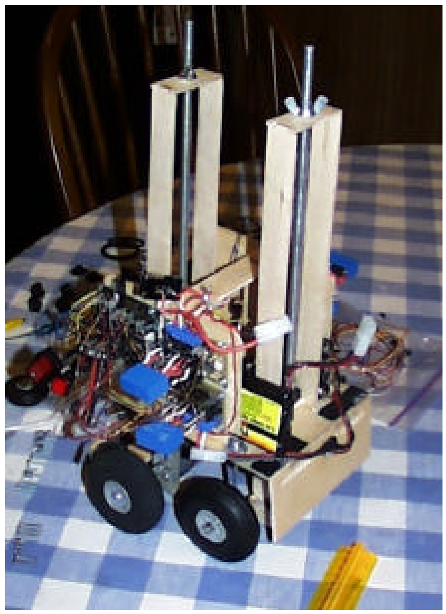
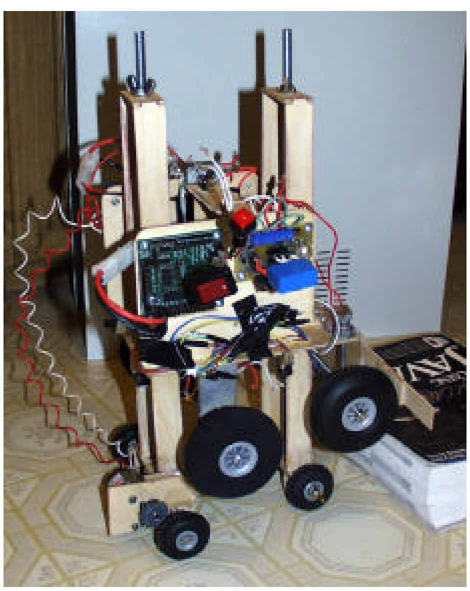
A.S.C.A.

AUTONOMOUS STEP CLIMBING AGENT

Steve Lewis
IMDL Spring 2000
University of Florida





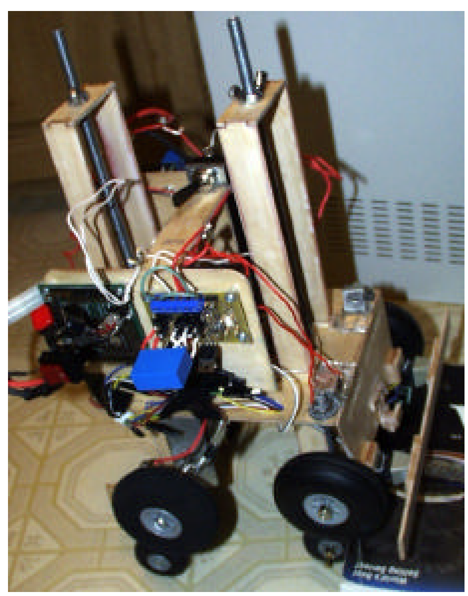


TABLE OF CONTENTS

ABSTRACT

ASCA is intended to explore the intricacy of designing and building an autonomous agent to climb stairs at low monetary cost. Numerous designs seem plausible, yet each requires considerable thought in its implementation. ASCA was implemented with a single platform that uses two posts (one front and one rear). Both post consist of a vertical threaded rod, flanked by two wooden supports. The body rests on two rotating supports, which are also threaded and attached to the rod. These rotating supports are turned by gears (via a DC motor). Both of the posts, which can raise and lower independently, are intended to allow ASCA to climb a step in a simple sequence of six motions. Two types of sensors are used, one for navigation and the other for state detection during the step climbing process. The ASCA design is able to climb a step, albeit slowly, but hardware problems prevent ASCA from doing so autonomously.

EXECUTIVE SUMMARY

The time period for the ASCA project was during the Spring 2000 semester at the University of Florida (beginning Monday, January 10 and final demonstrations presented Monday, April 24, a 15 week period). The first month consisted primarily of conceptual design and construction of the microcontroller board, the MTJPRO11 board (developed by Mekatronix, circa 1997). Visual models were developed in 3D Studio followed by cardboard mockups.

The first critical challenge was designing the vertical actuation process (that is, how ASCA would raise and lower its platform). Various ideas were considered, such as rack and pinions, hydraulics, or chain belts. For simplicity and low cost, the design chosen uses a threaded rod, noting that a rod coupler can be spun to traverse a threaded rod. Therefore, a number of simple mechanical designs allow ASCA to raise and lower using posts (threaded rods with various supports).

Early February[1], various components were ordered and gathered (such as motors, IR sensors, batteries, threaded rods, rod couplers, washers, bolts, nuts). Motor drivers were also built, one for each of the five required motors. With the components in hand, a great deal of time was spent finalizing a platform using AutoCAD[2]. The first cut of the design suffered from lack of fit and insufficient support for the post. A larger second cut was made and the improvements were well worth the cost of the wood used.

A second critical challenge appeared near the end of February. The motor drivers had various problems that took approximately two weeks to resolve[3]. By the end of March, the platform for ASCA was essentially completed. Unfortunately, as the April deadline approached, a more serious problem with the motors and motor drivers was found, which was never fully resolved. The motors caused interference with the microcontroller and IR sensors, even though the motor drivers were designed to prevent such interference. Various methods were attempted to eliminate the noise from the motors. Unfortunately, a distinct solution was not determined. The motor drivers are suspect since other MIL students experienced similar interference problems. Generally, the interference would cause the microcontroller to reset spontaneously, particularly during a direction change.

