triMAXion

Robert Nagel, Kenneth Perry, Jeremy Schueler,
Tarang Parashar & Daniel Abbott

Advised by: Dr. Robert Stone

Missouri University of Science and Technology

1. Introduction

The robot, triMAXion, has been constructed by team Abhiyantrik as a part of two design courses at the Missouri University of Science and Technology (Missouri S&T) to compete in the 2008 IGVC. TriMAXion was built with the objective of producing a modular robot both in software and in hardware which integrates off-the-shelf components with both the software and hardware systems into an innovative robot design. This report discusses the innovations in the design of triMAXion as well as the design process that was performed.

2. Innovations

A number of innovative design elements have been incorporated into the design of triMAXion including features for modularity in software and hardware, feedback to the operator and remote status monitoring and operability.

TriMAXion runs an innovative multithreaded software application developed to control and collect data from each of the sensors, provide user feedback via a graphical user interface, and control the robot's operation. The multithreaded application allows the software to continue running smoothly while taking advantage of both processor cores of the robot's laptop computer, and provides the benefit of software modularity. When a new device is added to the robot or a new application is desired, triMAXion's entire code base does not require modification. Instead, a new thread can be added to control the new hardware device and ready information for the rest of triMAXion's threads to access or a thread can be added to allow triMAXion to perform new operations for new applications.

TriMAXion is constructed of two main physical entities, a lower and an upper portion. The lower portion of the robot utilizes the chassis of a power wheelchair for batteries and motors and includes an undercarriage mount point for motor controllers and electronics. The upper portion is the body of triMAXion and is divided into two main compartments with one for the payload and other for the computer and electrical components. The two portions of the robot can be separated easily via four screws and four wire disconnects. All of the sensors have standard center-positive plug power connectors and are connected to triMAXion's onboard laptop computer via either USB or FireWire (IEEE 1394) standards allowing for quick and easy installation and removal of devices for testing and expandability.

The design of triMAXion includes an innovative graphical user interface (GUI) termed the dashboard, which displays the status of the robot on the monitor of the onboard laptop computer. The dashboard provides information on the obstacles detected by triMAXion's sensors, the current cardinal direction, coordinates received via GPS, sensor connection status, software thread status, and current control mode (automated or manual).

Since triMAXion is built around a standard laptop computer, remote operability is afforded via an ad-hoc network between the robot's onboard computer and a remote machine using both machines wireless network cards and Virtual Network Computing (VNC). VNC allows triMAXion's dashboard to be viewed on the remote computer's display for remote monitoring of robot status, and in the event of unexpected operation, a user can take control of the robot from a remote computer by switching the robot into its manual control mode.

3. Design Process

The realization of triMAXion consists of two parts: (1) conceptual design beginning with the identification of requirements and concluding with the selection of a concept, and (2) the detailed, physical design beginning with the specification of design parameters and concluding with the final physical prototype of triMAXion. The flow chart of the design process provided as Fig. 1 identifies both the conceptual and physical portions of the design process and subdivides each into the subtasks connected by arrows representing the progression of the design of triMAXion.

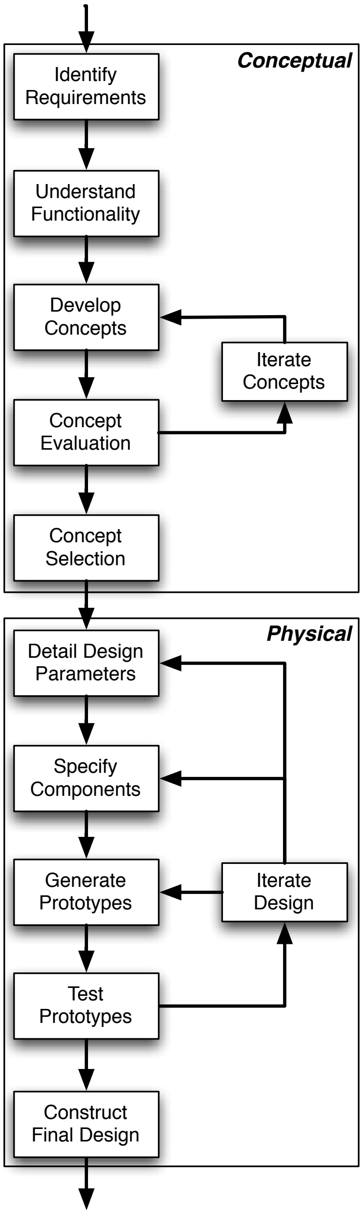


Figure 1. Design process

The design and construction of triMAXion occurred over the course of two semesters as a part of two design courses at the Missouri S&T. During the first semester, the conceptual design and the first two stages of the detailed, physical design were performed by mechanical engineering graduate students Robert Nagel, Tarang Parashar, Daniel Abbott and senior interdisciplinary engineering student Jeremy Schueler. Over the course of the second semester, the remainder of the physical design has been completed by Robert Nagel and sophomore computer science student Kenneth Perry. Over the course of the two semesters, Robert Nagel was the team leader for the project.

