

Appendix B

Primary Sidewinder Development Members List

(Hours are cumulative over two year development)

Name	Year	Department	Hours
John Bates	Freshman	Electrical and Computer Engineering	100
Gabe Brisson	Sophomore	Electrical and Computer Engineering	250
Stephen Ives	Freshman	Electrical and Computer Engineering	100
Rich LaBarca	Sophomore	Computer Science	100
Ryan Miller	Senior	Electrical and Computer Engineering	650
Jorgen Pedersen	Graduated	Electrical and Computer Engineering	500
Mark Sibenac	Junior	Electrical and Computer Engineering	600
Cecelia Shepard	Junior	Electrical and Computer Engineering	100
Nick Validis	Junior	Computer Science	700

Appendix A

Budget

Description	Supplier	Price	Notes
Pneumatics			
10 pneumatic cylinders	ARO	\$200.00	Donated
8 valves	ARO	\$200.00	Donated
Misc connectors	H.B. Parke	\$80.00	
20 ox. CO2 tank	Urban Assault Paint Ball	\$90.00	
High Pressure Regulator	Air Power	\$100.00	Donated
Regulator	Wille Company	\$100.00	Donated
Check Valves	Wille Company	\$27.00	
Electrical System			
24Vdc motor	Servo Systems	\$500.00	
Power Amplifier	Already Owned	\$700.00	
VME backplane	Field Robotics Center	\$150.00	
68030 VME processor	O+R	\$850.00	
Ethernet board	O+R	\$200.00	
DCX-MV100 motion controller	Already Owned	\$800.00	
VMIO10 I/O board	Already Owned	\$450.00	
Misc Electronics	Various	\$150.00	
4 8Ah 12V lead acid batteries	Battery Works	\$150.00	
24V Wall power supply	Keith's Dad	\$350.00	Donated
Mechanism			
Sprockets and Chain	Browning Inc.	\$80.00	
Idler	Browning Inc.	\$50.00	
10 Angular contact bearings	SKF	\$300.00	Donated
Stock materials for frame	Mech E and FRC	\$300.00	Stolen
Welding	Henry (our buddy)	\$90.00	
Threading arm shaft	Pittsburgh Pro Bike Shop	\$50.00	
10 radial bearing for feet	Bearings Inc.	\$65.00	

FIGURE 4.01: Simulator and Line Tool

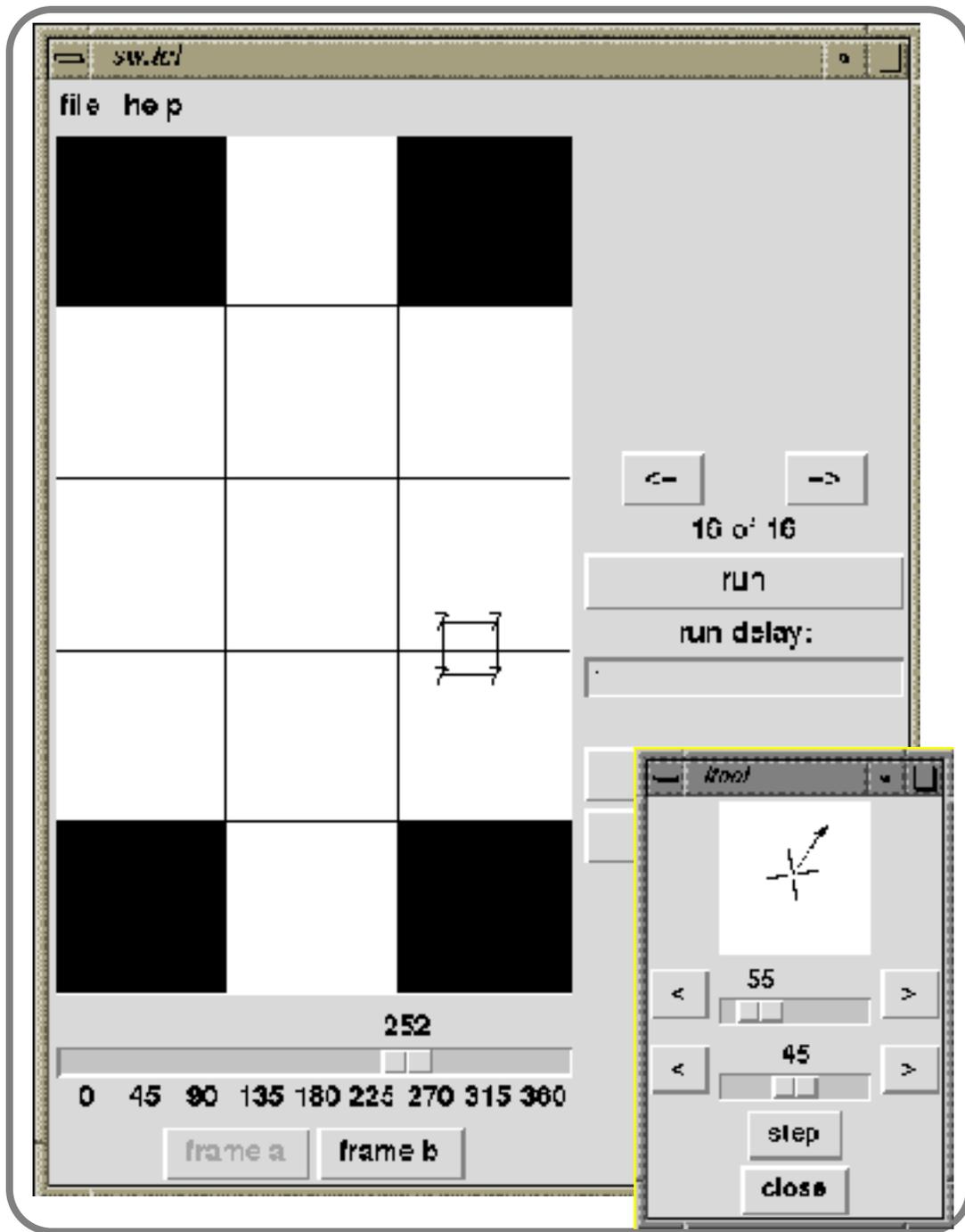
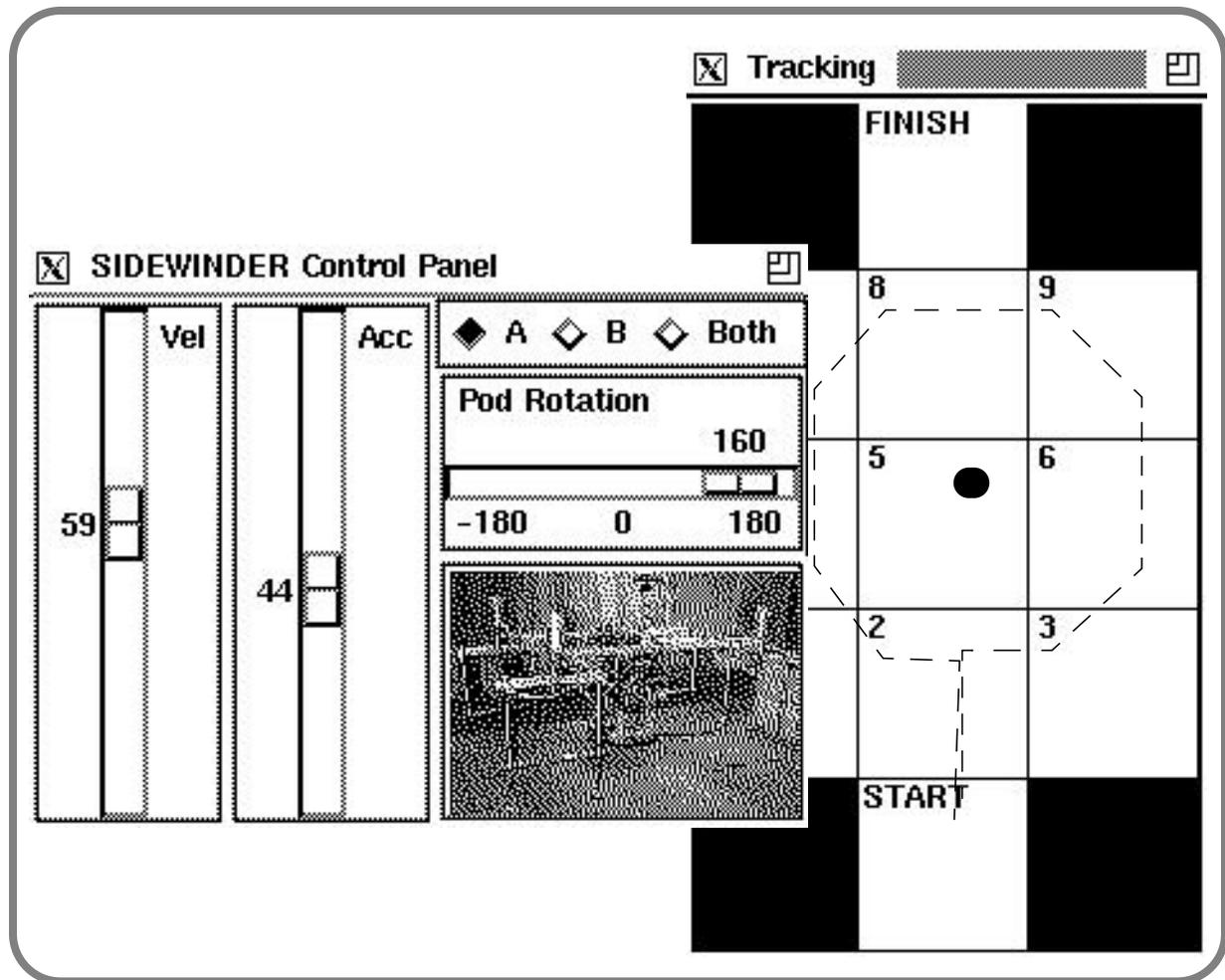