The cost for ASCA is close to $270 in parts. This does not include the cost of parts provided by the lab (such as wires, solder, headers, resistors, ICs, etc), batteries, development software (ICC11 and AutoCAD LT), nor building tools (drills, etc). The total cost of the project, excluding software and building tools, was approximately $400. For software, ICC11 was provided by the lab, while the student edition of AutoCAD LT 2000 was $130.

[1] During the second week of February my maternal grandfather passed away, therefore I spent the week with my family and attended his funeral in Indiana. May he rest in peace after combating Alzheimer's Disease for so many years.

[2] A student edition of AutoCAD LT 2000 was purchased to work on the design out of lab time, which saved a considerable amount of time. In addition, a greater amount was learned about AutoCAD.

[3] The problems, and solutions, are discussed in a separate document. The motor drivers used are ones developed by Erik Anderson (a former MIL student at UF).

INTRODUCTION

Background

From our perspective, climbing stairs is a trivial task. One might notice that some animals, such as certain dogs, might initially have trouble climbing stairs (yet can be taught or shown how to do so). But animals (including humans) have an enormous flexibility advantage over most robots, plus the ability to quickly shift weight, which is a key part of being able to climb stairs. Thus far, even expensive and well-designed robots are not highly agile. By requiring autonomy, the difficulty of designing a robot to climb stairs is increased even further.

While climbing stairs is difficult (from the perspective of a machine), it is reasonable to ask why climbing stair is even a worth while task for an autonomous agent. One answer is that adding any kind of mobility to a robot is a good thing. Consider a robot designed to clean the floor of an office building at night. If a robot could climb stairs from one floor to the next, it would then not be necessary to have one robot on each floor. Perhaps, however, such a robot could use an elevator, which might be a reasonable solution. This assumes that an elevator is available. In addition, perhaps the cleaning robot would also be required to clean stairs. This, of course, is only one example where an autonomous stair-climbing robot might be useful. Another example might include a robot that carries objects, even other robots, up and down stairs. And being able to do this autonomously is all the better.

Literature

Other similar projects have been attempted in the IMDL course at the University of Florida. The Miracle (Mike Mattress, Fall 1998) was almost successful. The project used two independent platforms, and a chain gear to shift weight between the platforms. Project MACO (Ted Belser, Summer 1999) was apparently modeled after a pair of legs, but on wheels instead of feet. Project Hannibal (Marc Poe, Summer 1999) designed a “six legged spider” that was intended to be able to climb smaller objects. Unfortunately, although it appears that a genuine attempt was made, none of these projects were highly successful. In most cases, it appears that the design should have worked, but the implementation was not properly done (such as problematic electrical components, weak motors, etc).

Objective

The objective of this project is to build a robot that can autonomously climb stairs. Therefore, we should define the parameters of the stairs being climbed. The width is not of great concern, but we should assume that the width is uniform (likewise, we can assume the height and length is uniform – that is, the stairway should not be curved, and each step is the same size). The smallest step to be climbed should be 4” high and 10” long. The largest step to be supported should be not more than 9” high and 24” long.

For a step smaller than 4” high, we could probably construct a sort of car with large wheels to simply drive up the step. A step larger than 7.25” high is unusual, therefore the height maximum of 9” should cover most standard stair steps. The minimum of 10” in length for a step means the robot should not be more than 10” long (practically, this means the robot should be approximately 8” long or less). The maximum length for a step is not really important. However, if the robot moves more than 24” after climbing a step, we can assume a step has been completed and perform some other behavior.

DESIGN

In a previous project (The Miracle), the design consisted of two platforms that could move independently and a chain drive that shifted a weight to the front and back. The front platform was raised and rested on the next step and weight was shifted to the front, where then the rear platform could be raised. The two platforms then moved forward and weight was shifted back to the rear. It seems fairly evident that this design should work. Unfortunately, limitations on time and money were among the reasons why the Miracle was not successful.

For project ASCA, it was decided that shifting weight is a difficult task for a robot. One reason humans must shift weight to even walk is because they are very tall (with respect to their height and width). Suppose an object was short and had uniform weight, such as a standard 12” ruler. It can be demonstrated that a ruler will not fall off the edge of a desk, until slightly more than 50% of the ruler is extended off the table. In general, part of physics studies such problems when dealing with center of gravity. But instead of relying on vigorous calculations, project ASCA is relying on intuition of the following observed principle: if most of the weight of an object is on a flat surface, some portion of that object may be extended beyond the edge of said surface.

Climbing Steps

Project ASCA uses a single platform that will attempt to have a semi-uniform weight. Weight adjustments can be added once the platform is built. In order to lift the platform to the next step, and to move the platform forward, two independent posts have been employed. These posts will operate as indicated in the diagram below:

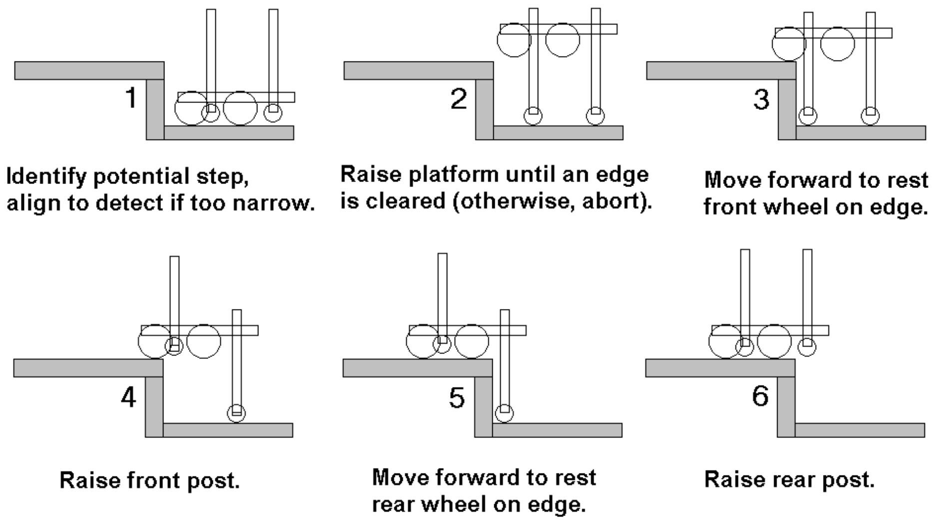


Figure 1

There are a number of issues that come to mind after reviewing the diagram above. First, between step 1 and 2, how is the body to be raised up? Furthermore, how is the front and rear post to move independently? Between step 2 and 3, how is the entire robot driven forward (likewise is steps 3 and 4)? Lastly, between steps 5 and 6, it is imperative that the robot does not fall backwards while lifting the rear post. However, the diagram above tends to over emphasize these problems. When implementing this design, we simply must ensure that by step 6, more than 50% of the robot's weight is already on the next step. In step 2, we assume that wheels can be placed at the bottom of the post (with a motor drive strong enough to drive the robot forward). There are a number of options in how to raise and lower the posts independently. The actual method used in this project is discussed below in the Implementation section.

Block Diagram

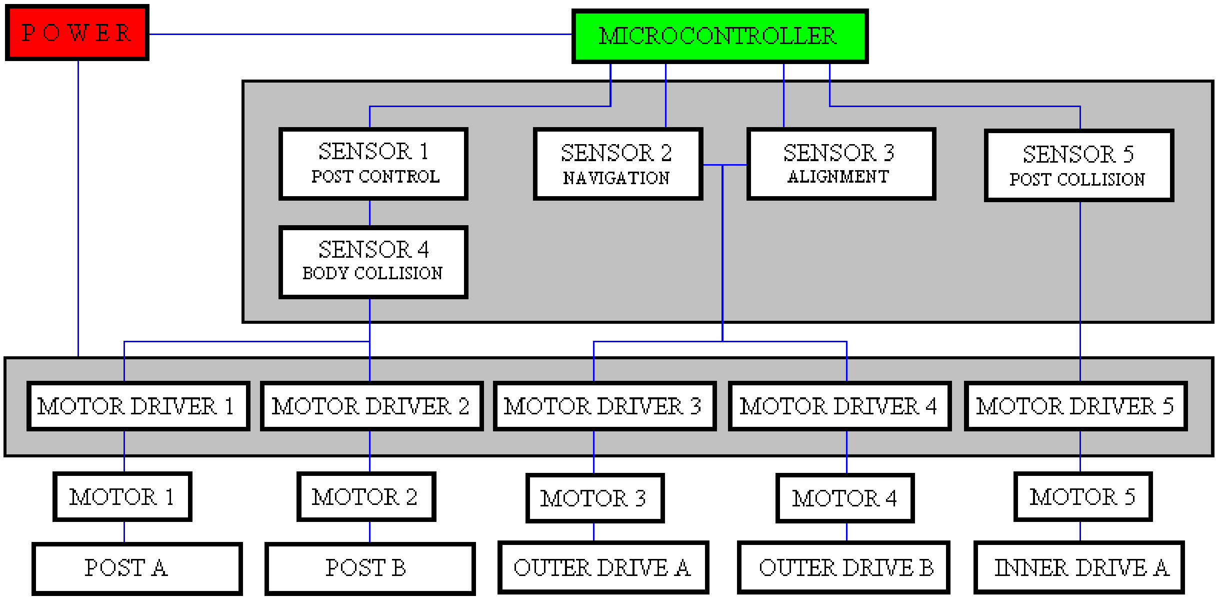


Figure 2

Above is a diagram showing the block layout of ASCA. First, power is provided to the microcontroller and the motor drivers from two separate power supplies. Next, the microcontroller initially gets input from sensor block 2 (navigation), which are used to find a wall and avoid obstacles using motors 3 and 4 (which drive the robot left and right). Once sensors 2 have found a potential step, sensor block 3 will help align the robot to the surface (to avoid attempting to climb at an angle). After the robot is properly oriented towards a surface, motors 1 and 2 will allow the robot to use the post to raise the body up. Post A designates the front post and Post B is the rear post. Sensor block 1 determines when the posts have reached their minimum and maximum heights. While the body is being lifted or lowered, sensor block 4 helps detect obstacles (such as a step with an overhang or the ground when the body is being lowered onto the next step).

Once the robot body has been raised up, it can move forward a short distance. Sensor 5 will indicate when the short distance has been traveled. At this stage, we must ensure that at least part of the robot is resting on the next step, then motor 1 can raise Post A. Again, the robot moves forward a short distance. Sensor 5 will also indicate when the maximum distance has been reached. At this point, we must assume more than 50% of the robots weight is resting upon the next step, then begin to raise the rear post (post B). Note that the Outer Drive drives the robot while not climbing stairs. When the robot body is lifted up, motor 5 drives the robot forward using Inner Drive A.

IMPLEMENTATION

Platform (body) Layout

Based on the block diagram specified in the Design section above, the following diagram is approximately how the robot board will be organized. Given the size of the components (motors, motor drivers, etc), a board size of 5”x7” has been chosen.

