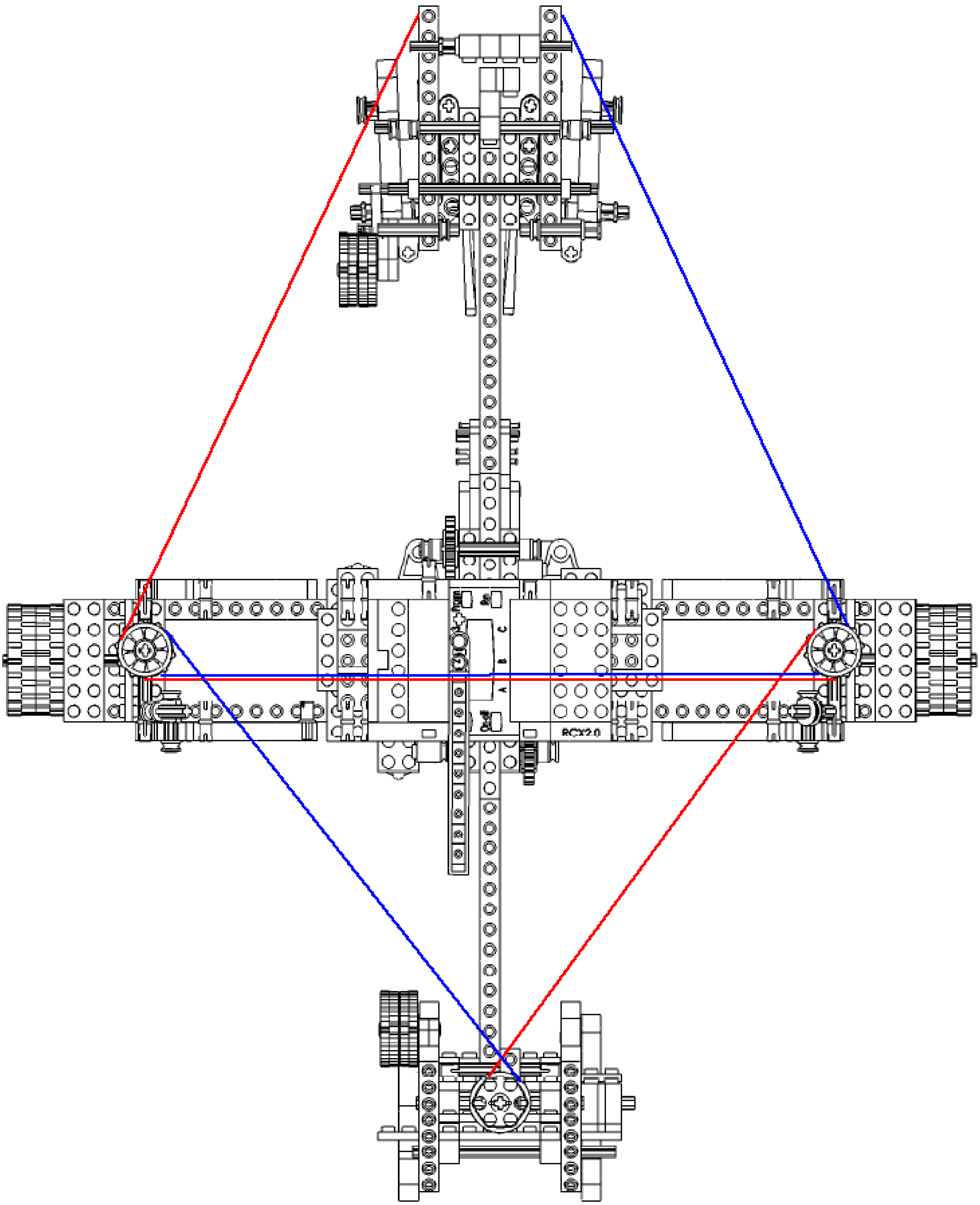

Figure 5: Cable Mechanism. Red cable pans the head one direction, while the blue cables pans the head in the opposite direction.