To begin the conceptual design, the team identifies requirements, regulations and objectives of the IGVC. Each requirement is reworded as an objective statement and organized into three main categories: (1) safety, (2) navigation and (3) physical attributes/computation to assist with understanding of the design problem. A functional analysis is performed to translate the abstract requirements into specific functions which must be performed in the selected design to meet the IGVC requirements. These functionalities are aggregated into a single functional representation of the entire robot. The functional representation, shown in Fig. 2, uses boxes to represent functions and arrows to represent the flow of materials, energies and signals in the design of the robot. The functional representation contains elements such as *detect solid* for determining the presence of vertical obstacles, *process status* for understanding and making decisions about navigational data, and *actuate electrical energy* to represent the startup of the robot.

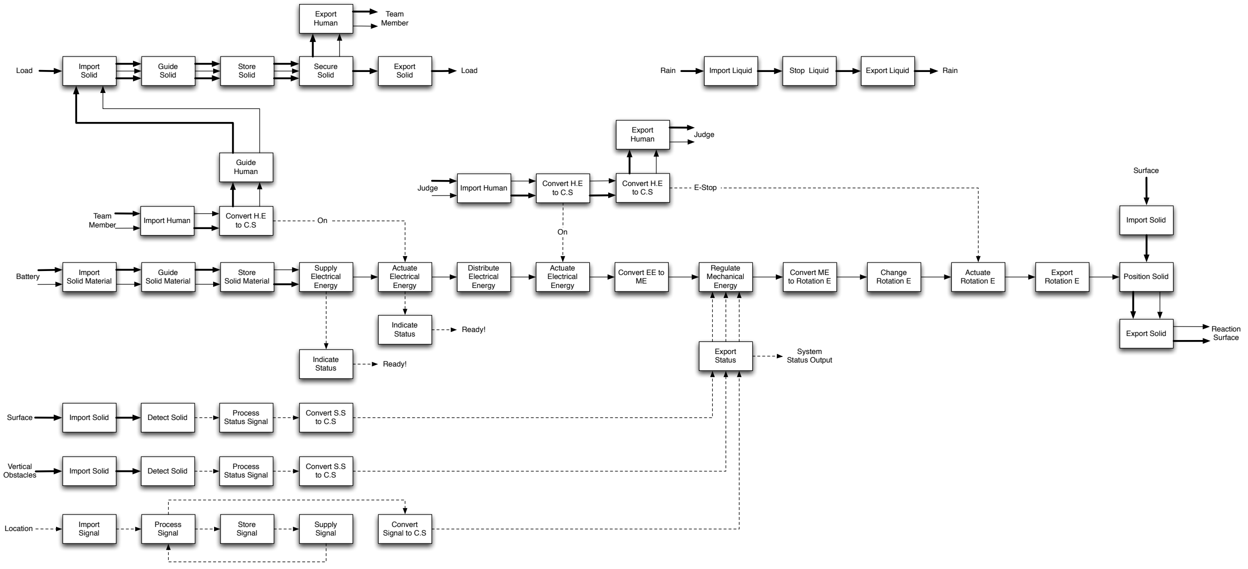


Figure 2. IGVC robot functional model

The mapping of functions to requirements allows the team to intrinsically understand how the robot must operate and perform without considering potential components as well as allowing the component identification to be decoupled during the identification of requirements, regulations and objectives. During the development of potential solutions, the team identifies components and design principles at an abstract level based on the functional requirements. Components and design principles are mixed-and-matched to develop thirty concepts, which are evaluated based on their feasibility and ability to meet the identified requirements. The concepts which are identified as having the most potential are refined until the team selects the best solution for the identified criteria.

Once a concept is selected the team moves into the physical portion of the design process. Specifications such as field of view and range for obstacle detection are considered as well as the robot speed and payload requirements. A physical component solution (or in some cases multiple physical component solutions) is paired to each abstractly defined component in the concept. Prototypes are built to test the feasibility of some of the more abstract design principles and to refine the selection of components until a final bill of materials is generated for the robot. Software algorithms are developed to pair with each solution. As the robot is constructed, hardware and software are assembled as modules consisting of: (1) sensors; (2) body and chassis; (3) motion and electronics; (4) pathfinding and (5) intelligence. As each module is added to the robot, testing is performed to confirm desired operation. The result of which is the completed triMAXion, IGVC robot.

5. Integration Strategy

The integration strategy for design of triMAXion is to modularize the components both in hardware and in software. All sensors and controllers are connected via a standard FireWire or USB connection and receives power from either a 24V or 5V power bus. All power connections use a standard center-positive plug connector to connect to the power bus. The software framework is such that each component has a unique thread for activating, accessing and providing data to the main application. The addition of a new component to the system can be performed by plugging the device into an available USB or FireWire port, connecting to the power bus if external power is needed, and adding a new control thread to the software. When a component is not attached to the robot, its thread sleeps so as not to use processor time, which allows for testing without the burden of running the entire robot. The modular strategy allows for increased applicability of the robot’s design. Beyond IGVC, triMAXion can be fitted with various other sensors or actuators to perform various desired tasks. The following sections detail the sensors, electronics, motion and software systems that are integrated into the design of triMAXion.