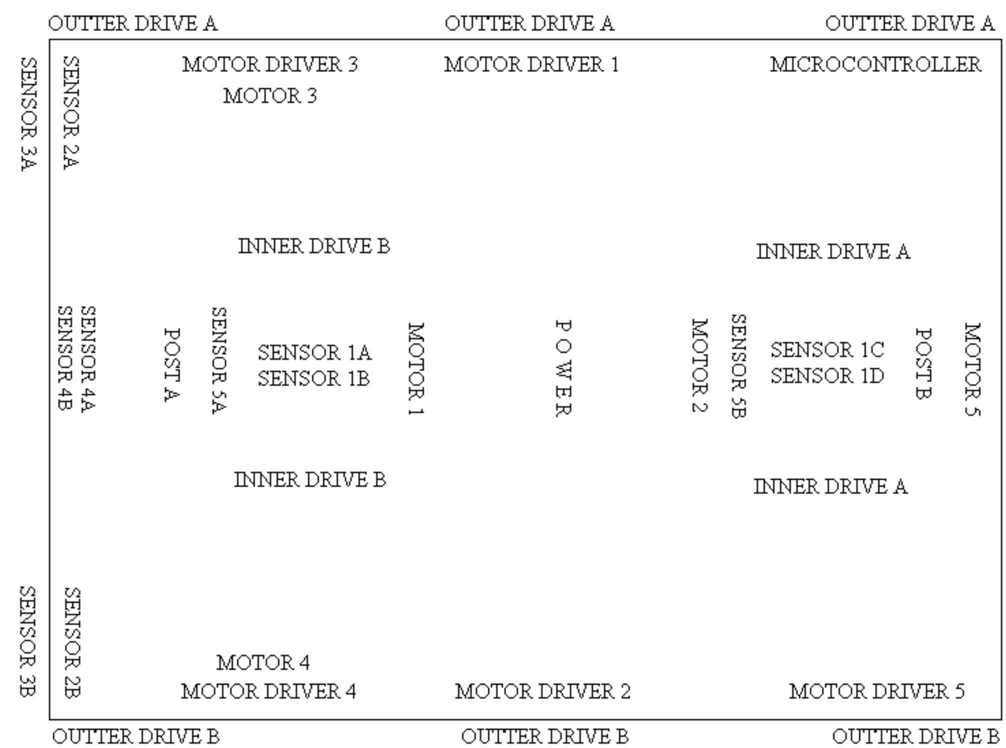


Figure 3

Two batteries will supply power, one for the motors and one for the electronics. Since the motors will likely consume a great deal of current, a large battery pack is being used (such as the 9.6v 8-packs used in R/C cars, which generally have 1000 to 1700 MAh). A less potent battery could be used for the electronics, since the microcontroller and motor drivers would draw less current then the motors. Any source over 6v should be sufficient since the MTJPRO11 has a voltage regulator.

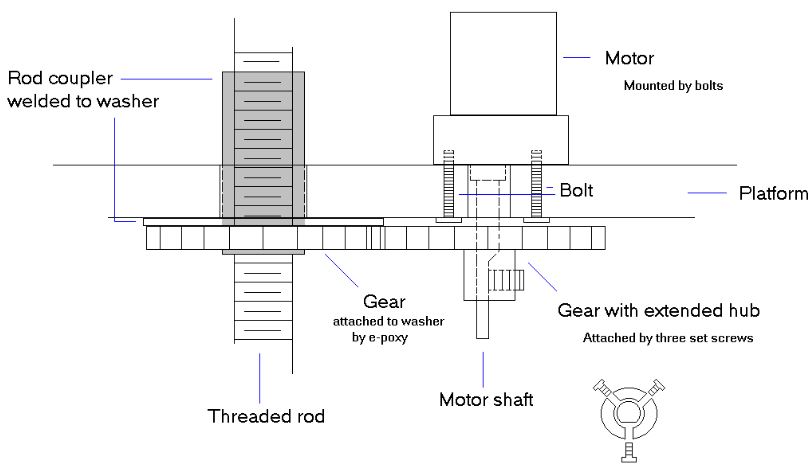
The motor drivers are 5A versions of the schematic provided on the IMDL web page. For the microcontroller, the MTJPRO11 board is being used. This board has a small footprint and can support many sensors.

In addition, 32K SRAM provides ample space for debugging purposes. The five output compares (PA3-PA7) are being used to provide pulse width modulation (to control the motor speed).

All the sensors will be bump sensors, except sensors 2A and 2B, which will be a combination of hacked Sharp IR emitters and IR receivers. Ten bump sensors will be used to implement sensor blocks 1, 3, 4, and 5. The following sections discuss the mechanical design of various parts of ASCA.

Post Actuation

The front and rear posts allow ASCA to move vertically. While the idea is simple, their implementation is not. Both post consists of a threaded rod with guide supports. A threaded rod is used for its effective mechanical benefit. Guide supports help prevent the rod from tilting. The next diagram shows how the rod is used to provide vertical actuation:



Deciding on the rod to use is an important issue. For ASCA, two 5/16”-18 rods are used (both approximately 13” in length). As the diagram suggests, the rods themselves are stationary (that is, they are not meant to spin). Instead, an somewhat opposite approach is taken: a rod coupler is spun (via a gear), while the robot platform rest on top of a washer attached to the rod coupler.

In ASCA, a hex shaped rod coupler was machined into a rounded coupler (using a lathe). Since the coupler is placed through the platform, the round shape reduces friction and allows for a tighter fit. A metal washer was then arc welded to the coupler (with the help of a skilled welder). If necessary, graphite or a ceramic washer can then be placed on top of the welded washer, to reduce the friction between the washer and the robot platform. The driving gear was attached to the welded washer by JB Weld (the gear used is Jameco part 162085 – a 30 tooth plastic gear).