FIGURE 4.0: GUI Interface



Another method we have developed for controlling the robot is the generation of movement sequences using a simulator (figure 4.01). This simulator, written in TCL/Tk, provides a top-down view of the Decathlon playing grid with Sidewinder superimposed on top of it. Using its simple interface, which includes direct leg angle control as well as trajectory-based control, we are able to map out a course for the robot. This course can then be saved and interpreted by the robot. This allows excellent planning for events and simplifies coding to a great extent.

Offboard, a teleoperation module can be executed. This is the communications core underlying our GUI, and also used to send commands to individual modules for testing and development.

To make software development more manageable, we defined a “virtual interface” to the machine. This consists of a predefined set of functions which control the robot at the actuator level. By keeping all hardware dependent code below this interface, we are able to simply link in one of several implementation of the interface. One is the actual robot, where each command causes the actual motion. Another is a diagnostic stub which simply allows early code testing without needing the robot hardware. The last interface is to a graphical visualization of the machine. This simulation runs in TGRIP, a robot visualization environment on SGI workstations.

The software architecture we have designed for Sidewinder II makes use of the concept of a server to pass both data and control. The server makes use of TCP/IP socket connections, within IPT. With these sockets, two processes could actually be running on two different machines or the same one. Since we had a gui running on a sun workstation over the network, the sockets provided the means for communications between the robot and the gui. This implies that a user could control the robot from anywhere in the world if he could connect to the internet. The gui provided a complete interface to the robot being able to control the robot from a low level servo position change to a high level walk command.

4.3 Graphical User Interfaces and Simulator

As was mentioned before, we have developed three new control mechanisms for Sidewinder. Two of which are graphical user interfaces (GUI's). In our attempt to create a fully-functional yet attractive and easy to use interface, we have discovered that the TCL/Tk graphical scripting language is the best language to use. TCL/Tk provides a fast, powerful way to create the GUI's we want and the ease of extensibility needed in an ever-growing robotic system.

To integrate the TCL/Tk code into the rest of the software architecture, we need to create a link between it and C. We accomplish this by using a robust library of C functions and TCL code which links the two languages via stdin/stdout redirection. The immediate layer of C underneath the GUI consists of IPT commands to allow the GUI to communicate with VxWorks via the ethernet link. Thus, the GUI has access to all functions and variables which are available on the 68030CPU. The result is a seamless interface to the robot which allows querying as well as commanding. One GUI is implemented using this technique, shown in figure 4.0.

The other GUI is a simpler, more limited form of interface which we call a “shell-interfacing” GUI. It links itself with the robot via a spawned telnet session through which it sends command strings. This GUI is unable to obtain information from the robot but can blindly execute functions and set variables on the 68030CPU. Simple communication such as this is extremely useful in the competition as well as for debugging and simple control.

4.0 Software Design

4.1 Development Environment

To provide a more usable and powerful development environment, we have decided to move away from PC software development and do all code development on SUN workstations. X and unix provide significantly easier development tools, and make working with the systems much easier.

We are also using the vxWorks real time operating system (version 5.2) on board Sidewinder II. VxWorks provides a very helpful front end interface which facilitates quicker debugging, as well as simple ways to interact with the robot's code. This multi-threading operating system affords us the ability to develop concurrent modules, and work with a more advanced software architecture than previously.

Probably the most ambitious software goal we are tackling for this competition is the creation and easy extensibility of a graphical user interface system for Sidewinder. Three new software subsystems are the result of this goal, two of which are being used in the competition. These three subsystems consist of a fully integrated GUI, a stand-alone "shell interfacing" GUI, and a fully-featured simulator which can generate command files. All three systems are based in the TCL/Tk graphical scripting language for X windows and provide a high level of abstraction to easily implement desired actions. These systems are described in detail in section 4.3

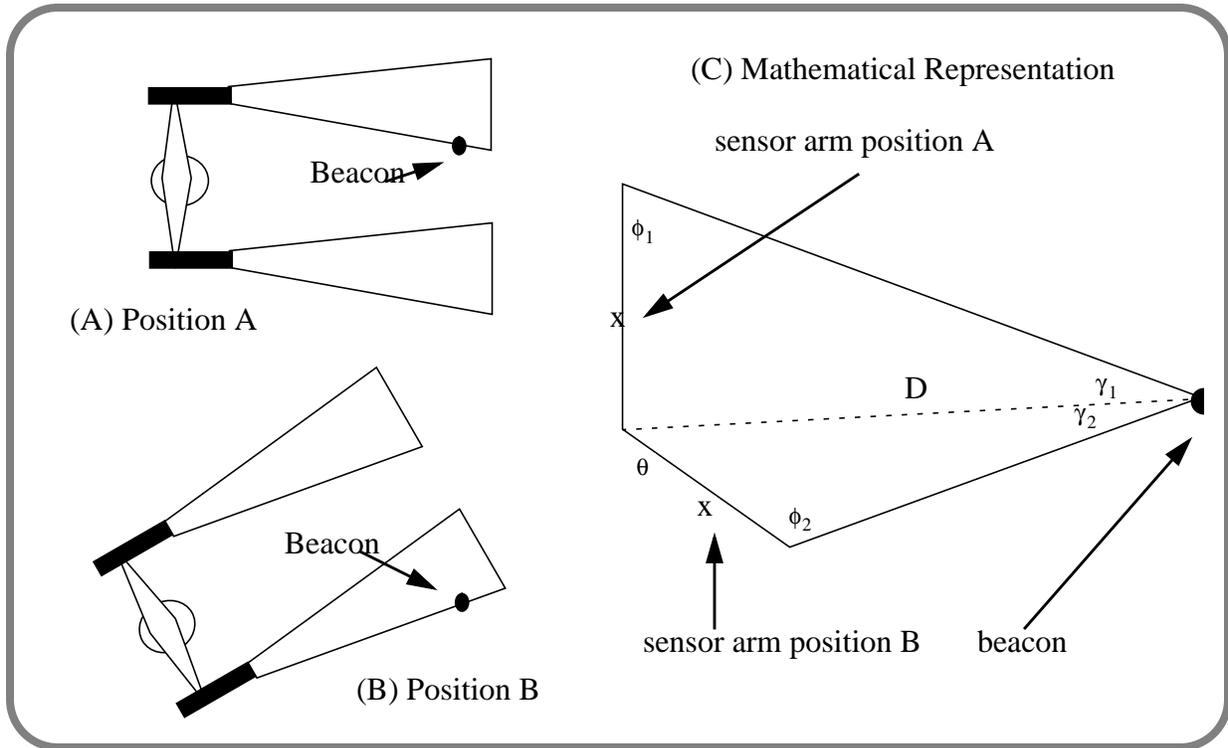
4.2 Architecture

One of the main objectives in moving to the vxWorks based system was the processing speed and amount of memory that it provided with the 68030 processor. Our organization was very interested in developing a very expandable and powerful architecture for Sidewinder's control. With these resources, we could develop high-level software that made use of advanced software engineering techniques. We designed a system that was based on messages. A server would receive all of the messages from the processes and route them to the appropriate destination process.

