Stark 2012

University of Massachusetts Lowell

Team:
Jordan Allspaw, Ken Cramer, James Dalphond,
Michael McGuinness, Brigit Schroder, Baochen Sun,
Nat Tuck, Anthony Vardaro

Advisor:
Fred Martin

May 4, 2012

Contents

1 Introduction

The University of Massachusetts Lowell (UML) Robotics Team is proud to present Stark 2012, an update to our robotics platform designed for entry in the 2012 Intelligent Ground Vehicle Competition (IGVC). Stark is based on our mature hardware platform, which was designed in 2010 as a collaborative effort by the Mechanical Engineering, Computer Science, and Electrical Engineering departments at the University of Massachusetts Lowell. Our focus this year is on software innovation and improved robustness to handle the various requirmenets of the combined Auto-Nav challenge.

2 Vehicle Design Overview

The 2012 iteration of Stark is controlled by a software system built using the Robot Operating System (ROS). This control system, described in Section 3, is designed to handle the unique challenges of the Auto-Nav challenge while implementing simpler waypoint navigation to effectively handle JAUS control.

Stark employs a 4-wheel differential drive system for movement. Two motors control the two sides of the robot independently. The outer dimensions of the frame are 28 wide by 37 long. With the mast in the vertical position it stands 57 tall, and only 27 when the mast is collapsed for transport. The estimated weight of the vehicle including payload is approximately 200 lbs.

The electronics are isolated from the rest of the robot in a removable case that sits atop the chassis. Power is provided by two 12 volt, 50 amp-hour batteries, run in series to give 24v. The motors run on 24 volts, and a 24-to-12v DC-DC converter is used to power the computer system and sensors. Under ideal conditions the power system will last approximately 3 hours. The vehicle also has fail-safe and emergency stop systems integrated to keep it safe for operators and spectators. The electrical and mechanical systems will be discussed in detail in the following sections.

3 Software Control System

Our control system is built on the Robot Operating System (ROS)[1]. ROS provides a networked communications scheme using the publisher/subscriber strat-

---

[1] http://www.ros.org

egy. It also provides a number of pre-built components, such as simulators, visualizers, and relative geometric position tracking. ROS encourages a modular design by nature. Our full control program consists principally of the following modules:

Four Sensor Modules:

- A position estimation module which keeps track of the position and orientation of the robot is based on GPS, compass, and wheel odometry readings. This is necessary for mapping and determining if the robot has successfully hit waypoints.
- A laser rangefinder module which feeds data from a laser rangefinder to provide very accurate obstacle distance data in a plane in front of the robot. This module is provided by the ROS system.
- A line boundry detection module which uses computer vision to detect clear areas to drive in. This module recognizes painted white lines and simulated potholes for the purpose of avoidance, and is also capable of detecting ramps as a traversable surface.
- A flag detection module which detects nautical-style flags that signal the robot to avoid them on a specific side.

Five control modules:

- A navigation module ("Adaptive Waypoint Planning") which keeps track of which GPS waypoint the robot is traveling towards and a sequence of waypoints to visit in the future.
- A mapping module which builds a map of the challenge course. This is done using an occupancy grid approach with a fusion of the sensor modules discussed below.
- A pathfinding module which uses the map to find the best path to the next intermediate goal. This is accomplished by running A* on the occupancy grid of the map.
- A obstacle avoidance module which issues actual motor commands to safely transition the robot to the next intermediate waypoint.

- A JAUS control module, allowing the robot to be controlled by standard JAUS commands.

3.1 Line Boundry Detection

A central task for robots in the IGVC competition is to detect and remain within the boundaries of the competition course. These boundaries are mainly defined by white lines chalked or painted upon the ground. If the robot cannot successfully detect these lines it has no way to remain within bounds and compete. Detecting these lines must be done in real time, in a dynamic environment, as well as the fact that the input to the robot (a camera image of the field in front of the robot) is noisy and filled with many things other than grass and the lines to be detected.