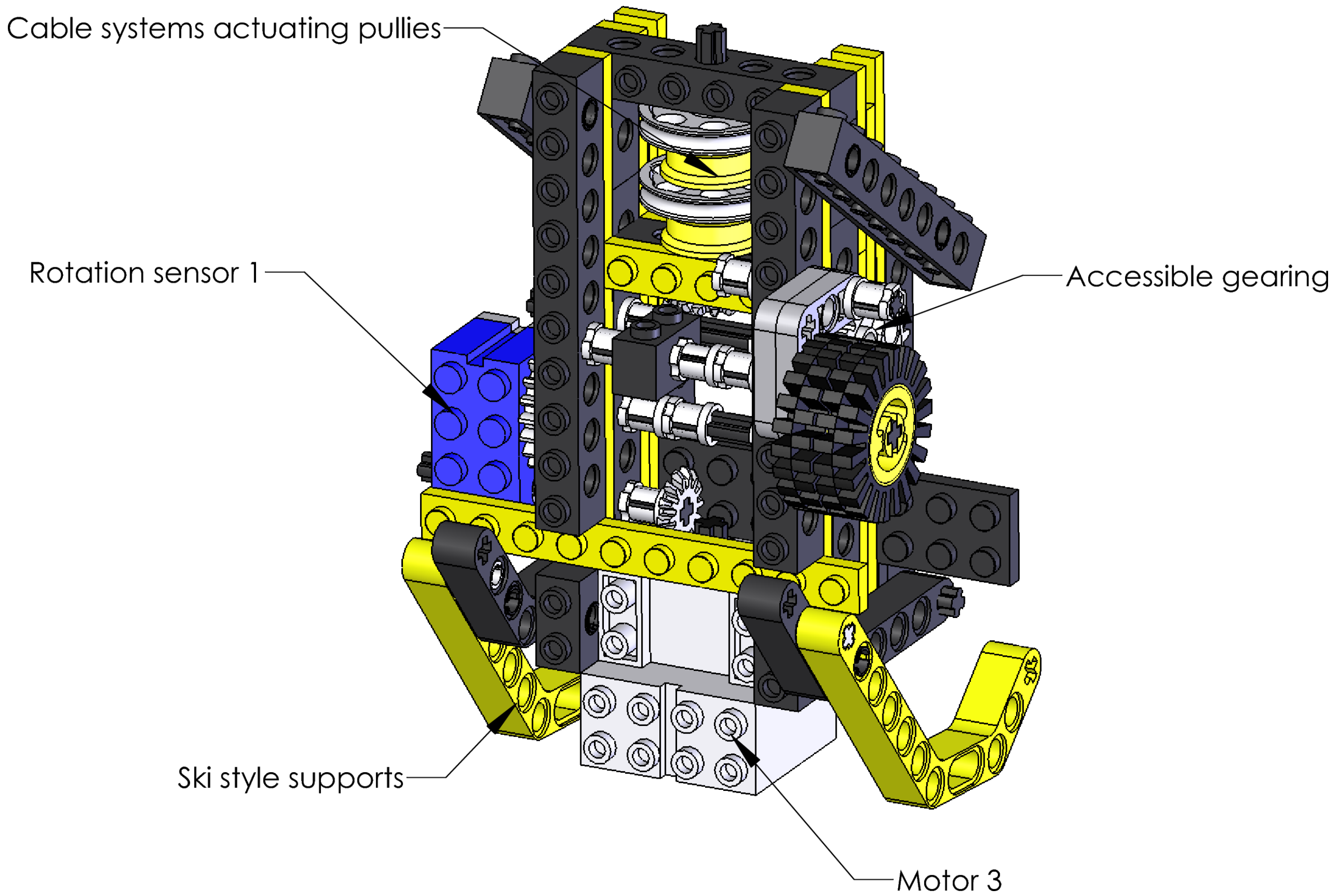

Figure 6: Tail construction

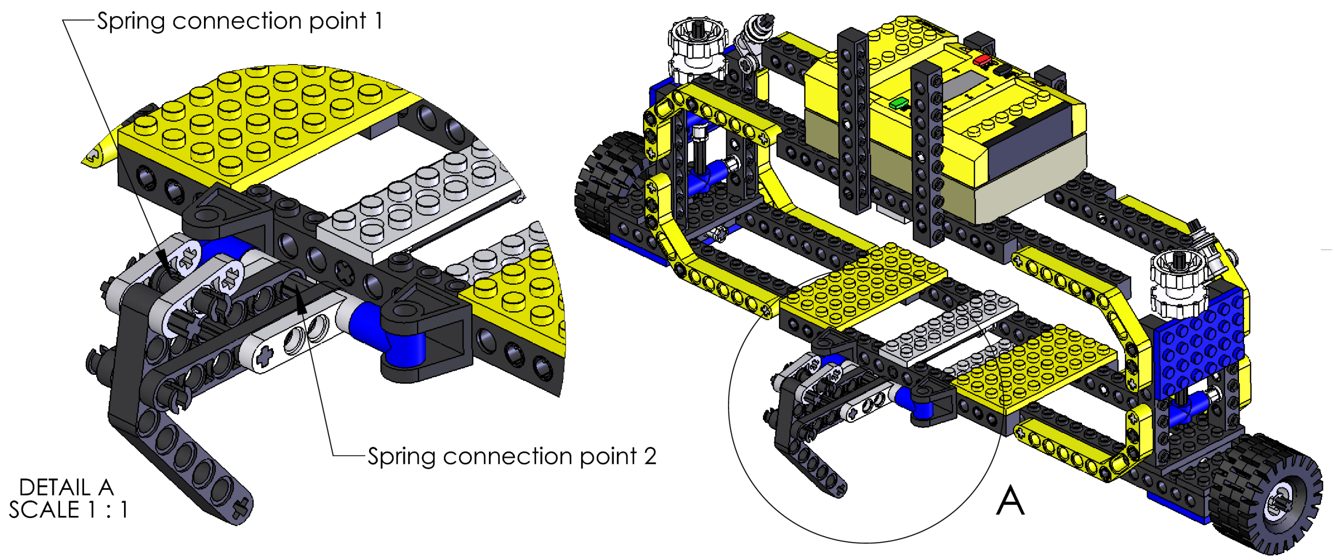

Figure 7: Base construction

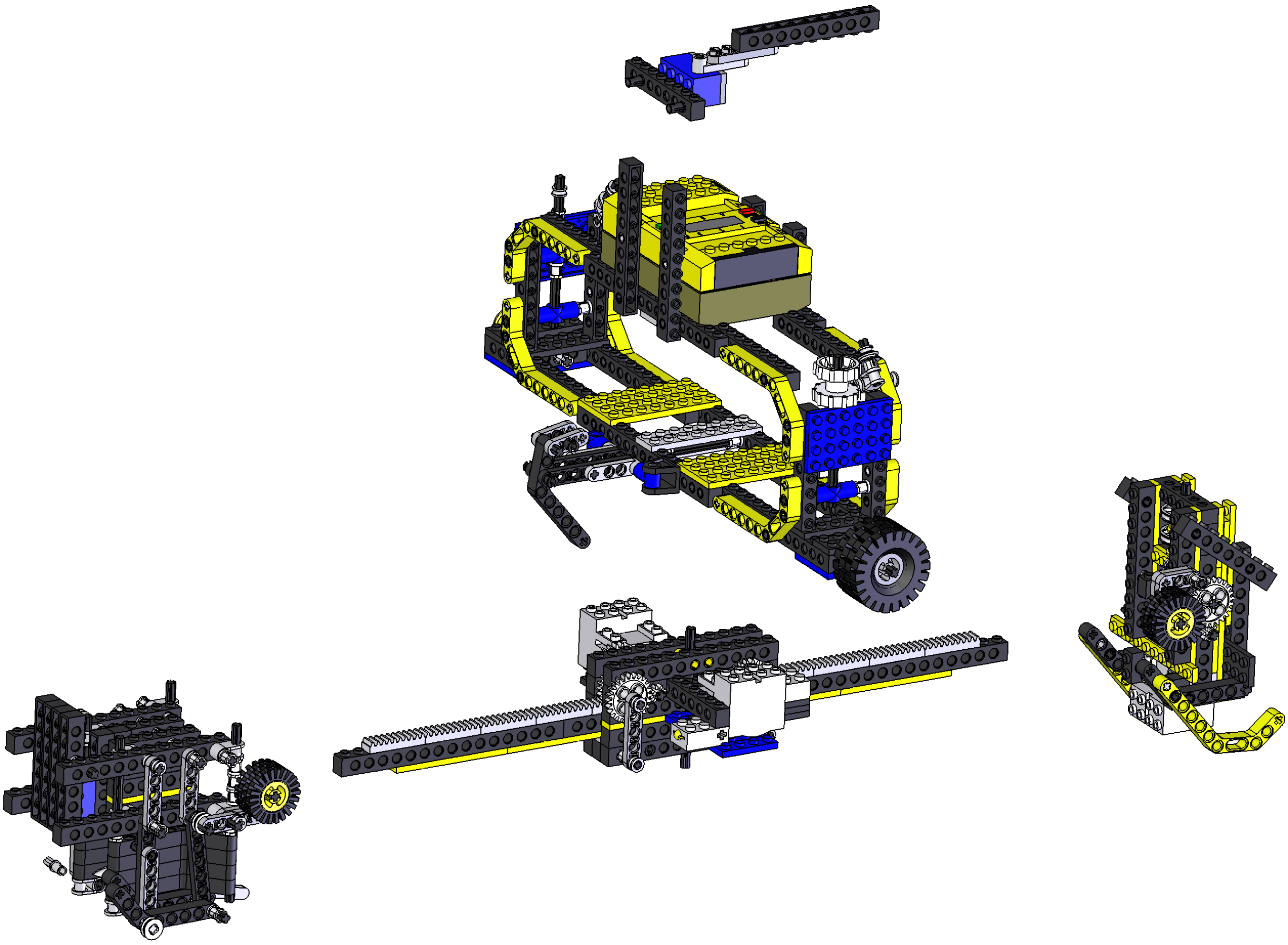

Figure 8: Exploded view of modular robot assemblies

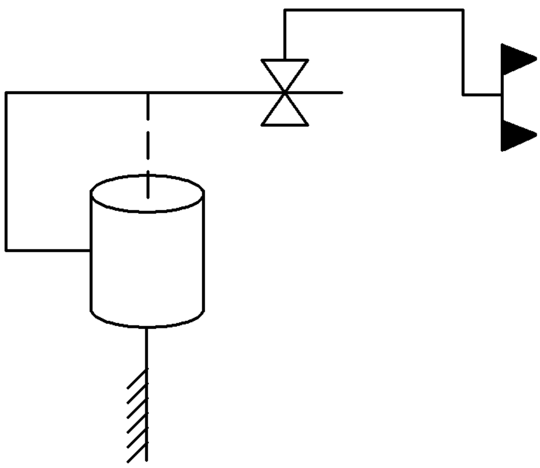

Figure 9: Robot Schematic Diagram

one dimension gyro. Since the Mindstorms RCX computer only has three inputs the two touch sensors and IR sensor were wired in parallel.

The IR sensor is mounted in front of the arm claw so holes can be detected prior to hooking to ensure the claws hook correctly. The touch sensors are mounted to allow detection of arms ends slightly before the arm reaches its full extension. The pan encoder could of been mounted to the axle that the arm rotates about, however to mount the encoder about the pan axle with the same resolution, one would need to mount the controller higher or elsewhere. Furthermore