4. Motion & Electrical Systems

4.1 Drive System

The drive system of the robot, triMAXion, is built on the chassis of a Spirit foldable power wheelchair shown in Fig. 3-left. The wheelchair is stripped down to the base chassis which includes the wheels, 90-degree gear motors with locking gear boxes, mechanical wheel brakes, batteries and battery over-current protection. The original controller and joystick for the wheel chair are removed and replaced with a Victor 885 speed controller for each drive wheel, shown in Fig. 3-middle. The Victor 885 speed controllers have been selected for their high current rating of 120A at 30V maximum voltage, which can handle the 60A current draw observed from the wheelchair motors during testing. Each Victor 885 is commanded via a pulse width modulation (PWM) signal generated from an Acces counter-timer module (Fig. 3-right). The Acces counter-timer module provides 15 16-bit counter-timers which interface with and are powered by a USB port on robot's onboard computer.

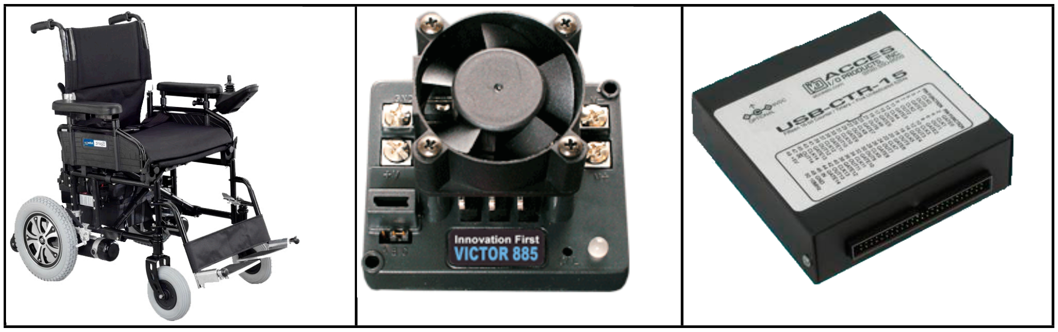


Figure 3. Spirit power wheelchair (left), Victor 885 speed controller (middle) and Acces USB counter-timer module (right)

To generate PWM signals for each speed controller, three counter-timers are wired such that the first timer is used as a square wave to change the 10 MHz internal clock to 1 MHz and is wired as the clock input for the second two timers. A second counter-timer is used as a rate generator which is accessed through code to change its output pulse width to affect the speed of the motor. The third counter-timer uses the output of the rate generator to generate a PWM signal for the Victor 885 speed controller. This wiring is shown in Fig. 4.

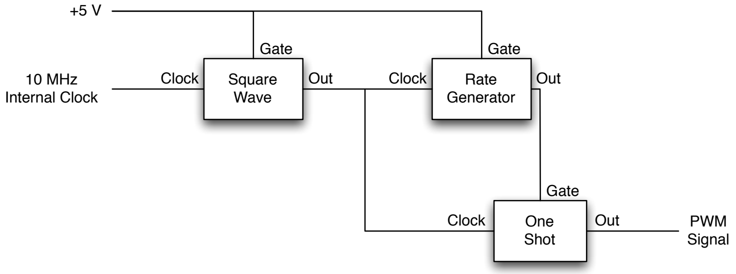


Figure 4. Wiring diagram for Acces USB counter-timer module

4.2 Emergency Stop System

The drive system of the robot is wired to integrate into the emergency stop (E-Stop) circuit such that power to the motors is turned off immediately once the wireless or the onboard E-Stop button is pressed. For the onboard E-Stop system, a 1.5 inch red button is located 27 inches from the ground at the rear of the robot for ease of access, and a RF module from a Heath Zenith model LE-6153-B wireless doorbell is modified for use in the wireless E-Stop system. The E-Stop circuit is shown in Fig. 5. The wireless and onboard E-Stop buttons are wired in parallel so that when either are pressed, the two internal switches of the latching relays open. Once the latching relay circuits open, current is stopped from flowing through the two 80A motor relays that control the flow of power to the motor controllers. A reset switch allows the circuit to close (reset) the latching relays. Latching relays are chosen for this circuit since they do not require continuous power to retain their state.