A motor is used to drive the gear. Mounting the motor depends on what motor is being used. For ASCA, the same motor used in the rear wheels (below) is used for post actuation (Jameco part 162190). They are mounted using short bolts through the platform into the supplied threaded mounting holes in the head of the motor (the bolts were obtained at Rice Hardware, local to Gainesville). However, it can be difficult to determine (precisely) where the motor is to be mounted. Moreover, to drive the gear attached to the coupler, a gear must be attached to the motor shaft, and together the gears must mesh precisely. This is a difficult thing to measure by hand, therefore some guessing may be required. For this project, a test fit was first done on a spare cut out of the robot platform. Narrow slits were drilled into the body, allowing space for the bolts to go through the platform and screw into the motor mounts. When a good cut was determined, the same slits were applied to the final build of the robot. To reinforce the bolts, washers were used on the lower side of the platform.

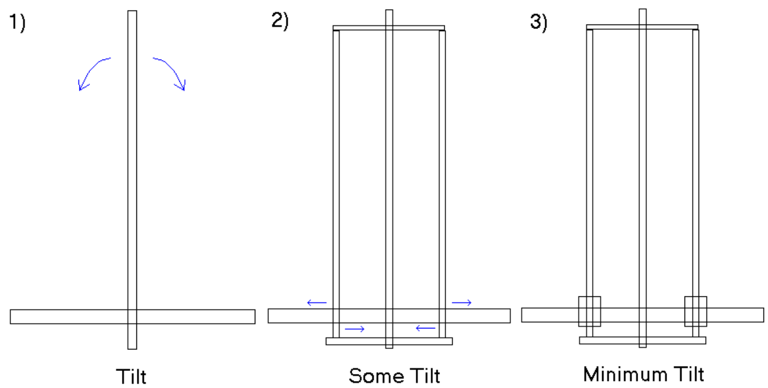
The slits were necessary, to allow the motor position to be adjusted for the attached gear. Finding a suitable gear turned out to be a challenge. There are a number of machine shops that will manufacture custom gears, undoubtedly at great cost (Ross Gear offers a free catalog on a great selection of pre-designed gears, a number of other gear-related companies can be found on the Internet with online catalogs).

For ASCA, a set of suitable gears was provided by a machine shop located on UF campus (Elise Machine Shop). These gears were somewhat larger than the Jameco gears and made of a stronger material. The gears have a short bore, in which threaded holes could be placed. In both gears (for the front and rear), three threaded holes were made (each 120 degrees apart, as shown in the diagram above). By using only one threaded hole, the gear would become unbalanced after the setscrew was applied (the setscrews attached the gear to the motor shaft). Using three setscrews allowed for adjustment, providing a more balanced gear. As a subtle detail, the center hole of the gear was too larger for the shaft of the motor. To compensate, a piece of rubber tubing was placed inside the center hole of the gear. The rubber tubing provided a very tight fitting between the gear in the shaft.

The motors used to drive the post gears (Jameco part 162190), have the following specifications: 4.5-12V, 65mA, 225rpm, 700g/cm, 30:1 ratio. High-speed motors tend to have very low torque. This motor was chosen for a number of reasons, with the most practical being its physical size, cost, and low current. Unfortunately, there is a large amount of friction in the post supports. Measures were taken to reduce friction, such as the use of graphite and grease. Even so, larger motors with even greater torque probably would have been a better choice. The 225rpm speed is tolerable, but much less would make the step climbing process unbearably slow.

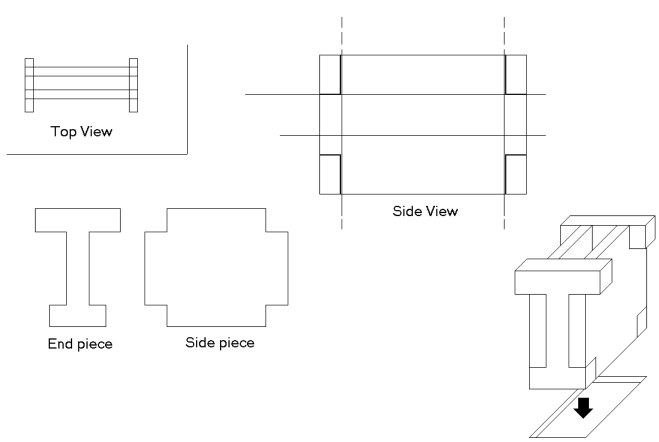
Post Guide Supports

As already mentioned, independent front and rear posts are used to allow ASCA to raise and lower its platform. As discussed above, a threaded rod is used to support such movement. However, the following diagram shows some issues with using a threaded rod:



In drawing (1), the rod by itself will obviously tilt if the rod extends too far above the platform. In the center drawing, the tilt can be reduced by adding guide supports, which might also raise and lower through the platform. Like the rod, these guides can also tilt excessively. To prevent this tilting, some sort of support should be placed at the base of the guides (as shown in the right most drawing above). Such a support could be placed on the rod – in the case of ASCA, a long rod coupler is used.