Sidewinder's architecture breaks the code up into modules. Each module is a task which performs a specified task, and communicates with other tasks using IPT. IPT is a multi-platform, multi-processor interprocess communications system developed at CMU which handles all of the details involved in passing messages and data around within the system. The flexibility of IPT allow us to execute some modules onboard the vxWorks processor, while others are executing on the Sun workstations. This helps make developing modules much simpler. Also, in the future when we decide to go to multiple onboard processors, IPT handles the division of processes seamlessly.

There are six basic modules. The gait controller had the high level code to control the rotation of the legs and placement of the feet to generate a walking gait based on commanded parameters. The dead reckoning module keeps track of the current coordinates of the robot with respect to the starting location. An event controller runs a sequence of commands to complete an autonomous event. Typically, the event controller adjusts the gait controller's parameters to perform the desired task. The beacon module keeps in constant communications with the 68HC11 running the beacon to calculate the angle and range to the various beacons.

FIGURE 3.2 Beacon Tracking



Since we know θ , ϕ_1 , and ϕ_2 , we solve for γ , which we define as $\gamma_1 + \gamma_2$. So we have $\gamma_2 = \gamma - \gamma_1$. Substituting for y , we get,

$$\frac{\sin \phi_1}{\sin \gamma_1} = \frac{\sin \phi_2}{\sin(\gamma - \gamma_1)},$$

which becomes, $\sin \gamma_1 = c \cdot \sin(\gamma - \gamma_1)$, where $c = \frac{\sin \phi_1}{\sin \phi_2}$.

Hence, $\sin \gamma_1 = c \cdot (\sin \gamma \cos \gamma_1 - \cos \gamma \sin \gamma_1)$

$$\sin \gamma_1 (1 + c \cdot \cos \gamma) = c \cdot \sin \gamma \cos \gamma_1$$

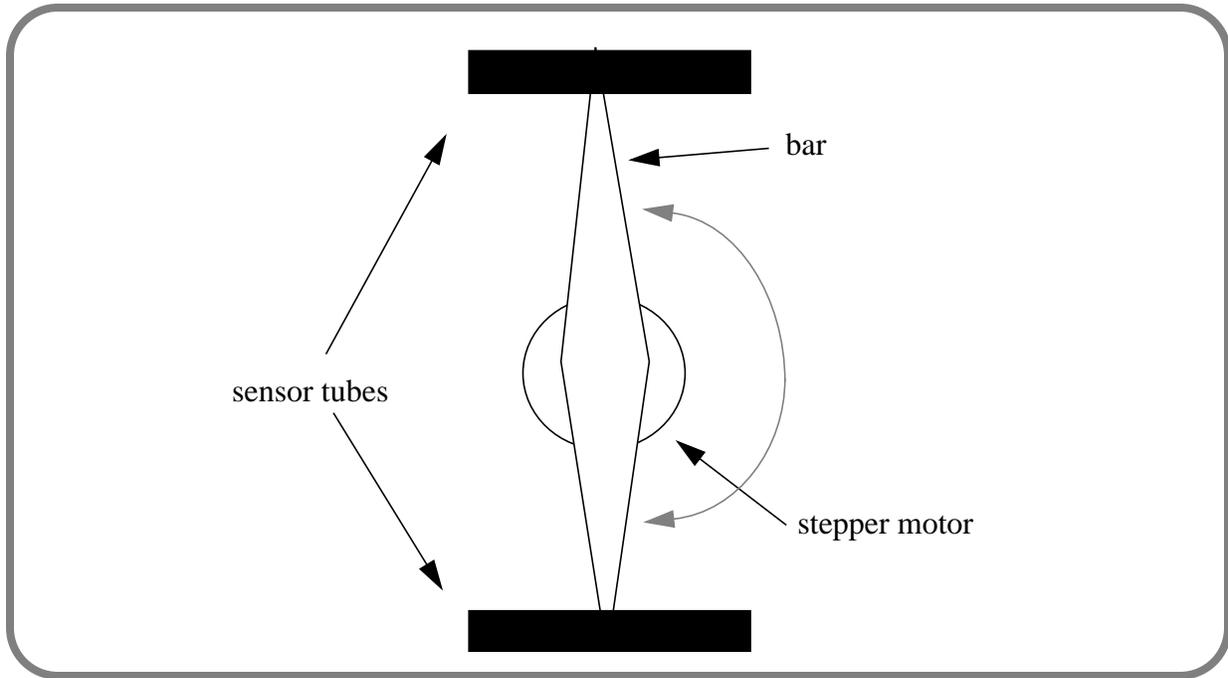
$$\tan \gamma_1 = \frac{c \sin \gamma}{1 + c \cos \gamma}$$

$$\gamma_1 = \text{atan}\left(\frac{c \sin \gamma}{1 + c \cos \gamma}\right) \quad (\text{Equation 3.3.4})$$

Substituting into equation 3.3.1, we have, $x \frac{\sin \phi_1}{\sin \gamma_1} = D$, and can solve for D.

The beacons are infrared 40 kHz carrier modulated at 150 to 500 Hz, depending on beacon channel.

FIGURE 3.1 Beacon Sensor



Since the stepper motor has no absolute position sensor built in, we have provided an IR slot sensor which generates a pulse at a constant rotation of the shaft. This is used during calibration to determine the actual angle of the sensors.

The pair of IR sensors are SHARP IR receiver modules. They detect IR signals modulated at 40Khz and provide a digital signal upon detection. These two signals are fed into two of the input comparison pins of the 68HC11. The 68HC11 monitors the observed frequencies, and uses this value to differentiate between multiple beacons set up on the course.

The infrared sensor uses two parallel sensor tubes, mounted a known distance apart to find not only direction, but distance to the beacons placed on the course as well. Mounted on a stepper motor, the sensor scans back and forth, finding the angles where the beacon enters and leaves the fields of view of the sensor tubes. Figure 3.2 demonstrates how the sensor converts these angles to distance data using the right edges of the fields of view. Figures 3.2a and 3.2b show the sensor position as it senses the right and left sensor tube fields of view. Figure 3.2c shows the geometric representation of this problem. Known are: θ , the difference in angles, ϕ_1 and ϕ_2 , the angles that the fields of view make with the perpendicular bar, and x , the displacement of the sensor tubes from the point of rotation. To find D , the distance of the beacon from the point of rotation, we solve for γ_1 or γ_2 using the law of sines.

$$\frac{\sin \phi_1}{D} = \frac{\sin \gamma_1}{x} \quad \text{and} \quad \frac{\sin \phi_2}{D} = \frac{\sin \gamma_2}{x} \quad (\text{Equations 3.3.1 and 3.3.2})$$

which gives us

$$\frac{\sin \phi_1}{\sin \gamma_1} = \frac{\sin \phi_2}{\sin \gamma_2} \quad (\text{Equation 3.3.3})$$

reset. Additionally, we needed to keep the button-box control available to take over at any time. Our solution allows either the button box or computer to command the signals at any time. This prevents the need for a form of mode switch. The button box is set up with higher priority, and is able to override the computer.

Due to the position servo control supplied by the DCX board, the tachometer feedback to the motor amp is not needed (or desired) while operating under computer control. Since we need the feedback while controlling it from the button box, a single relay has been included to disengage the feedback when the computer initializes the robot before a run.

To make software development easier, we have provided a tap into the power system for an external +24V supply for the VME cage. This supply and the 24V batteries are connected with a pair of power diodes to allow either the wall supply or batteries to run the cage. To be truly useful, we had to make sure we could transition from battery to wall and vice versa without affecting the computer. This way, we can save all of the battery energy for the event itself, not preparing for it.