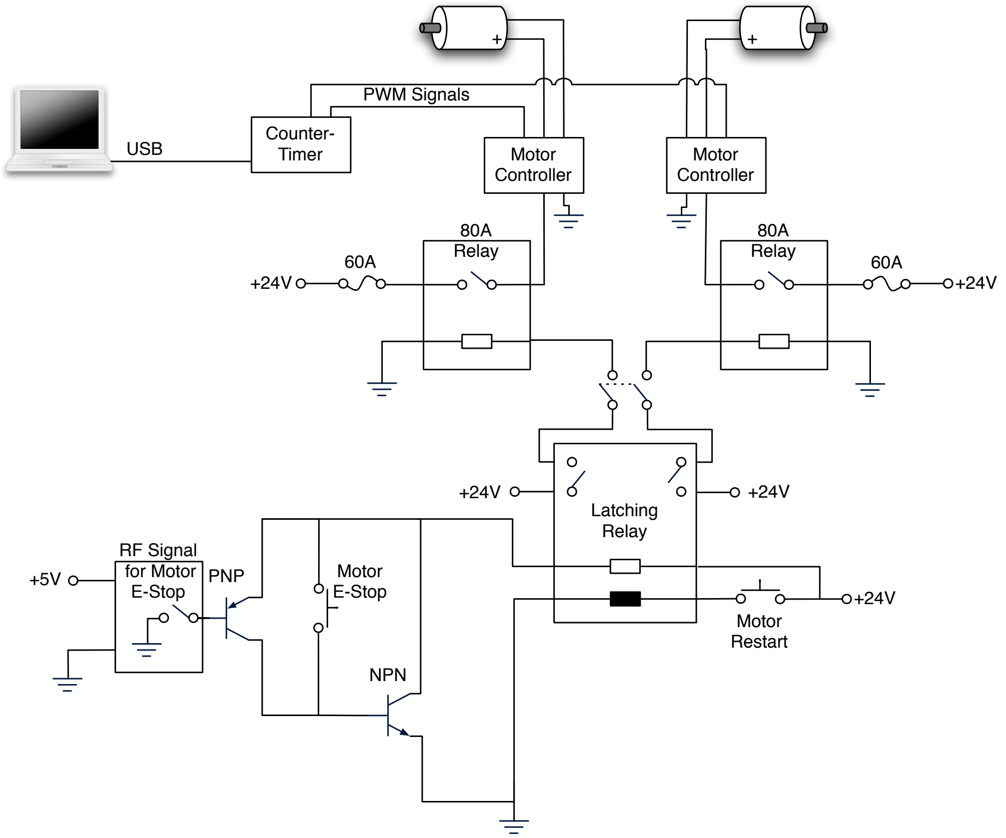


Figure 5. Wireless and onboard E-Stop circuit and motor drive circuit

The 80A motor relays add a layer of safety by separating the E-Stop circuits and the operator from the high current motor circuits. When the operator toggles the motor circuits on, the 80A motor relays toggle closed allowing current to flow into the two motor controllers. The motor drive line is wired with two 60A fuses to insure the safety of the two 80A motor relays.

4.3 Electrical system

The electrical system of triMAXion derives all of its power from the original batteries of the power wheelchair or if the device is USB powered, from the laptop battery. Power wires leading from the batteries are 8 AWG wire and carry 24V from two 12V lead acid batteries. A double pole, double throw (DPDT) toggle switch controls the power state of the robot from either charging, off or on. The charging circuit of the robot uses the original wheelchair charger which may be connected to the robot via an XLR connector when an operator wants to charge the batteries. For safety, the DPDT switch isolates the charger and batteries from the rest of the robot during charging. When the DPDT switch is in the on position, the motor drive circuit, the wireless E-Stop, and the 24V and 5V power bus for the sensors receive power. The power bus has been wired with center positive female plug connectors to allow for simple, quick connections.

5. Computational and Sensory Systems

5.1 Computational System

An Apple MacBook computer is used by triMAXion for all onboard computing. The MacBook computer runs Mac OS 10.5, and has a 13 inch display, 2.0 GHz Intel Core 2 Duo, with an 80GB hard drive and 1GB of RAM.

The MacBook computer provides a simple plug-and-play interface with both USB and FireWire for all sensors as well as the mobility interface with the Acces counter-timer module. The two USB ports of the laptop computer have been increased to five with a four port USB hub. To insure that the USB power bus on the MacBook is not exceeded, 5V is provided to the USB hub from the 5V power bus. A diode insures that if external power is lost to the USB hub, then power from the MacBook's USB cannot back-feed into the 5V power bus and electrical system.

For remote operability, the MacBook's wireless card is used to create a computer-to-computer network allowing for VNC to be used between the MacBook and a remotely connected computer. VNC provides screen sharing and remote operation abilities allowing for easier status monitoring, debugging, and remote mobility. Remotely connected machines can also switch the robot into manual operation mode from automatic providing a software based emergency stop.

5.2 Sensory Systems

The sensory requirements of triMAXion are grouped by the team into three main categories: (1) detection of vertical obstructions, (2) detection of horizontal obstructions and white lines, and (3) knowledge of location and direction. The following sensors are used to fulfill each of these three identified requirements.

5.2.1 Hokuyo URG-04LX Laser Scanner Detection of vertical obstructions is performed with Hokuyo's URG-04LX laser scanner (LIDAR), shown in Fig. 6-left. The LIDAR provides a 240 degree field of view with a range of 20 mm to 4000 mm, and can be interfaced via RS-232 which has been adapted to USB via to interface with the MacBook.