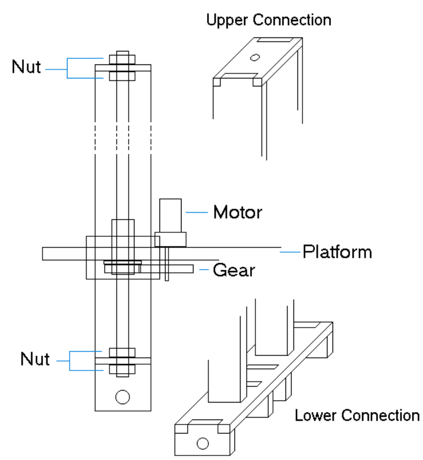
The design of the guide supports is important, since any tilt can easily cause the robot to fall during certain stages of climbing a step. Below is a diagram showing the guide support design used for ASCA:



As shown above, the guide support is placed into an appropriately sized hole in the robot platform. The T-shape end piece allows for the structure to be glued or otherwise mounted to the platform easily and consistently. For ASCA, a great deal of sanding was necessary to allow for the post supports to easily slide between the guide supports. Graphite or a suitable wax could be placed inside the guide support, to allow the post supports to raise and lower with very little friction.

Post Connection

Two post supports are placed at the front and two in the rear of the robot. These post supports must be connected together at the top and bottom. Below is a diagram showing how this was done for ASCA:

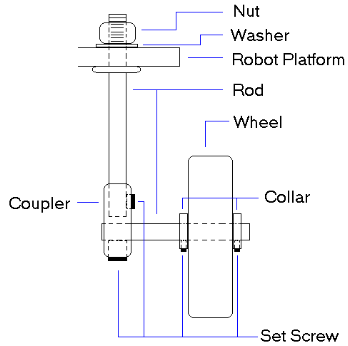


Consider the upper connection. The hole near the middle is where the top of the threaded rod will be placed. Nuts are tightened at the top and bottom of the upper connection for a tight fit. It is best not to place anything on the upper connection – particularly anything that will require a wire connection (such as a sensor). The front and rear post must be able to raise and lower freely from one another.

The lower connection is more involved, since an axle must be mounted. A hole should be placed near the center of the lower connection, for the same reason as the upper connection (to support the threaded rod). Four squares extend below the lower connection, with each square having a hole near the middle. These holes are designed to support the inner drive, discussed below.

Front Wheels

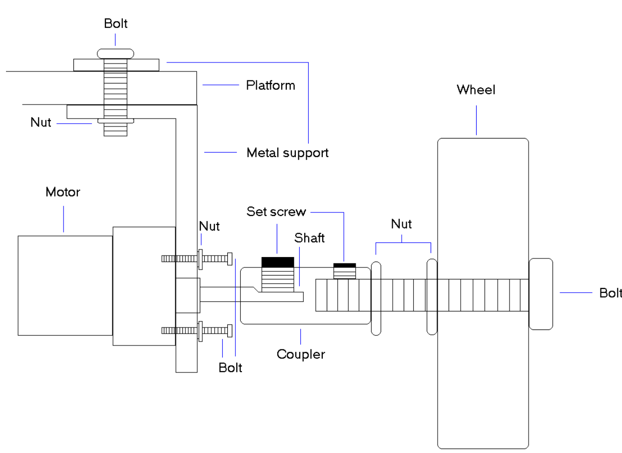
The front wheels are not motor driven, they are designed to spin freely. Their purpose is to provide balance to the robot. These wheels have been attached to the left and right side of the robot as follows:

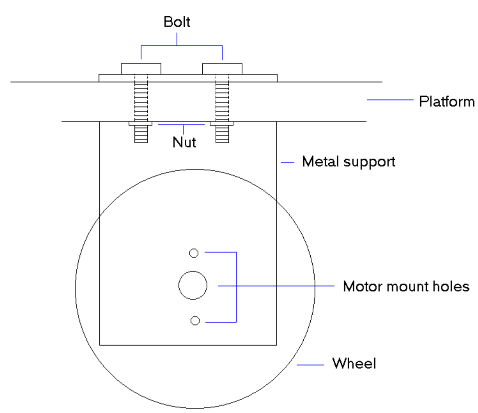


Note that the wheel is some distance below the robot platform. The coupler and horizontal rod for ASCA are manufactured by DU-BRO (cat. No. 614), and are normally used in model airplanes. The wheels are DU-BRO cat. No. 300T (3" shock absorbing wheels with rubber tires). The collars are DU-BRO cat. no. 598 Dura-Collars (brass plated for less friction). They are attached, by setscrews, to hold the wheels in place (loose enough to allow free spinning, but tight enough to reduce lateral play in the wheels. The vertical rod is DU-BRO cat. no. 248, and effectively allows the wheel to be attached to the platform.

Rear Wheels

The rear wheels are what drive ASCA when not in the process of climbing stairs. The distance between the front and rear wheels is approximately equal the distance across the platform (left to right). Therefore, it was decided that a steering mechanism was not necessary. To turn, the robot simply enables the appropriate left or right rear wheel. The following two diagrams show how the rear wheels and motors were attached (front and side view respectively):



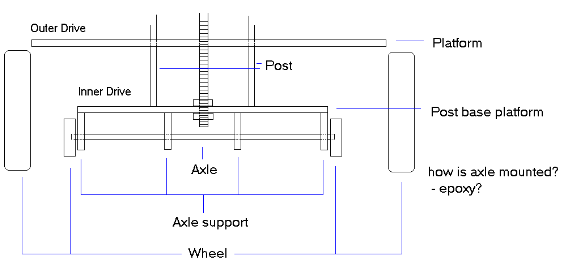


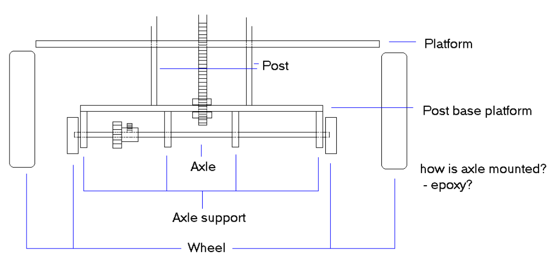
A piece of sheet metal was formed into the inverted L metal support shown above. The shorter end of the L piece (metal support) is attached to the robot platform. A small piece of the sheet metal was cut into a rectangular square, then used as a washer above the platform. The longer end of the L piece is where the motor is mounted to drive the rear wheel (separate left and right motors are used, to allow the robot to steer). How the motor is attached depends on what motor is used. For ASCA, the motor used is Jameco part 162190 (GH12-1324Y225, which at 12 volts gives 225 rpm and a torque of 700g/cm). These motors have small mounting holes already pre-made (3-48 thread size bolts were found to fit). Holes had to be drilled into the sheet metal to match the support at the base of the motor shaft, as well as allow for a bolt to be placed into the motor mounting holes.