3.3 Beacon System

To provide a limited form of perception for Sidewinder II, we have developed a beacon location system which is used as landmark objects for certain autonomous events. This sensor, mounted atop the robot, continually tracks the beacons and determines their angle from the robot, as well as distance. This is accomplished by a stereo pair of IR detectors. A 68HC11 microcontroller controls the beacon system, and interfaces to the 68030 CPU by a serial link. A dedicated 68HC11 was chosen to allow a separate development of the beacon system, as well as for its internal timing circuitry which allowed us to handle all of the signal processing without additional hardware.

The sensor consists of three components: a stepper motor, pair of IR detectors and a home position sensor for the stepper (figure 3.1). The stepper is used to aim the sensor in a specified direction. The stepper is a four-phase unipolar stepper which is driven by a custom stepper motor driver board controlled by the 68HC11. For normal operation, the phases are excited in adjacent pairs for additional holding torque.

The power is supplied by a pair of 12V lead acid batteries. The batteries have also been fused to prevent dangerous current levels due to a short or motor stall. Additionally, the button box contains an emergency stop button for safety during development and the competition.

3.2 On-board Computing

To facilitate some of our project goals, we decided to mount a small VME computer rack on Sidewinder II. While this gives the software development many advantages, and great processing power, it provides quite a challenge from the electrical standpoint.

A large challenge in using the VME system on the robot is supplying it power. Sidewinder initially used a CPU which required a supply of more than 10 amps at 5V! However, in the newest processor configuration, the required current is only 5 amps. The VME cage requires +5v, +12v and -12v for proper operation. Additionally, these need to be well filtered supplies to ensure proper operation. Due to size and weight constraints, we decided that two batteries which run the button-box system would be used to run the VME cage. The concern with this was sharing a supply for both the motor and computing, as the inductive kick from the motor can cause serious damage to sensitive components. We utilized a number of Vicor's DC-DC converter modules, as well as an input attenuator to provide an isolated, well filtered supply to the VME cage.

Within the VME cage we mounted 4 cards. They are: a 68030 CPU board, ethernet board, DCX-100 motion control board and VMIO digital I/O board.

The CPU and ethernet boards are setup to allow direct access to the robot over the internet for both software development, and also to provide a communications link when manually controlling the robot from the computer. When in tethered mode, a single RG-48 coax cable is used as the only tether attached to the robot, minimizing interference while maintaining complete functionality and control.

A motion control board is used to execute position servo control for the main drive motor. An optical encoder mounted on the motor provides feedback to the board. This allows our computing system to accurately track the angle of the arms to within 0.01 degrees. (Easily within the tolerances of the mechanism.)

While the encoder is able to provide extremely precise accuracy, it is of little good if we are not able to repeatably and reliably initialize the legs on start-up. To facilitate automatic calibration, and more repeatable performance, a reflective sensor is added to one of the arms and provide a pulse at a known arm angle. This position sensed will be fairly close to the real home position, the encoder also has a built in index pulse which can be used to complete the calibration, thus allowing initialization accuracy to the resolution of the encoder.

The digital output board is configured to output digital control signals for each of the eight legs to command raising/lowering of the leg. These signals are optoisolated on an external board, and attached to the solenoid driver board to control the legs.

Another challenge was designing the interface between the computer and the robot, it was very important to ensure that all signals coming from the computer defaulted to safe values on start-up/

3.0 Electrical Design

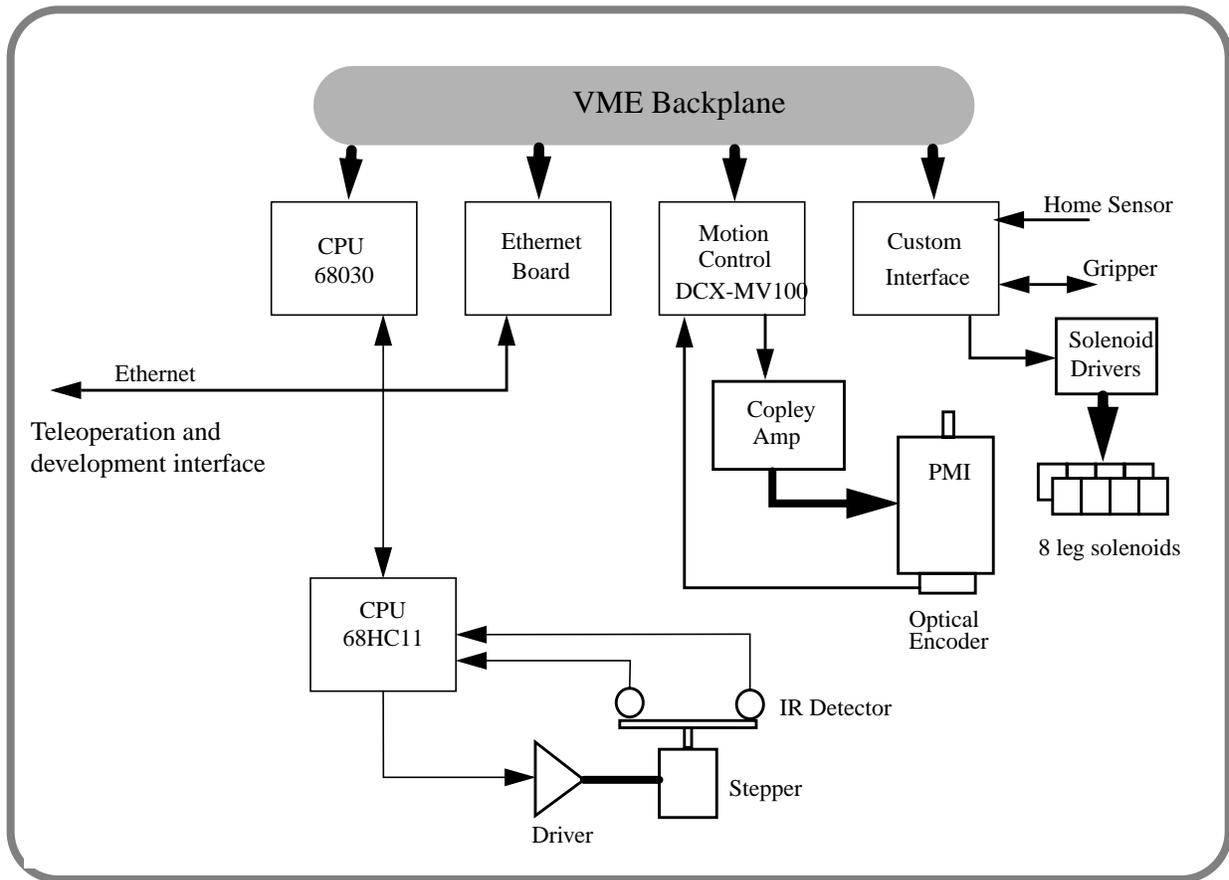
3.1 Button Box Design

The button-box system has been designed to be the lowest level controller for Sidewinder II. The button-box system's primary purpose is to provide a collection of switches which allow control each of its actuators to a human operator. Additionally, the button box is required to override any computer commands, thus serving as a backup control for events initially under autonomous control.

To facilitate walking, the controller has a single switch which alternates the set of legs which are extended, thus providing support. This switch acts as an open collector output to the solenoid driver board. This board takes eight digital inputs, and supplies the needed current to control the solenoid valves for the pneumatic valves.

A second switch is used to rotate the arms. A potentiometer provides a speed control for the rotation. To allow smooth, constant speed rotations of the arms, a tachometer was used to provide feedback to the motor amplifier. The motor amplifier drives the single DC motor with up to 14 amps of current at 24 volts. A block diagram of the electronics required to implement this system is shown in figure 3.0.

FIGURE 3.0 Electrical Overview



The area the wiper covers is approximately 1 foot x 1 foot. This parameter is used to determine the minimum acceptable positioning accuracy of our beacon system, in order to allow autonomous completion of the rock recovery event.

of the arm shaft. If the tubing were to be directly routed to the valves, the software would have to have a memory of how many times it has been wrapped in the same direction or else the tubing would get entangled and potentially rupture. Movement would be limited to “x” amount of arm rotations in the same direction. Hence, a rotational coupling of the pneumatics is required to provide unrestricted movement. Specifically, dual channel pneumatic rotating unions were affixed atop the arm shaft allowing the arm to rotate freely while passing the CO₂ through the union to the cylinders. The pneumatic system is depicted in figure 2.3.