Figure 6. Hokuyo URG-04LX laser scanner (left), Apple iSight digital camera (middle), Garmin GPS 17HVS (right)

5.2.2 Apple iSight Digital Camera For the detection of white lines delineating the course and horizontal obstructions triMAXion employs an Apple iSight digital camera, shown in Fig. 6-middle. The camera features autoexposure for automated shutter speed adjustment and autofocusing from 50-mm to infinity with a video capture rate of 30 frames per second. The iSight interfaces with the MacBook via a FireWire connection.

5.2.3 Garmin GPS 17HVS The Garmin GPS 17HVS, shown in Fig. 6-right, provides triMAXion with information about its location on the obstacle course and the current bearing for waypoint navigation. GPS 17HVS is a WAAS enabled for certified accuracy to within 3 meters and provides information in the standard NMEA 0183 format. NMEA 0183 format allows for easy integration with readily available C++ frameworks. GPS 17HVS has an RS-232 interface and is connected to the MacBook using a Sewell SW-1301 USB to Serial Adapter.

6. Software System

The design of triMAXion's software is a multithreaded application where different elements of the robot operate on separate threads. It includes seven threads: (1) main/dashboard, (2) GPS, (3) LIDAR, (4) camera, (5) path finding, (6) motion control and (7) AI. The code physically is split into two parts: (1) a library which handles the device controllers, the path finding, the motion control and the AI, and (2) a main application which combines all of the elements in the library with a graphical user interface (dashboard) and operator control. All software is written in C and C++ using open source libraries. Analytical models of the software algorithms, provided in Fig. 7, 9 and 10, show the overall structure of the software (Fig. 7) as well as detail on individual threads (Fig. 9 and 10). The convention that has been followed is: Rectangular boxes are actions to be performed, while diamond boxes are flow control based on a condition. Conditions include single case based true/false or the completion of groups of tasks. Dashed boxes represent the data of a thread and are used to share information between threads. Black boxes with white text are actions, similar to other rectangular boxes, but are detailed in separate sub-models. Dashed arrows represent the generation of a new thread or forcing a no longer required thread to rejoin the main thread.

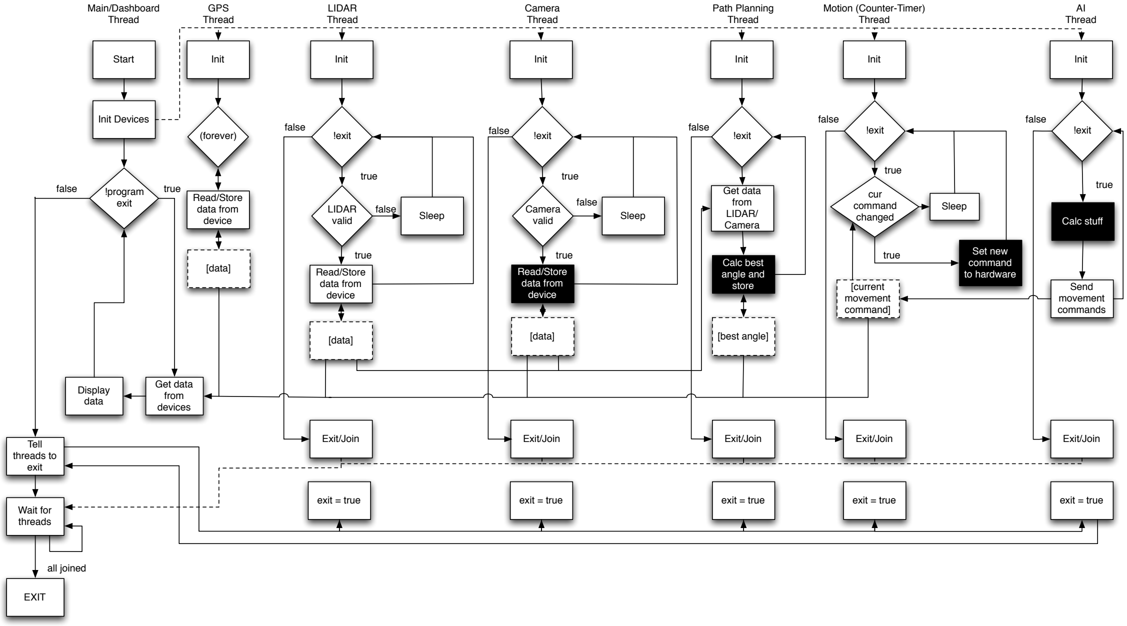
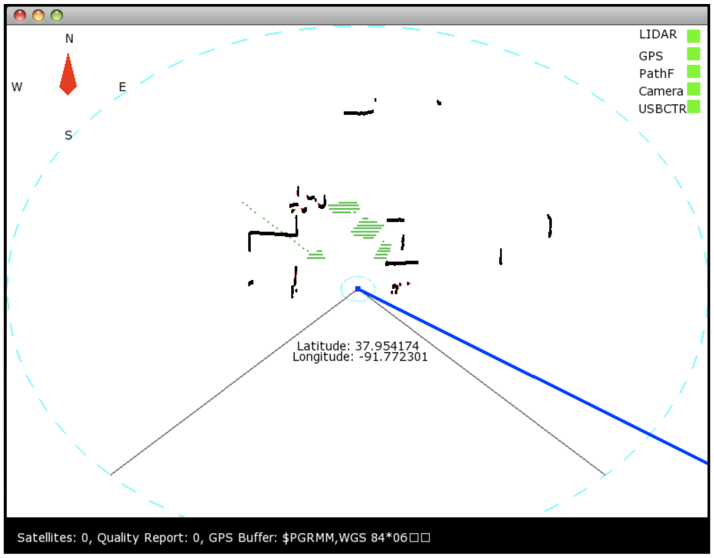