The main strategy employed to isolate lines in the input images is a series of simple spatial filters. First we examine the overall brightness of the image, to sort the input into two categories: images in normal lighting conditions, and those taken in low-light conditions. Low-light images are converted into HSV and have their saturation and value channels normalized. When converted back into RGB, this gives white lines a bluish tint, making them easy to isolate by subtracting the green and red channels from the blue channel, and then smoothing and thresholding the result.

Images taken in normal lighting conditions have white lines show up with a high value, but a low saturation in HSV. Subtracting value from saturation therefore highlights the lines quite nicely, giving good results after smoothing and thresholding.

After lines are isolated the image undergoes a birdseye transform and is then segmented to produce an occupancy grid for merging with other maps.

3.2 Flag Detection

Flags are detected by an image processing technique based on blob-finding and ground plane projection. For each video frame a sequence of steps are performed to extract flags. First, the image is converted to the HSV color space. Then, the pixels are filtered for sufficiently red (or blue) pixels; such pixels have an appropriate hue and are above a threshold saturation. Next, a blob finding algorithm is run on the resulting image to extract the “objects” in it. Finally, the blobs are filtered by actual size based on a ground-plane projection. Anything that makes it through this filter is a flag. The same ground-plane projection is used to localize any detected flags.

3.3 Position Estimation

Stark has three sensors that provide pose information: GPS, rotary encoders, and an electronic compass. The compass provides near-perfect yaw data, but both the GPS and rotary encoders have known large errors. GPS error is estimated using a sequential Monte Carlo method, using compass and encoder data to make predictions and then using the actual GPS data to weight the predictions.

3.4 Mapping and Pathfinding

Stark's movement is driven by a system of dynamic mapping and pathfinding. Information from the laser rangefinder and the line detection module are each used to fill an occupancy grid showing the areas that have been detected by that sensor as unsafe to drive on. These separate maps are then merged to create a combined map that can be used for pathfinding.

Pathfinding is performed using modified A* search, operating on the combined occupancy grid. The occupancy grid is inflated at least the maximum radius of Stark so the search considers the robot's size. The resulting path is then trimmed based on map information so that the first waypoint in the path is the farthest waypoint in line of sight of Stark.

3.5 Obstacle Avoidance

Path following is preformed using a similar method to the 'Tentacles' algorithm. A number of possible motor commands are considered (and modeled) and the set with the most progress towards the intermediate waypoint without striking known obstacles is used to drive the motors.

3.6 JAUS Control Module

The IGVC rules require the implementation of a core set of the JAUS protocol, specifically, the JAUS Transport Specification (AS5669A), the JAUS Core Service Set (AS5710) and the JAUS Mobility Service Set (AS6009). This is useful because it allows standardized control software to give high level commands to Stark and also monitor robot status. The JAUS control elements are

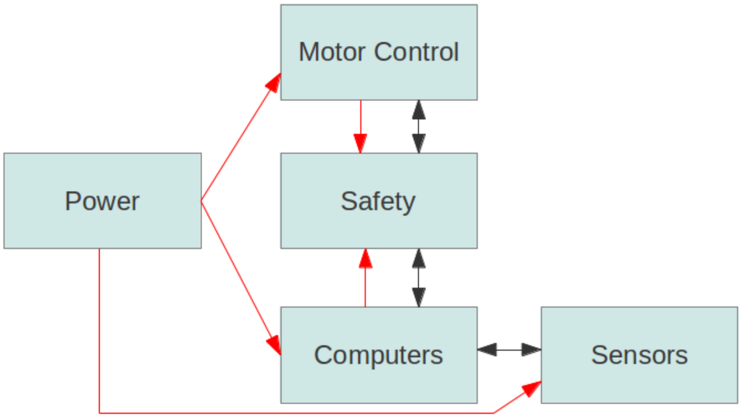

Figure 1: Block Diagram of the Stark Electrical System

implemented as a ROS node which monitors sensors to provide status information and forwards commands to other elements of the system to provide remote control and waypoint navigation capabilities as required for the IGVC.