more torque would be required to pan the arm and hence reduce the run time and performance of the pan motor. Since the cable mechanism seems quite reliable, measurement of the cable movement was proven to be the best solution.

Given the limited inputs no weakness in the sensor strategy can be foreseen. Since the pan encoder is used to provide a resolution 0.15 degrees, which is over kill for the arms searching area to find holes. However if more inputs were available it would be ideal to add another encoder to measure the position of the arm extension and more light sensors to increase detection of holes.

5 Program Strategy

The program design chosen was a concurrent multitasking approach for both the ideal algorithm and the algorithm used on competition day.

The ideal algorithm in section 2.1 was implemented with ten tasks, one being the `main()` task. Three of the tasks were implemented as interrupt service routines (ISR): one to read the sensor data and update the program variables that store the current sensor values, one to control motor requests and the final ISR was used to monitor the gyro and take appropriate action.

The final six tasks where dedicated to searching and hooking holes. One of these tasks was the primary `search()` task, while another task (`is_at_hole()`) monitors the IR sensor readings to detect holes. If a hole is found, the `search()` task is stopped and the refining search for the hole begins by starting three tasks: two individually control each arm and pan axis, while the third task monitors the IR value waiting for the reading to indicate the IR sensor is in the middle of the hole. Once the IR sensor is in the middle of the hole, these three tasks are stopped and finally a hook task is started to allow the arms claw to hook into the detected hole.