A coupler was used to attach the wheels to the motor shaft. In fact, this is the same coupler used in attaching the front wheels (among other parts, two are included in DU-BRO cat. no. 614). The right most setscrew in the above diagram is not actually present. While a setscrew would be helpful, the coupler used did not support a setscrew at that position (a hole is present, but it is not threaded). However, it is possible to attach a bolt into the left side (as drawn). To secure the bolt, two nuts are used (one tightened towards the coupler and the other tightened towards the wheel). The wheel seems well attached, with no significant play. If this was not sufficient, some kind of epoxy or glue could be used.

Inner Drive

The inner drive is used to drive the robot while it is raised up. The design of the front and rear inner drives is as follows:

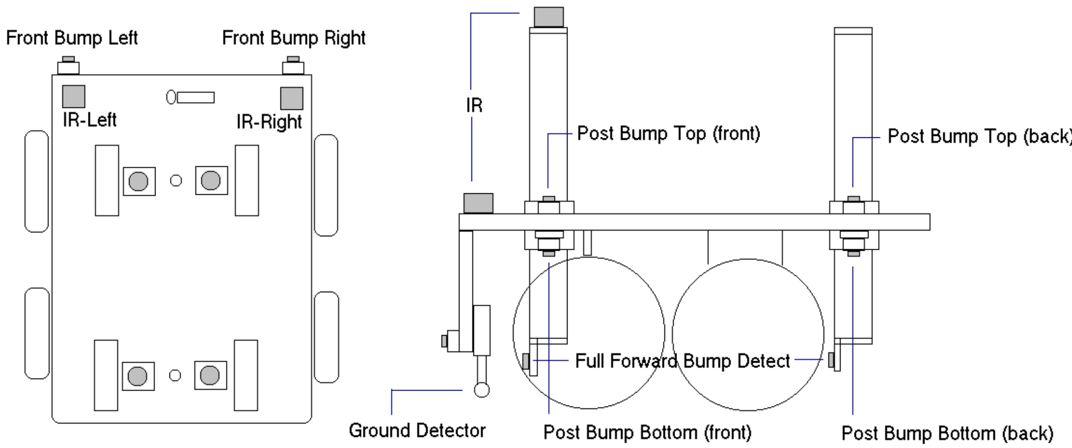




1.75” DU-BRO wheels were used as the drive wheels, with a small copper rod as the axle. Mounting the motor to drive the rear axle was not an easy task. A belt drive system was considered, but a suitable belt could not be found in time. Instead, small gears from a K’Nex toy was used. The actual motor was mounted by bolting it to a wood extension attached to bottom of the rear post.

SENSORS

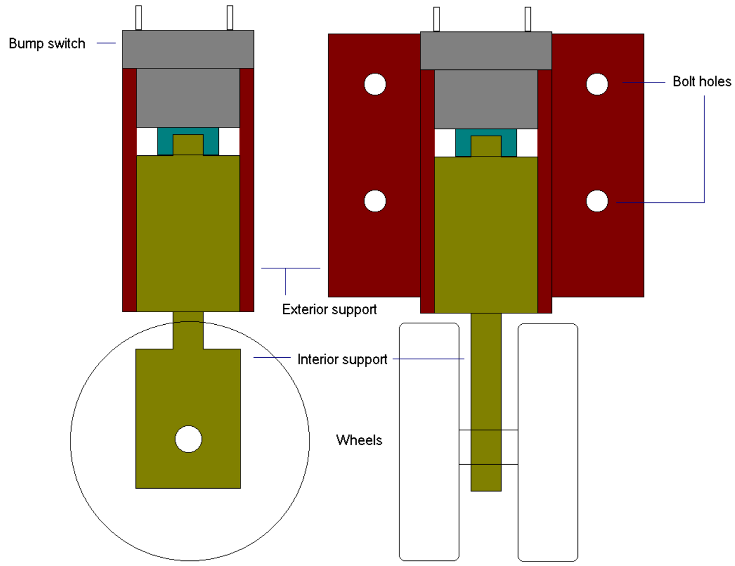
ASCA requires a number of sensors – three IR for general navigation and stair detection, eight bump switches for various detection, and a special ground detector. The following diagram outlines where the sensors are located:



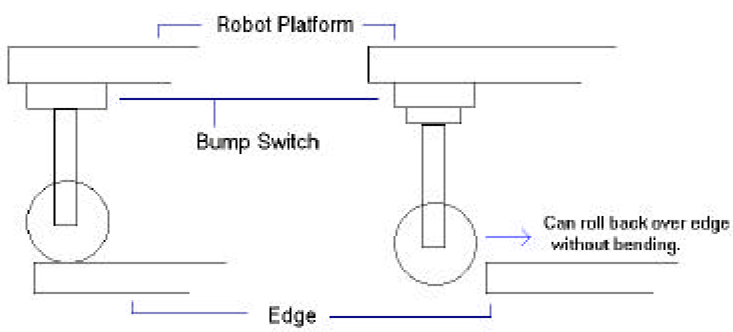
The Sharp Can and generic IR emitter has been used as the IR sensors. These sensors tended to have a range of approximately 18". The various bump switches are needed at different times during the step climbing process.