3.7 Stage Planar Simulator

Stage simulates the physical robot in a 2-dimensional world which contains various sensor models that the Player server can connect to. The software team created a virtual version of Stark that ran in a simulated environment. The simulated robot contained most of the sensors the real robot has, excluding the camera and computer vision elements, and performs very close to how the real robot does in the real world. Stage is able to provide simulated position data to the GPS sensors on the simulated robot, making it very useful. Test courses were created and the team was able to run the control code quickly and safely without ever having to touch the physical robot.

4 Electrical Design

In the following sections the electronics system of Stark is explained in detail. There are several electronic systems working together to make the robot functional. These include power, computer, sensors, safety, and motor control systems. Below is a simplified block diagram of how the system is set up.

4.1 Power System

Stark uses a pair of 12v lead acid batteries to produce 24V for distribution throughout the robot. While the motors run on raw 24v power from the batteries, the computer and sensor systems draw power from a pair of high-efficiency Vicor DC-DC inverters. At nearly 95% effiecieny these inverters maximize battery utilization while providing the exact voltages needed regardless of battery state.

As shown in the Figure 1, the safety system gets power directly computer and motor-controller subsystems. The robot should always fail safetly. If the safety system drew power directly from the power system, in the event of a failure the robot could still drive without being able to emergency-stop. With power being drawn from both the computer and motor control systems, the safety system will be powered whenever the robot is capable of moving.

4.2 Charging Circuit

Battery charging is one of the major improvements we've made in the Stark hardware platform for the 2012 competition. Charging a sealed lead acid battery takes three steps. The first two steps charge the battery until it draws below a threshold current while the third lowers the voltage to keep the battery charged without overcharging it. Not overcharging a lead acid battery is important, as overcharging causes hydrogen loss and thus permenantly reduces battery capacity.

Charging a battery wired in parallel to a load will result in the battery charger never being able to detect the battery being charged, Our pre-Stark design, the MCP, didn't take this into account. This resulted in the capacity of its batteries being reduced to less than half of their initial capacity.

Our first solution to this problem was to design a power switching circuit that would sit between the computer and the two sources of power. To do this a Linear Technologies LTC4414 IC was used to sense when the charger was plugged in. When the charger is sensed an external power MOSFET is triggered, cutting the batteries off from the load. With the battery cut off from the load, it is possible to charge the batteries with one bank of the charger, and run the computer with the second bank. The charging circuit designed to solve these problems for Stark in 2011 is depicted in Figure 2.

This power circuit solved the problem perfectly in theory, but ended up being complicated in practice. We went through several revisions that a variety

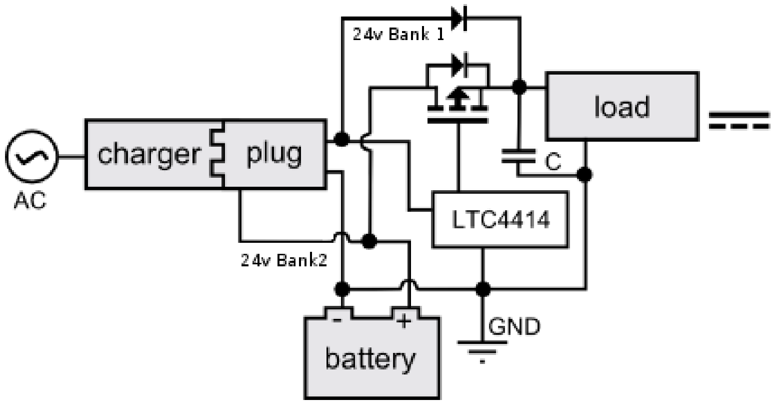

Figure 2: Old Stark Charging Circuit

of problems, including cooling the high current transistors.