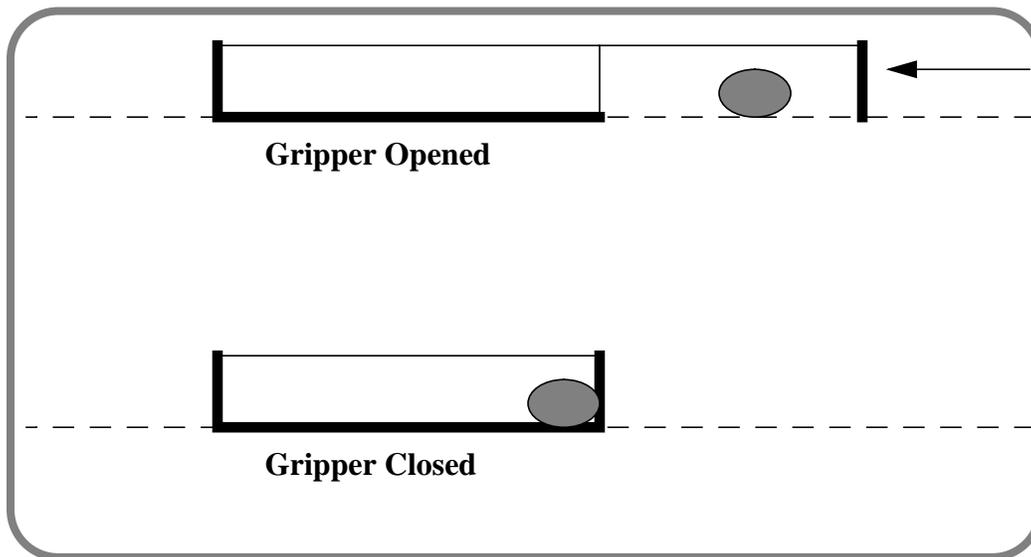
2.5 Rotating Feet

As Sidewinder walks, one of the two sets of legs are supporting the vehicle. During this motion, the bottom of the leg is has to rotate with respect to the floor. To facilitate this motion, we have built feet which include a rotational bearing. This bearing of allowing the leg to turn freely, while supporting the full weight of the body. The foot is make of aluminum, and is coated with a layer of rubber to protect the floor from scratching.

2.6 Gripper System

In order to pick up the river rock, we have designed a gripper attachment for Sidewinder. Attached by two clamps, the gripper appears much like a desk drawer turned upside down under the center of the robot. The gripper has only one degree of freedom, see figure 2.4. A movable “wiper” is drawn towards the open end of a holding box on the gripper. This guides the rock into the gripper’s holding box, presuming the rock is located within the grabbing area of the gripper.

FIGURE 2.4 Gripper System



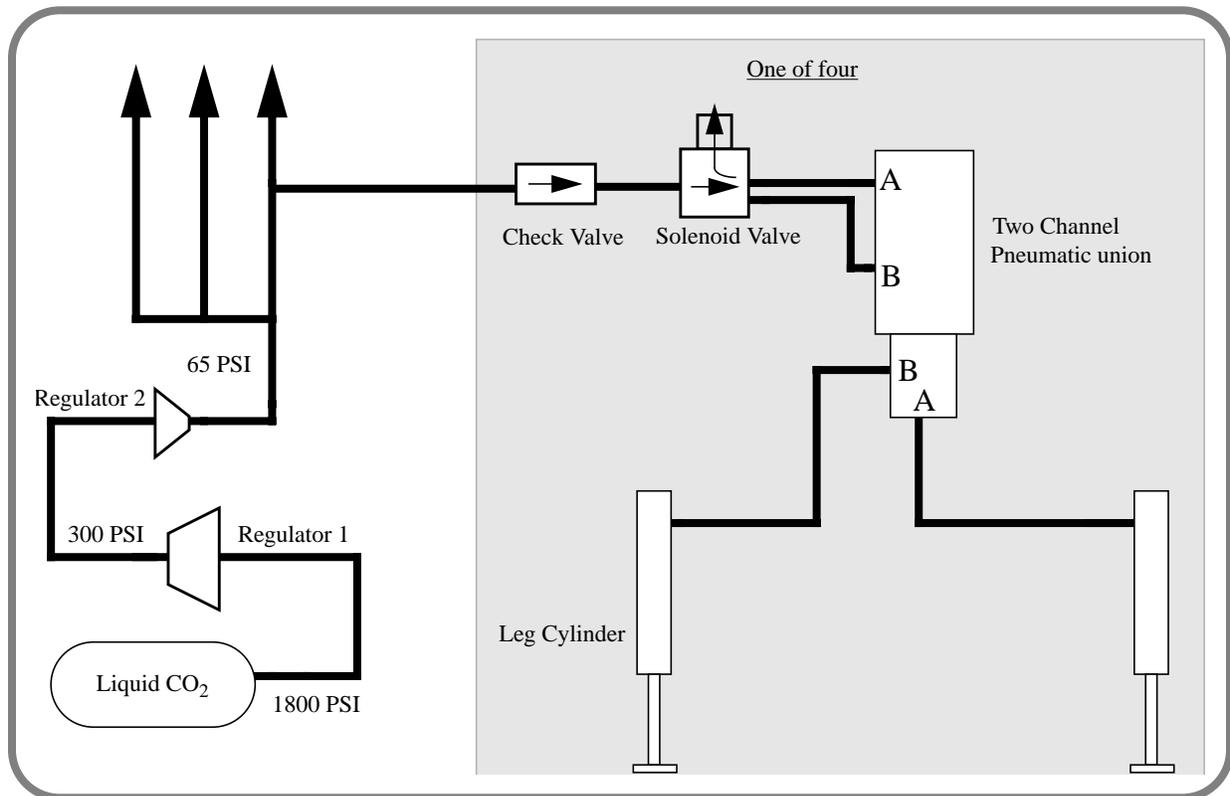
The gripper is mounted so that it does not contact the floor when the robot is walking. Instead, we take advantage of the robot’s ability to raise/lower the entire body via its legs. When sidewinder raises all legs, the robot sits, and the gripper is at the floor level.

drive the cylinders. The advantage of using a liquid versus a completely gaseous CO₂ tank is two-fold. First, we get the advantage of the phase change. That is, as CO₂ is released from the tank it experiences a phase change from liquid into gas. This allows the tank size to be drastically decreased as compared to a tank which was solely gaseous, thus saving weight and volume. Secondly, the phase change of the CO₂ is endothermic, and cools the ambient air surrounding the tank. By redirecting the cooled air through the VME cage, we save energy by not having to run the cooling fan at high speeds.

The pressure of the liquid CO₂ is 1800 psi in the tank. We do a double regulation to get our desired 65 psi for walking. The first regulator reduces pressure from 1800 psi down to 300 psi. The second regulator then does a reduction from 300 psi down to 65 psi.

With a centralized pneumatic source that needs to supply CO₂ to cylinders at the corners of the robot, there is a significant amount of hosing used. Consequently, the more weighted side of the robot supplies more backpressure against the cylinders on that side which in turn causes a back-flow of pressure through the hosing. To remedy this problem, we installed check valves, which allow air to flow in only one direction. That way, once the air has extended the cylinders, it cannot backflow through the system. Furthermore, these valves ensure safety. If any of the hosing were to burst before the valve, the machine would not drop or droop until we commanded it to do so. The switching of air is controlled by solenoid valves.