It should be noted the refining search routine is not trivial. Currently it is implemented by first taking a IR sensor reading, then moving each axis in a random independent direction. A few milliseconds later another IR sensor reading is taken. If the values indicate the robot claw is moving away from the hole, then toggle the direction of the appropriate motor.

Since the touch sensors were wired in parallel with the IR sensor, the software must be programmed to deal with this. The LEGO© touch sensors used are actually pressure sensors giving readings from 0 to 100%. However the IR sensor readings never seem to increase above 90%, so the software double checks the touch/IR sensors readings to see if, after a small delay, are the reading now above or below 90%. If readings are above 90%, the reading must be from touch sensor, if the readings are below 90%, the reading must be from the IR sensor.

Some of the sensors require calibration. The software needs knowledge of the IR readings that indicate if the sensor is in a hole or not. The one dimensional gyro sensor (actually just a encoder) needs initial knowledge of the direction of gravity. And the pan encoder needs to know its zero position. To implement this the software first initializes via going thing a beeping process requiring someone to show the robot when it is in a hole, and when it is not and furthermore know the zero position of its encoders for both the gyro and pan axis.

The algorithm on competition day also used the above ISR's, however was just given simple goto position commands, which will not be discussed in this report since it's not an approach this report will promote nor glorify.

The entire source code can be found for both algorithms in Appendix C.

6 Experimental results

6.1 Mechanical

Throughout the testing phase the design was continually evaluated until a suitable solution was found. While the current design does work, various aspects can be modified to attain better controllability and reliability.

The structural design of the robot was very robust, however certain aspect were identified early as potential weak spots. The flexing of the main arm has been apparent from the early stages. Insufficient LEGO© supplies of particular pieces prohibited the remedy of this problem. As expected over time the member attached to the head of the robot fatigued and broke at connection to the head. Such a problem illustrates the cyclic loading experienced by LEGO© pieces when used on a climbing robot.

The cable system gives the robot good controllability and can add to the structural integrity of the robot. This is not the case however if the cable system is incorrectly strung. On the previous version of the cable system it was found that the cables were attempting to rotate the spindles in opposite directions, this resulted in un expected loads that caused the robot to bend while rotating

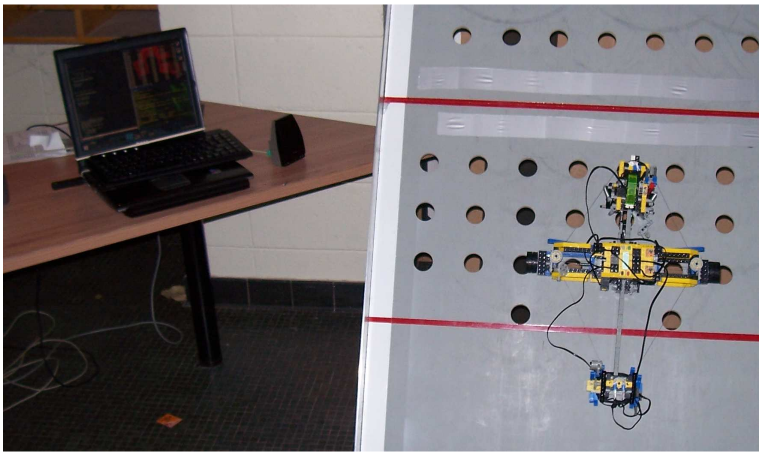

Figure 10: Search Testing

6.2 Software

The development of the software was primarily completed without the need for the robot itself. However to iron out bugs and to test the real world behavior, the robot software was finalised on the competition wall itself.

Figure 10 shows the setup of the search test. Here gaffer tape was simply used to block possible holes that the robot could find and hook into. These tests where very simple to perform and enabled fast development and refinement of the algorithms. It should be noted such tests could be used to train the robot to learn the position of possible holes on a wall, which would be interesting research.

Tests where also done to check the performance, accuracy, resolution and repeatability of the pan encoder, which reads the cable movement. This was done via requesting the robot to repeatably move back and forth from common positions. Through simply marking the real world movement of the head using the setup in Figure 10 it was found the encoder and mechanical design ment the robots panning ability is highly accurate and repeatable using the compeition wall.