For this year, we settled on a much simpler solution to the problem. Instead of using a three stage charger we instead use a two stage charger that does not use a high voltage, contant current stage. This slows charging slightly, but makes overcharging impossible regardless of load. This reduces the complexity of the system and removes a single point of failure from the robot.

4.3 Motor Controller Interface

The Motor Control Interface (MCI) board is a custom control circuit designed specifically to meet the requirements of the UMass Lowell IGVC robots. The MCI was designed to replace three separate systems, combining them into a single reproducible unit which could be easily replaced. This board is the main part of the safety system.

The MCI combines the Motor Controller Communications board, the Wireless Emergency Stop System, and the Main Emergency Stop System into a single unit. This new board acts as a single fail-safe point of contact to the Roboteq Motor Controller, both fulfilling safety requirements and maintaining a flexible interface for controlling the robot.

Having all of these systems combined into one point of failure allows us to make sure that the vehicle will fail-safe should anything happen to the radio controller or computer. Loss of power or communication between the MCI and any connected system will cause an immediate emergency stop.

For physical emergency stops, we are using C&K Rafi-x locking emergency stops which are within regulation for the IGVC. A single e-stop is placed at the rear of the robot mounted as part of the monitor. Since the monitor is the part that the operator will most likely be looking at while behind the vehicle. A second e-stop is placed on the front of the robot. If someone were standing infront of the robot while it was moving, this e-stop would be easily reachable. The two e-stops are wired in series in a normally-closed configuration; pressing either e-stop or a wire coming loose will open the circuit, causing an e-stop condition.

For the wireless e-stop, a 900MHz Zigbee tranciever was chosen. This radio system is rated to maintain communication within a six mile line-of-sight range. Additionally, the wireless controller periodically sends a heartbeat signal to the robot. If the robot does not recieve the signal as intended, it assumes the controller is disabled or out of range and engages the emerency stop. We believe that this is an important feature for a robot of Stark's size and power. The controller can also send an explicit emergency stop signal when the button is pressed, which causes an immediate e-stop. In either case when the robot is in e-stop mode a signal is sent to the controller so that the operator knows that the robot is safe to approach. At last year's competition, this system was proven to work from over a quarter mile away, well above the minimum required by the competition.

5 Sensors

Navigating the two courses at the IGVC requires the use of several different sensors. These sensors include cameras for vision, SICK laser for obstacle detection, and GPS receivers for positioning data. Although these are not the only sensors that can be used for the competition, they can be used to provide a good approximation of the world around the robot. In the following sections we will discuss each of the sensors and why they were chosen.

5.1 Laser Rangefinder

The single most expensive part this robot is the SICK laser. This sensor allows the robot to see 180 degrees in front of itself. Though we have experimented with several other range finders including Hokuyo scanners and the Microsoft Kinect, neither worked well in testing outdoors.

Figure 3: Dual GPS Recievers Mounted on Mast

5.2 GPS

In previous years, we found that the drift from a single GPS receiver can cause problems when trying to find a location. A solution to this problem is to use a Differential GPS (DGPS) system; however, these are very expensive and require a lot of power to run. A few of us hypothesized that given two different GPS chip sets, a Kalman or particle filter could be used to determine the position of the robot with higher accuracy. While this did not provide a drastic accuracy increase, it did provide high enough accuracy for the purposes of this competition. Stark uses two off the shelf USB receivers from Garmin and USGlobalSat to navigate. The cost of this solution is around $120 whereas DGPS systems used by other teams could cost upwards of $1,000. We believe that the cost benifits are well worth the loss of precision which can be compensated for in software.

5.3 Camera

One of the requirements of the IGVC competition is the ability to detect and avoid white lines on the ground. This task is done using computer vision techniques. Stark is equipped with a Playstation Eye USB camera. It is capable of streaming 640x480 images at 60 frames per second, and has automatic brightness adjustment to compensate for changing lighting conditions (e.g., clouds). The Eye is low-cost, available for as little as $15, which is inexpensive compared to high end cameras.