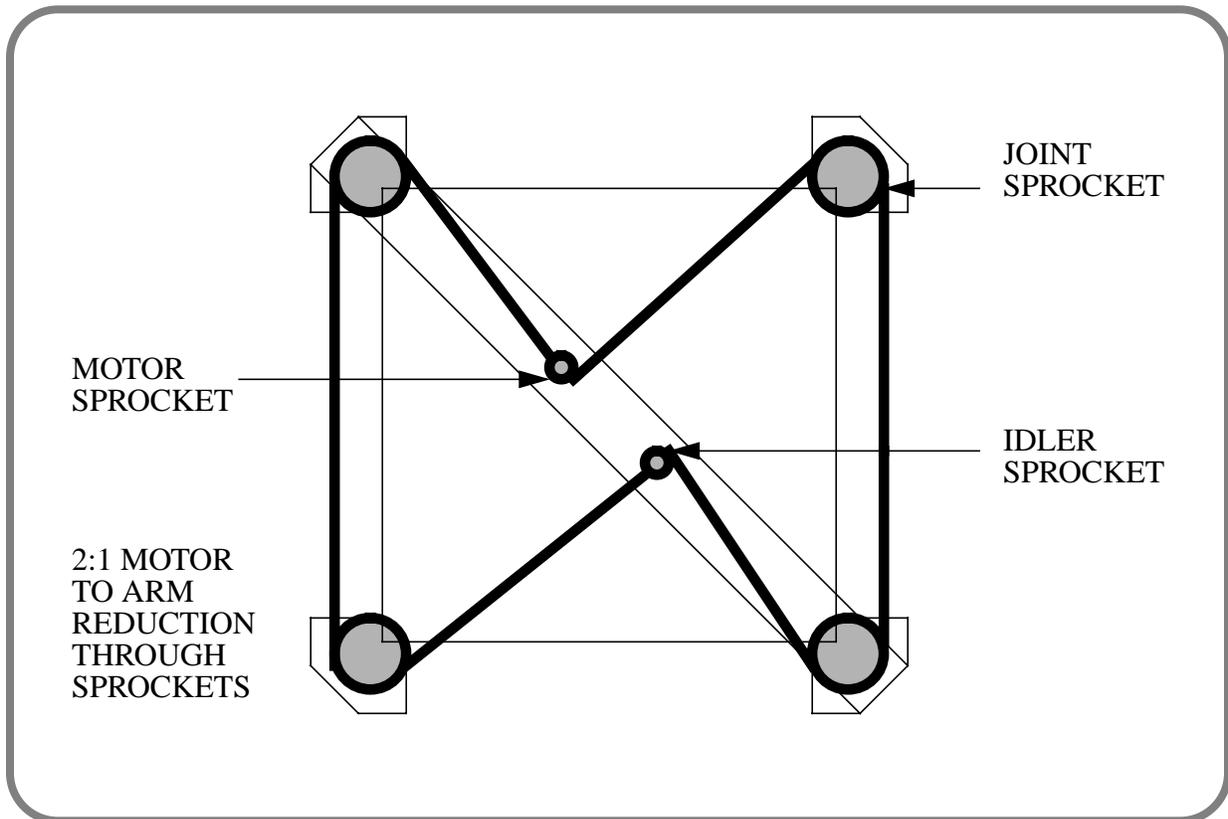
FIGURE 2.3 Pneumatic System



The pneumatic tubing from the cylinders are fed into the hollow arm shaft and out through the top

ets are clamped to the arm shaft with a special compression bushing-sprocket pair that grip the shaft from all sides, providing uniform friction. An idler sprocket with roller bearings is used in a symmetrical fashion with respect to the drive sprocket to provide high tension throughout the chain, which ensures that all the arms are in synch. The drive sprocket and idler sprocket are both mounted to the c-channel cross support. They were mounted symmetrically to achieve counteracting forces on the c-channel.

FIGURE 2.2: Drive Train



Drive train specifications:

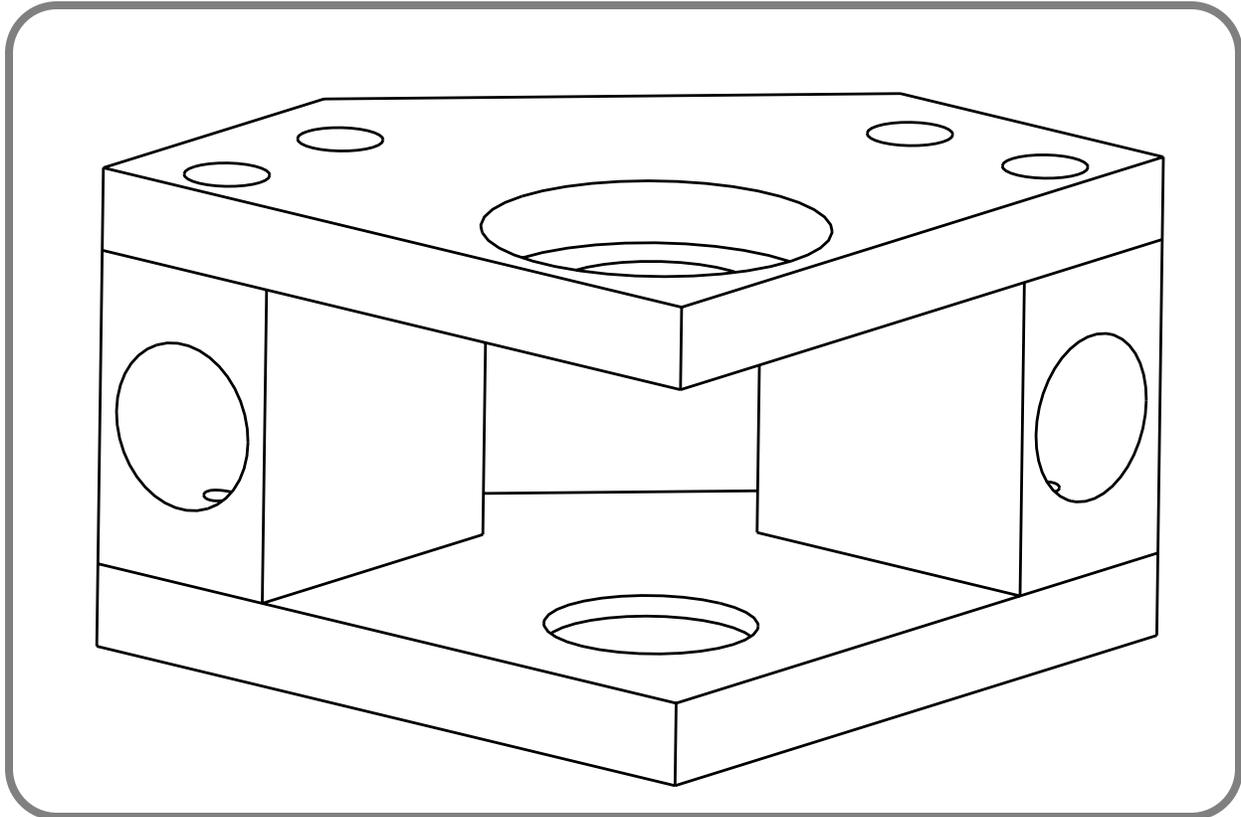
- | | |
|-------------------------------------|--------------------|
| • Continuous available motor torque | 97 in-lb @ 150 RPM |
| • Peak motor torque | 386 in-lb |
| • Motor weight | 5 lb |
| • No. 41 riveted chain | 1/2" pitch |
| • Chain weight | 0.5 lb/ft |
| • Chain speed reduction ratio | 2:1 |
| • Continuous available torque | 194 in-lb @ 75 RPM |
| • Peak Torque | 772 in-lb |

2.4 Pneumatic System

Sidewinder is unique in that it combines both a drive train for translation with pneumatics for step transition. The pneumatic system uses a liquid CO₂ tank, which provides the pressurized air to

when standing on only one set of legs). The arm shaft extends through both arm supports, providing two points of contact, as is provided in the joint block at the top of the arm shaft. The arm shaft provides the structural strength needed, but also acts as a pneumatic hosing management device. The shaft was made hollow so that the pneumatic tubing could be funneled through the tube and out two drilled holes to the pneumatic cylinders.

FIGURE 2.1: Joint Block



Atop the pyramid is a plate that serves as a mount for the beacon sensor and button box connector. The pyramid side tubes are connected by this top plate by welds, adding yet more rigidity to the structure.

2.3 Drive Train System

The drive train was specified to allow a 100 lb robot with the dimensions specified in section 2.1 to walk up a 20 degree incline. The required torque at each joint was determined to be 71 in-lb. This requires 273 in-lb supplied by the drive train. For reasonable progress, we decided that the rotational speed of the arms while climbing the hill should be at least 10 RPM. To allow sufficient flat floor walking speed, we required the drive train to supply 60 in-lb of torque at 75 RPM.