While the software that run on competition day was incomplete, the robot still showed very gracefully and reliable climbing ability.

Once the competition was completed the ideal algorithm was completed partially tested. These tests showed the robot to have insect like behaving when searching for a hole, primarly due to the refining hole search routine. Unfortunely the hardware modifications discussed in this report are first required to ensure a more graceful, uninterrupted climb.

It was also hoped more intelligent searching routines could be implemented and tested. For example a bayesian filter could be implemented to learn from previously found holes. Or more sensors could be added to increase the robots knowledge of the climbing environment.

6.3 Final design

A specific goal that was set for the robot was reliability, this is something that was seen as an issue from early on. Many robots at the competition failed merely due to the reliability of the mechanical design, an issue that does not persist with this robot design. Although several mechanical design issues have arisen throughout the testing phase, these were identified and action taken where appropriate. Some future modifications can be found in section 9.

The only major failure that occurred with the design of the robot happened at the early stages of the first competition run on Friday afternoon. The reasoning for this failure was due to software, unfortunately during all the fuse on competition day the robot was programmed with an incorrect software version resulting in the tail encoding not reading correctly and causing the claws mechaism to foal and cause the robot to fall. It should be noted this was also the first recorded fall for the robot, which further demostrates its reliability.

6.4 Competition Day

The software used on compeition day was not complete, however the robot still performed as expected and proved to be a very reliable and graceful hole climber. While the robot was capable of climbing to the programmed location on the wall, the true intelligence was unable to be demostrated through the use of the ideal algorithm outlined in section 2.1. Even so, this again demostrates the robots climbing ability since less software is required to ensure the mechanical system works as required.

Milestone Date	Description
21/8/07	Preliminary design report due
1/9/07	Finish assembly of modular robot design
15/9/07	Working climbing robot completed
21/9/07	End of competition
24/9/07	Final Report submission

Table 1: Project Milestones

7 Project Plan

The project milestone as detail in table 1 were not unfortunately 100% achieved. While the initial modular mechanical design was targeted at being completed completed on the 1st of September, it was found that primarily due to the initial choice of claw design a partial redesign was in order. This delay pushed back the time frame and choked the available programming time. The revised design was completed on about the 8th of September. From here, it took approximately another 1.5 weeks to weed out mechanical design issues and polish the construction, this was done in combination with implementing the software algorithms. The final working climbing robot milestone was completed on competition day, 21st September. However this didn't effect our performance on competition day, only dampen the performance to show its true full potential on competition day.

8 Conclusion

The robot clearly has much potential due to it novel mechanical design. While the software developed is not completed to its full potential. The software already is at a stage that demostrates the robots climbing ablity and intelligence ablity in searching and hooking into holes. The robot also performed as expected on competition day, which reinforces its climbing ability. In conclusion, given the progress in the project timeline and successful mechanical and software design and implemention performed by two members the project is clearly a complete success.

9 Future Work

There are a certain modifications that could potentially benefit the functionality of the robot and allow for a more efficient design. Insufficient LEGO© supplies of particular pieces prohibited the the main arm mechanism to be reinforced so it display some amount of flex, which currently does not appear to greatly restrict its abilities, but a more rigid arm would be ideal.

The head of the robot is particularly large due to the extreme development process. However potential improvement would be to replace the dual claw and guidance system with the base claw, having one claw on the arm and two on the base, or even more more then two spread across the base. This change has the potential in greatly decrease the complexity of the robot and importantly reduce its mass.

The wall used on competition day had ahole pattern at the very top that diverges into a diamond shape. Unfortunately the robots base is to wide to traverse around the edge of the diamond[1]. As a result the wide wheel base collides with the edge of the wall and restrict the climbers motion. This restriction is sufficient enough to block an further motion of the climber. A solution of this problem would be to move the wheels that are located external to the base module onto the inside. While this would marginally decrease the stability of the robot, it would also allow sufficient space for the robot to complete climbing the wall.

As previously discussed in this report there is still work be been done to test various searching algorithms which could also lend to possible research interests.

[1]The robot unfortunately t have great enough reach to trek over the wall section to avoid traverse the edges of the diamond.