Figure 7. Analytical model of the IGVC thread layout

6.2 Main / Dashboard

The dashboard is the visual aspect of the main thread of triMAXion's software. The lifetime of all other threads is controlled by this thread through the signaling of thread detachment (initialization) and joining (termination). The dashboard feature, shown in Fig. 8, is generated using OpenGL for the graphical interface. It provides user feedback about objects detected with the LIDAR, white light detected by the camera, coordinates collected from the GPS, sensor and motion system status and the desired path for the robot. The main thread also handles mode switching from automatic to manual and controls the user input for manual operation.



Latitude: 37.954174
Longitude: -91.772301

Satellites: 0, Quality Report: 0, GPS Buffer: $PGRMM,WGS 84*06□□

Figure 8. Screen shot of the Dashboard

6.3 GPS

The GPS thread, detailed in Fig. 7, controls the USB-serial connection with the GPS device. The GPS thread runs the GPS continually once the serial communication is initialized. Within the thread, NMEA 0183 strings are read into the buffer and are translated into usable latitude and longitude data and the cardinal direction of the robot's motion. Information is also collected about the number of satellites currently being interfaced and the quality of the information being downloaded.

6.4 LIDAR

The LIDAR thread, detailed in Fig. 7, sends information about vertical obstacles into the serial-to-USB controller's buffer. This information is in the form of distances and sweep number for obstacle hits. Sweep numbers correspond to and are translated into the angles where obstacles are detected. The distances and angles are converted to x and y coordinates and stored in an array for access by other threads.

6.5 Camera

The camera vision thread, detailed in Fig. 9, uses the data from the camera to detect the white lines of the obstacle course. Once the camera vision thread is initialized, the FireWire camera turns on and autofocuses based on the ambient light. The OpenCV framework is used to capture frames from the live video feed, convert frames to grayscale, and then to black-white images. The white pixels in the image are analyzed to reduce the trapezoidal effect caused by the angle of the camera to a horizontal surface by considering the change in x and y on a horizontal plane for increasing distance from the camera. The image is then filtered to remove errant white pixels and is split into eight sections for line analysis. A least squares analysis is performed on each section to calculate lines based on the white pixels in each section. The white lines for the sections are recompiled into a complete image and stored in an array for path finding. When dashed lines are detected, the camera thread will essentially 'connect the dots' to make all course boundaries appear as solid lines. Once analyzed, the x and y coordinates of all white pixels are stored in a vector to be accessed by the other threads.

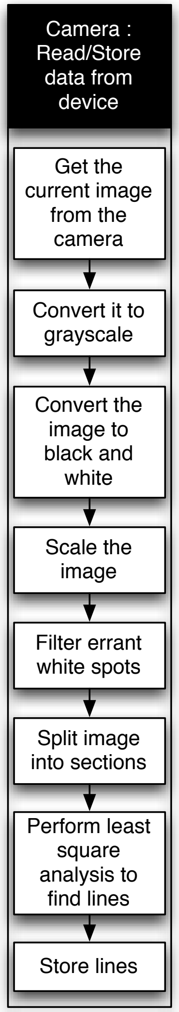


Figure 9. Camera thread algorithm

6.6 Path Finding

The path finding algorithm, detailed in Fig. 10-left, reads data from the LIDAR and the camera sensory threads for processing. The path finding algorithm uses ray casting to determine if objects or white lines have been detected that must be navigated around. The ray casting algorithm determines where a ray should be cast, if it will strike a line in the camera and/or LIDAR vectors, expands hit lines infinitely to determine where the potential intersection would be, and then performs a check to see if the intersection falls on the real line. If the intersection is imaginary (not on a real line), it is considered a miss; otherwise, it is considered a hit. An array is generated of missed lines or openings that the robot will fit through. Openings large enough for the robot to fit through are considered as potential paths. The potential path closest to the forward direction is chosen as the path of the robot.

Since the path finding algorithm finds the largest open hole, closest to center that triMAXion will fit through, it should be able to navigate around or through complex obstacles and remain in a forward direction. To navigate around islands, triMAXion will decide which path around is closest to the forward most direction and navigate in that direction giving precedence to the left most of two viable options. When navigating through switchbacks, the ray casting algorithm will give precedence to forward paths since side openings will appear considerably smaller due their apparent steep slope during the robot's approach.