The drive train consists of a single DC motor (with integral gearhead) to which a drive sprocket is attached to drive four driven arm sprockets via chain. The chain routing is shown in figure 2.2. The drive sprocket is attached by a setscrew keyway to the motor shaft. The driven arm sprock-

shape. Although the base of the pyramid is square, we have added structural support across one pair of opposite corners, splitting the base into two triangles. To make use of this cross support, we widened it to provide the support for the computing, motor, batteries, and pneumatic tank. A 5" wide c-channel was used for this cross support. The c-channel shape was chosen to minimize the natural tendency of the cross support to twist. The wide c-channel cross support prevents compression and expansion of the frame horizontal to the base. The pyramid, coupled with the c-channel, prevents torsional forces on the frame.

The dimensions of the pyramid were determined by maximizing the size of the base to meet the one meter constraint, while trying to maintain the equilateral triangle shape of sides of the pyramid. Adjustments to the frame dimensions were done by calculating clearance between the pyramid and the motor, VME cage, and batteries. The robot was to have a maximum stride of 42 cm (35.4"). This dimensioning of the arm length was determined by maximizing the crevasse clearance, while minimizing the induced torque. The minimum clearance of the underside of the robot to the ground was specified to be 34cm (13.4") to insure passage over the peak of the 20 degree hill with a 5" leg stroke.

Having determined the shape of the robot, we designed the most crucial structural part of the robot - the joint block. This block serves multiple purposes:

- Support three pyramid side tubes
- Support the cross support c-channel
- Withstand high torques
- Provide double point contact of the rotating arm shaft
- Shield the driven arm sprockets for safety
- Serve as a mounting area for absolute positioning sensors
- Allow for easy assembly/disassembly

The joint block consists of a top and bottom plate, which sandwich two blocks, as shown in figure 2.1. The joint block is fastened by four through bolts. The two blocks that separate the top and bottom plate contain larger holes through which the base tubes of the pyramid were inserted. The through bolts go through the inserted tubing as well. The pyramid base tubes were welded after being inserted into the separating blocks, ensuring a rigid frame. The tubes coming from the top of the pyramid are welded to a plate which is mated to the top of the joint block. This plate is affixed by two of the joint block through bolts. This allows for easy disassembly of the pyramid, without completely dismantling the base frame. The top and bottom plates house two high capacity angular contact bearings that combine to provide two points of contact to counteract the highly torqued arm shaft, but allowing free rotation of the shaft. Finally, the joint block serves as a mounting platform for the c-channel (on two of the blocks) and the absolute positioning sensor (on one of the blocks).

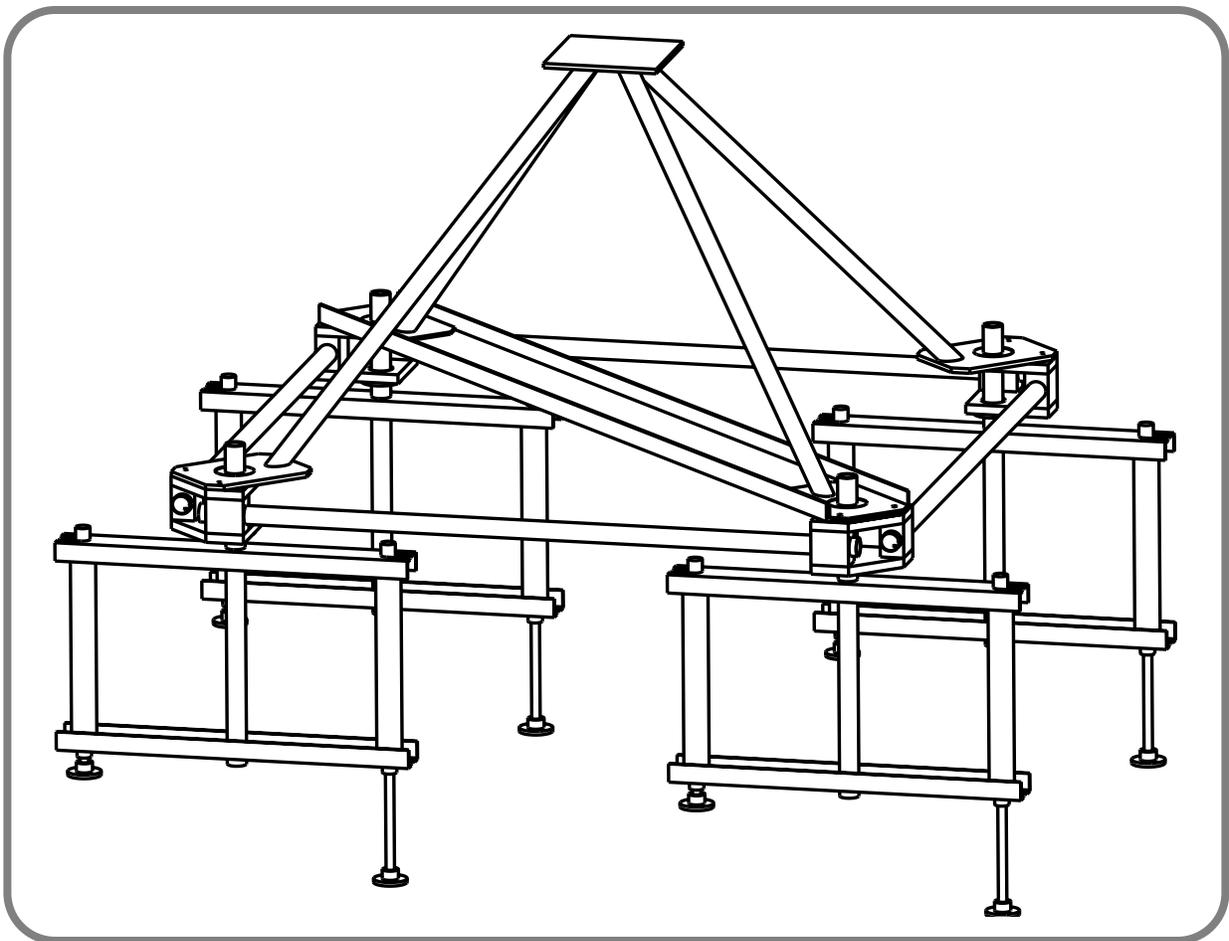
The arm shaft is a 1/8" wall 1" OD tube that is free to rotate from the angular contact bearings in the joint block. The top of the shaft is threaded. A retainer ring grabs the inner race of the lower bearing and a nut that is screwed on the top of the shaft grabs the inner race of the top bearing providing adjustable pre-tensioning of the bearings. Welded to the shaft are the arm supports for the pneumatic cylinders. These arms are c-channels to resist flexing due to uneven support (such as

2.0 Mechanical Design

2.1 Goals

- Satisfy the event requirements.
- Construct a rigid frame that will yield fast, repeatable motions.
- Minimize weight.
- Utilize pneumatics to increase step transition rate.
- Simplify the control software and enforce repeatability with a ganged drive train system
- Develop a simple gripper attachment that can lead to fully autonomous object retrieval.
- Provide an aesthetically pleasing shell
- Shield against potentially dangerous areas of the robot

FIGURE 2.0: Structural Drawing.