A Robot kinematics

The robot kinematic relationships can be simply derived with use of the Denavit Hartenberg parameters. Here θ represents the angle of rotation of the arm relative to the base from the calibrated position and L_1 represents the distance from the base rotating axis to the head of the robot. Figure 11 shows the directions of the coordinate systems for each joint. Note however that the location of the origin of each system are actually the same, the coordinate system for joint 2 was moved up for clarity.

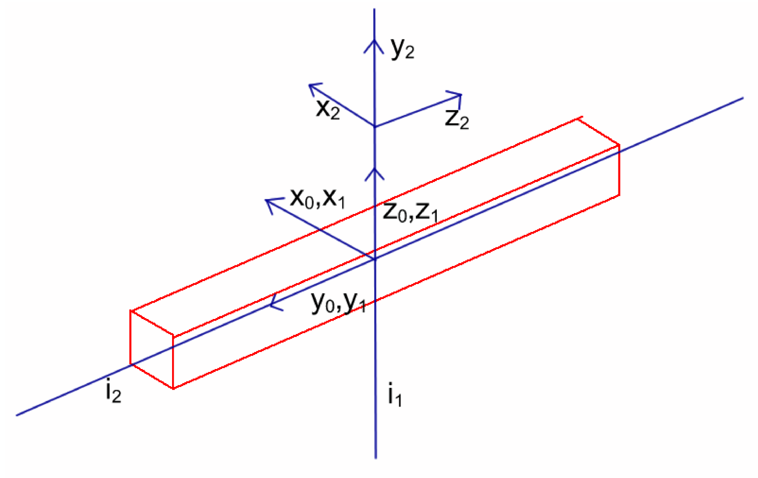

Figure 11: Coordinate systems for Denavit Hartenberg parameters

Figure 11 shows the diagram of the appropriate parameters that can be determined, which are also tabulate below.

i	α_{i-1}	a_{i-1}	d_i	θ_i
0	0	0	0	0
1	0	0	0	θ
3	90	0	$-L_1$	0

Using these parameter the transfer matrix from each joint and the series of joints can be determined.

$$ {}^{0}_{1}T = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1} $$

$$ {}^{1}_{2}T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2} $$

$$ {}^{0}_{2}T = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) & -L_1 sin(\theta) \\ sin(\theta) & 0 & -cos(\theta) & -L_1 cos(\theta) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3} $$

A.1 Frame arm relationship

In order to make use of the above kinematic transformation the value of theta must be determined. This is done by measurements from an encoder that directly counts the motor rotations. In order to estimate the angle of the robot a trigonometric expression was derived.

Given an angle of β degrees that the motor has changed relative to the calibration position the robot position θ is given by

$$\theta = cos^{-1}\left(\frac{W_{0.5}^2 + L_2^2 + \left(W_{0.5}^2 + L_2^2 + \left(\frac{2\pi\alpha}{360}\right)\right)^2}{2W_{0.5}L_2}\right) - 90 \tag{4}$$

where, α is the rotations of the main pulley, r is the radius of the actuating pulley $W_{0.5}$ represents half the width between the base pulleys and L_2 is the distance between the tail and the rotating axis.

B Applied forces

The force that ar generated by the motor is highly significant to the robots structure. Particularly due to the high torque in these circumstance as a result of the high gearing.

B.1 Applied prismatic joint force

Given a series of gear coupling together two 8 to 24 tooth gears a drive ratio of 9 is achieved. The torque determined of the LEGO motor in the preliminary report was calculated to be 20Nmm. This leave a total exerted torque of 180Nmm applied to the driving shaft. The mean gear radius of the cog driving the racks was measured to be 7.6mm, which results in a force of 23.7N. This force is then applied by each of the two motor resulting in total prismatiic joint force of 47.4N.

B.2 Pan torque and cable force

In a similar setup to the prismatic joint actuation the panning mechanism has a 9 times drive ratio. Resulting in 180Nmm of torque to the main drive pulley. This pulley has a diameter of 15mm, which creates an applied force to the fishing line of 24N.

The fishing line used in this application is rated at 6lb. While this is only about 26N, the robot should never see a full tension in the line if operating correctly and is thus well matched to the situation.

C Software