After IR has detected a potential step, the front left and right bump switches are used to align the robot with the step surface. The Post Bump Top switches determine when the platform is touching the top of the respective front and back post. Likewise, the Post Bump Bottom switches determine when the platform is touching the bottom of the respective front and back post. When the platform is raised up the posts, the Full Forward Bump switches determine when the inner drive has driven the robot fully forward (that is, the respective front and back post have hit the step).

To avoid falling over an edge, and to potentially support climbing down stairs, a ground detector is necessary. There are various ways to detect when the robot has touching the ground. It was decided that a switch would be the most accurate. The following switch design I used:



The switch would normally be closed, pressed in by the wheels being set on the surface (ground). When an edge is reached, a spring in the switch would extend the wheels out, with the appropriate signal being sent to the microprocessor. The wheels below the switch allow for the robot to reverse back across the edge, without damaging the switch. This switch is essential for when the robot has half way up a step (to determine when the robot wheels are rested on the next step). Below is a diagram showing how the bump switch would be used to detect the ground)



CONCLUSION

Problems

ASCA can be shown to work with manual user controls. The primary problem is noise generated by the motors. This noise has a tendency to cause the MTJPRO11 board to reset, IR sensors to detect abnormal values, and/or corrupt SRAM. To prevent this, several ideas were tried: placing a capacitor across the motor terminals, wrapping the motors in aluminum foil and grounding the foil, placing a tri-state buffer between the signals between the microcontroller and the motor drivers, increasing the distance between the electronic components and the motors, among other things. The tri-state buffer did seem to eliminate the problem of resetting the board. One apparent flaw with the motor driver is that the direction should not be changed while the motor is turning at any speed (this will invariably cause the microcontroller to reset). Unfortunately, any of these problems tend to occur spontaneously. Therefore, it is possible that problems were caused by excessive wiring, or improperly shielded wiring. As should be noted, the motor drivers do appear to work well in isolated conditions.

The most significant result of this project, aside from much personal experience gained, is the development of motor driver software. Output compare interrupts on PA3-PA7 are used to provide pulse width modulation. While the code developed is not highly sophisticated, it can be used to drive five motors nonetheless.

Cost

The final ASCA robot uses about $270 worth of parts. However, the total cost for the development of ASCA is approximately $400. Most of the cost is in motors, motor drivers, the microcontroller, sensors, and wheels. Other cost include batteries, construction parts (bolts, rods, couplers, nuts, fasteners, gears), wire, and switches. The excess project cost comes from purchased parts that end up not working and replacement parts (such as motor drivers and sensors).

A great deal of time was spent on designing, building, and finding necessary parts. I would estimate that IMDL requires at least 15 hours a week in addition to lectures.

Future Work

It is hoped that future students will attempt to build ASCA-like step climbing robots. A goal should be to build a small-scale robot, similar in size to the TJ-Pro, that can climb standard size steps and cost under $300 in parts. The design for the post actuation may also apply to other projects. Though instead of using a threaded rod, some sort of hydraulic might be more suitable (in that it would be faster and more stable). Hydraulics were not used for ASCA due to factors such as cost and weight.

Using small motor drivers would also be good, such as the solid state drivers from Allegro. I was advised that solid state motor drivers tend to break easily (such as overheating by excessive {bouncing} change in direction). The use if servos would also be a good experiment, as they seem to generate very little noise. Hacked servos were not used for ASCA, since it was feared that they would not have sufficient torque. However, the torque required is difficult to measure since it depends on the amount of friction (among other things). The Jameco motors, with 700g/cm or torque, may simply be underpowered for the job. However, the inner drive motor with 3000g/cm of torque (and approximately 115rpm at 12v) seems well suited for its purpose.

Final Words

While brainstorming a project idea at the beginning of the semester, I consider numerous ideas. I wanted a project that was not overly ambitious and required attention to mechanical details. In addition, I wanted a small scaled robot that would be easy to carry. Given my limited experience in building anything, a step climbing robot seemed to fit the criteria well.

The many minute details involved in building the design made the project interesting the entire semester. I've also gained an appreciation for good craftsmanship and the work that goes into developing an electronic product. While ASCA did not use any particularly special sensors, I've inherently learned about various sensors from other projects (such as the digital compass, sonar, UV detector, etc).

I have applied much of what I learned from Digital Logic (EEL3701) and Microprocessors (EEL4744), two of the few EEL courses I took. However, I realize there are many details to consider that are not mentioned in class, and can only be learned through real-world experience. As a Computer Science major, I know a lot about object-oriented code and software design (it was hard to convince myself that a GOTO statement might be the best solution). But I've found that procedural code still has its place and I've been reminded about the benefits of assembly level programming.

My biggest problem was noise from the motors and not knowing how to handle the noise. But I must conclude that this is a very real and serious problem in industry, as some MIL students had similar problems and had yet to find a solution. However, it was my mistake by not testing and dealing with the problem earlier. I tested each motor driver individually, but not true to how they would be used. More vigorous testing earlier might have given me time to find alliterative motor drivers.

ASCA did cost about what I expected it to, but it required more time than I had anticipated. Even finding suitable gears and bolts became a chore at times. I believe IMDL was worth the time and expense, and I hope that others can also learn from this project in the future.