2.2 Structural Design

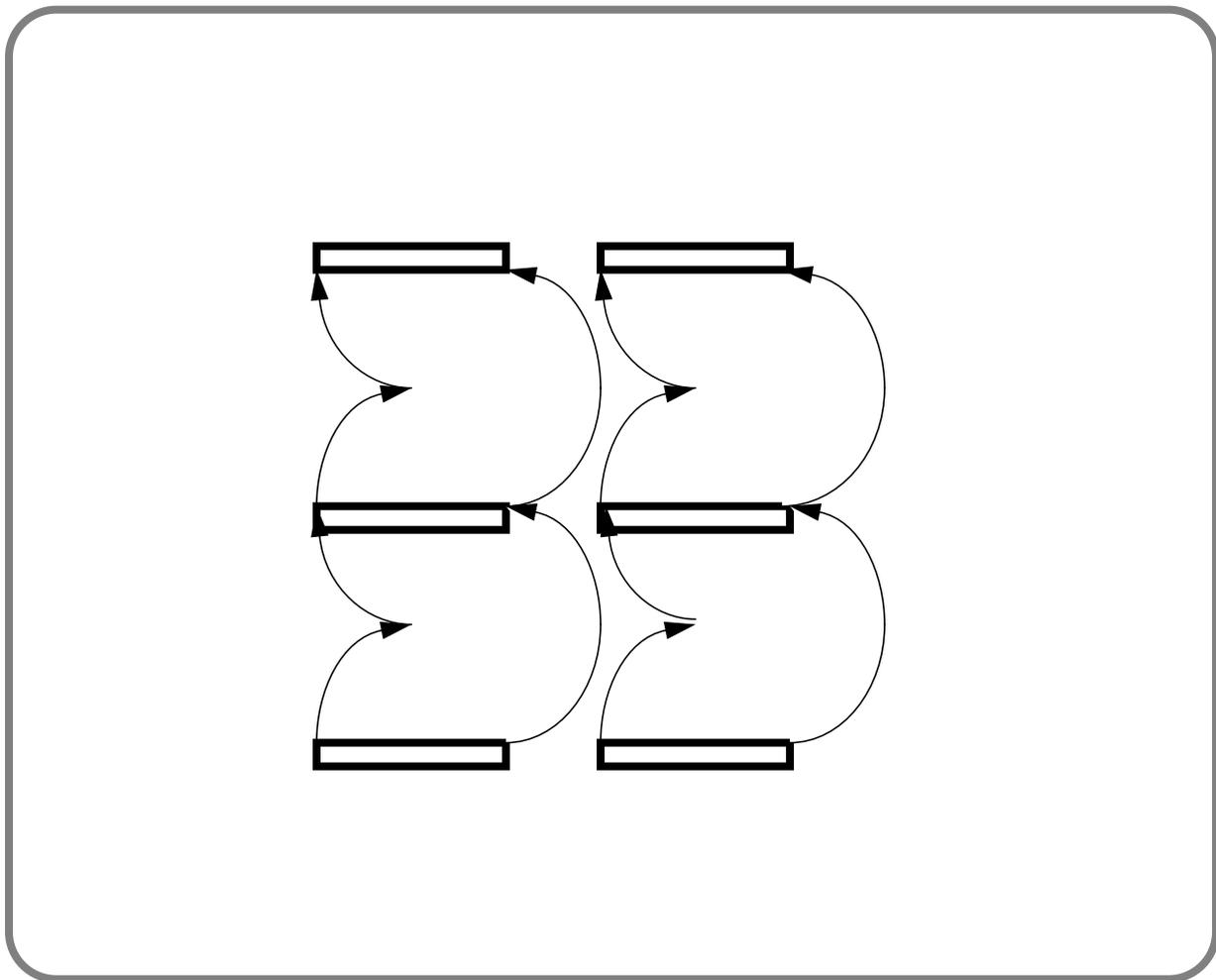
To provide the structural stability required to handle the competition events and minimize both weight and cost, we used primarily 6061-T6 aluminum stock. The sides of the pyramid frame were constructed of 1/16" wall, 1" O.D. tubing as depicted in the structural diagram (figure 2.0). The pyramid shape was chosen for optimal strength, since the triangle is the strongest structural

1.0 Kinematics

Sidewinder received its name from its kinematics. It swerves back and forth during walking in a snake like manner. The cover of this report illustrates Sidewinder II in its bare form, void of any electronics or coverings. Referring to this illustration, notice that all the legs on the right side of the arm are extended, while all the legs on the left side of the arm are retracted. In this configuration, the frame can be moved in a circular motion, given that the arm joints are free to rotate. If desired, the frame could be rotated 360 degrees back to its original position. By rotating the frame 180 degrees from its current position, the robot will have taken the largest stride possible (the frame will have translated arm distance). Upon reaching that position, the robot can extend the four other legs and retract the ones that it just rotated about. We call these two sets of four legs frame A and B. Now the robot is free to translate the frame forward again. However, the stride can be any value, as can be the bearing at which the robot walks.

To illustrate the gait of the robot, figure 1.0 shows the robot doing a forward walk changing its stride as it goes

FIGURE 1.0 Motion Tracing



0.0 Overview

In the 1995 Annual Walking Machine Decathlon held at Colorado State University(CSU), the CMU robotics club presented Sidewinder II. Having successfully demonstrated the mechanical abilities of Sidewinder, the club has concentrated on developing a new software architecture which allows for truly autonomous operation. Our interests have been primarily in improving Sidewinder's abilities beyond those which are needed for this competition and to explore software control techniques which would be applicable not only in the Decathalon events, but in many areas of robotic manipulation

0.1 Acknowledgments

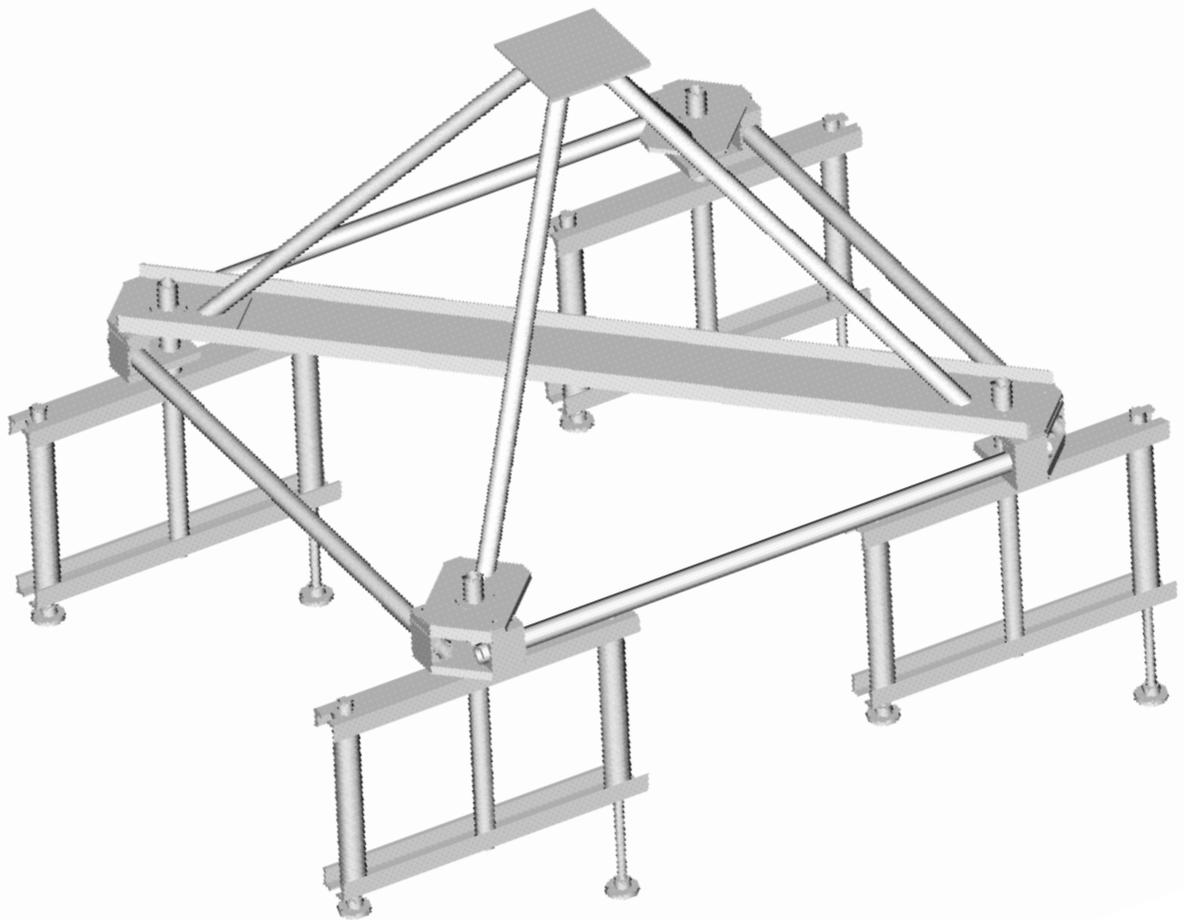
The CMU Robotics Club is very thankful for all of the support we have received while developing Sidewinder. In particular, we would like to thank the following individuals:

John Wiss
Jim Martin
John Bares
Scott Boehmke
Dimitrios Apostolopoulos
Charles Ouyang

We would also like to acknowledge the following companies and organizations for supporting us with donations and use of facilities:

CMU Field Robotics Center
Society of Automotive Engineering
Deublin Co.
Performance Motion Devices
Vicor, Inc.
SKF USA, Inc.
ARO
Neumatics, Inc.

Sidewinder II



**presented by the Carnegie Mellon University Robotics Club
1996**