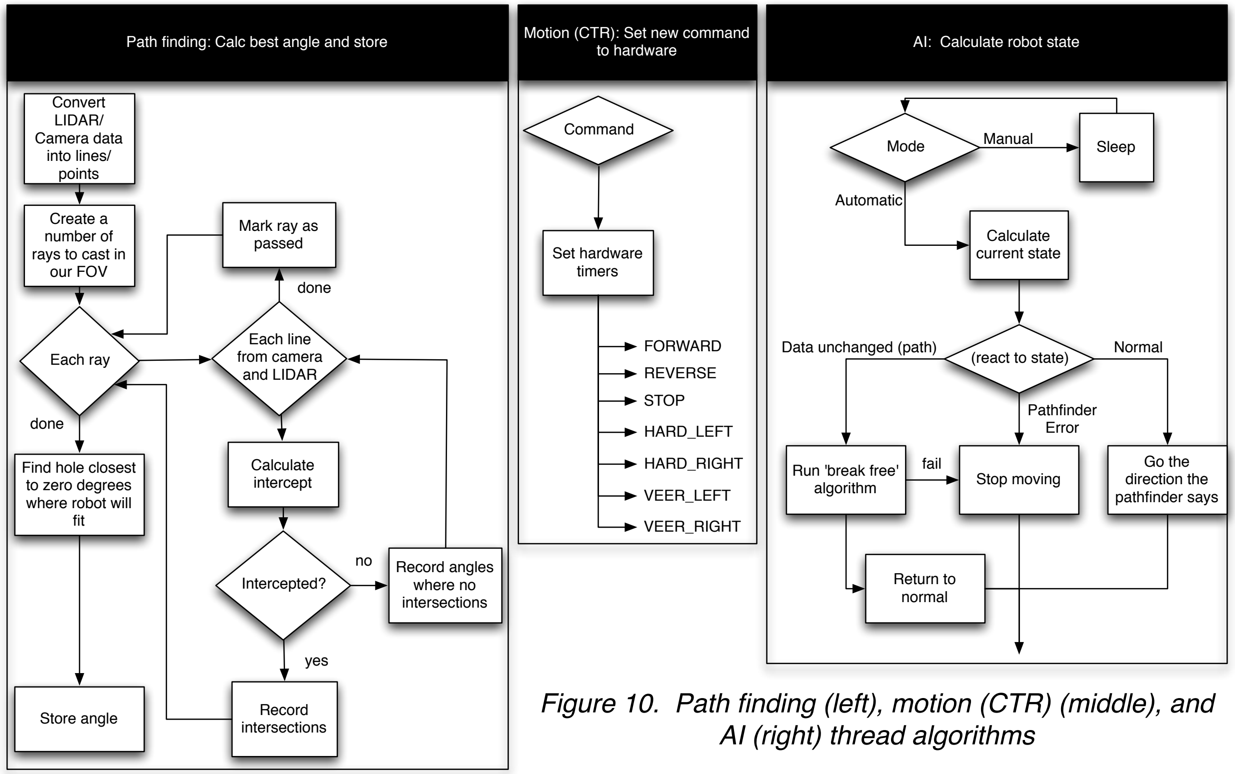


Figure 10. Path finding (left), motion (CTR) (middle), and AI (right) thread algorithms

6.7 Motion

The USB controller is the interface between the computer and the wheelchair motor controllers. This thread, detailed in Fig. 10-middle, has seven states which control the counter-timers (Section 4.1) to generate the pulses for the PWM required for variable speed, direction and turning. The seven states are: (1) forward, (2) reverse, (3) stop, (4) hard left, (5) hard right, (6) veer (soft) left and (7) veer (soft) right. The motion states are the same whether in manual or automatic operation.

6.8 AI

The AI thread, detailed in Fig. 10-right, is responsible for the overall control of the motion of the robot. The AI thread is based on a state machine using data from the other devices analyzed in each of the aforementioned threads to determine an internal state. The default state of the robot is basic forward motion, which consists of veer left, veer right, and forward. Basic movement follows the angle calculated by the path finder. If the AI determines that a failure has occurred that prevents it from entering the default state, it will try to return to its normal state. Examples of departure from the normal state include the path finder sending up an error flag or the robot failing to move after a given time period.

7. Physical Architecture

The physical structure of triMAXion is built on a Spirit mobile wheelchair chassis. The design, shown as a solid model image in Fig. 11, includes a box built where the seat of the wheelchair originally resided and an undercarriage mounting point. The box replacing the original wheelchair seat has two compartments, one for the payload and another for the laptop computer, the wireless emergency stop receiver, and the counter-timer. Also, at the front of the box, are mount points for the LIDAR and the digital camera. The undercarriage mounting point separates the high current electronics from the computing and sensing of the robot and contains the motor controllers, the 80A motor relays, the recharging circuit for the batteries and switches to toggle overall robot and motor electrical flow. Images of the completed triMAXion are in Fig. 12.

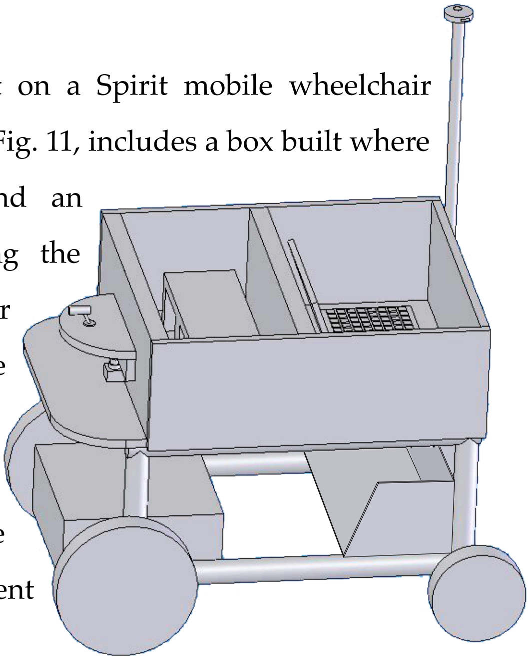


Figure 11. Solid model drawing of triMAXion

Figure 12. Images of the completed triMAXion

8. Performance

To evaluate the performance of triMAXion a series of tests were performed on the robot during its construction. Table 1 provides a comparison between predicted performance and the measure performance.

To test the velocity, triMAXion was run for a fixed time and the distance traversed was measured. The measured velocity turned out to be significantly below the manufacturer quoted velocity for the power wheelchair. The velocity has also been restricted due to high current concerns and sensor/computing time.

The LIDAR was tested by measuring the field of vision in a closed environment. The viewing angle has been capped so as not to mistake the front corners of the robot for walls or other vertical obstacles. The viewing distance has proven to be better than specified by the manufacturer.

The camera minimum and maximum viewing distance are a function of the camera viewing angle. The camera has been placed at a 15 degree angle from the horizon and can view white lines over a range of 57 inches starting 42 inches from the robot. While the team anticipated a loss of camera resolution when converting images to black/white, it was more drastic than anticipated.

Testing the E-Stop systems consisted of testing the wireless E-Stop and the stopping distance. The wireless E-Stop was measured by stopping the system at varying distances, until its maximum distance of 60 feet was determined; the wireless E-stop had an anticipated range of 50 feet as per the door bell's original specifications. The stopping distance of the robot was determined by running at full forward velocity and sending an emergency stop command. The predicted stopping speed was less that 1 foot based on initial testing with the wheelchair; however, with the slower speed of the robot, the stopping distance is less that 3 inches.

The final testing performed was with the laptop battery. The laptop was left running the software while communicating with the sensors until the battery was drained. This lasted between 45 minute and 1 hour during tests.

Table 1. Performance analysis for triMAXion

Tests Performed	**Predicted**	**Measured**
Velocity	4 mph	.5 mph (restricted)
LIDAR viewing angle	240 °	210 ° (restricted)
LIDAR viewing distance	157.5 inches	223.5 inches
Camera minimum viewing distance	36 inches	42 inches
Camera maximum viewing distance	180 inches	99 inches
Wireless E-Stop range	50 feet	60 feet
Stopping distance	<1 feet	<3 inches
Laptop battery life	1.5 hrs	45 min -1 hr

9. Cost Analysis

The following table provides a cost breakdown for triMAXion:

Table 2. Cost Analysis

	Number of Components	Cost per Component	Total Cost
Motion			
Spirit foldable power wheelchair	1	$1,300.00	$1,300.00
ACCES USB counter-timer	1	$249.00	$249.00
Victor 885 motor controller	2	$179.00	$358.00
PWM signal driver cables	2	$15.00	$30.00
Sensing			
R283-Hokuyo laser scanner	1	$2,375.00	$2,375.00
Apple iSight digital camera	1	$100.00	$100.00
Garmin GPS 17HVS	1	$199.99	$199.99
Computing			
MacBook laptop computer	1	$999.00	$999.00
USB hub	1	$26.29	$26.29
Sewell SW-1301 USB to Serial Adapter	2	$12.45	$24.90
E-Stop			
1.5″ red button	1	$26.87	$26.87
80 amp relay	2	$4.56	$9.12
Latching relay	1	$7.67	$7.67
Components	N/A	$5.56	$5.56
Heath Zenith wireless doorbell	1	$19.87	$19.87
Electrical			
Wire (feet)	30	$0.25	$7.50
60 Amp Fuse	2	$3.99	$7.98
DPDT Switch	2	$3.99	$7.98
XLR Socket	1	$5.75	$5.75
Wire Connectors	N/A	$20.00	$20.00
Box/Chassis			
Wood	N/A	$50.00	$50.00
Paint	N/A	$5.00	$5.00
Fasteners	N/A	$20.00	$20.00
18″X24″ clear acrylic	1	$5.97	$5.97
Total Cost			**$5,810.26**

10. Conclusion

TriMAXion represents the design goals of the team toward a modular robot design with an integration of off-the-shelf parts to create an IGVC robot with applicability toward multiple robotic tasks.