5.4 Compass

New for this year is the addition of a tilt-compensated compass from OceanServer. The OS5000 compass is only 1” square and weighs less than 2 grams fitting easily onto the sensor mast of the robot. With 0.1 degree resolution, and .5 degree accuracy even when tilted, this compass provides better heading measurements than any previously used solution including Devantech and Phidget compasses.

5.5 Encoders

Though our school has successfully competed in the IGVC for the past 5 years, no robot to date, including Stark has had wheel encoding. For this year’s competition Encoders from US Digital were installed onto the high-speed shaft of the NPC T-74 motors. With 512 counts per revolution and a 20:1 gear ratio, we get over 10,000 counts per rotation of the wheels.

6 Mechanical Design

The following sections describe a few of the main parts of the robot and their design.

6.1 Chassis / Drive Train

To cope with the extra forces caused by the wheels skidding to turn, a rigid frame was necessary. Extruded aluminum was used for all structural components due to its strength and light weight characteristics. The diagonal bars of the main frame rails serve a dual purpose. First, they provide diagonal strength to the frame much like the truss of a bridge, a theme that is carried throughout the design to reduce weight, add rigidity, and keep the robot aesthetically pleasing. Second, the diagonal bars give a convenient location for a tensioning sprocket for the drive train. A sprocket positioned at an optimal location restricts the idler slide to a single translational degree of freedom, effectively serving the function of removing slack in the chain. This also increases the chains contact on the gears, increasing the strength of the drive-train.

All components were placed to keep the robot perfectly balanced around its center of mass and low to the ground. The new chassis has been tested and will remain safe to a 35 degrees incline. Batteries, motors, payload, and all

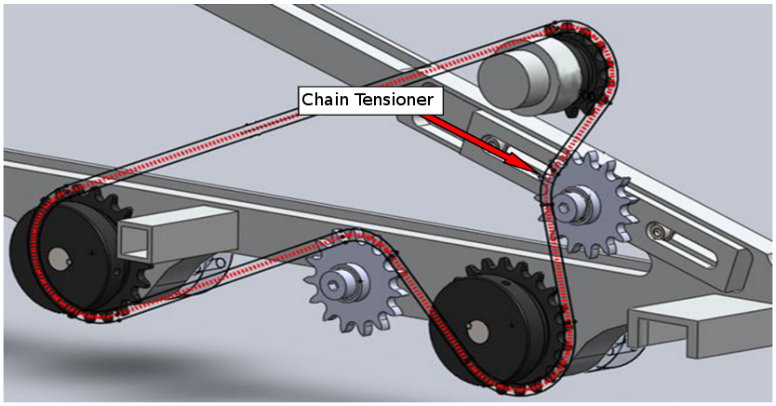

Figure 4: One Side of Stark's Drive Train

other components were modeled in SolidWorks allowing the center of mass to be precisely placed.

After the design was finalized, all heavy parts of the robot were analyzed to determine if weight reduction was possible. Heavy components were lightened by removing material where it would not affect the rigidity of the design. All attempts to lighten the robot were first stress tested in Solid Works to ensure a strong design.

6.2 Sensor Mast

The function of the sensor mast is to position one or more cameras at a high vantage point to maximize the robot's field of view. Minimum functional requirements incorporated into the towers design included the need to accommodate one USB webcamea at a minimum height of five feet. Since Stark was designed to be expandable several mounting points were made for the camera. Two cameras may now be mounted with focal points at known distances allowing for stereo vision capabilities or a single camera to be mounted in the center.

In addition to housing the cameras, the mast is a convenient place to mount additional sensors such as the two GPS devices and a single compass. Two lightweight mounts were placed on the outside of the mast keeping the two GPS devices at a known distance apart. The reason for two GPS units will be explained later. The compass mount in the center of the mast in addition to the camera mounts provides increased structural integrity keeping it rigid even

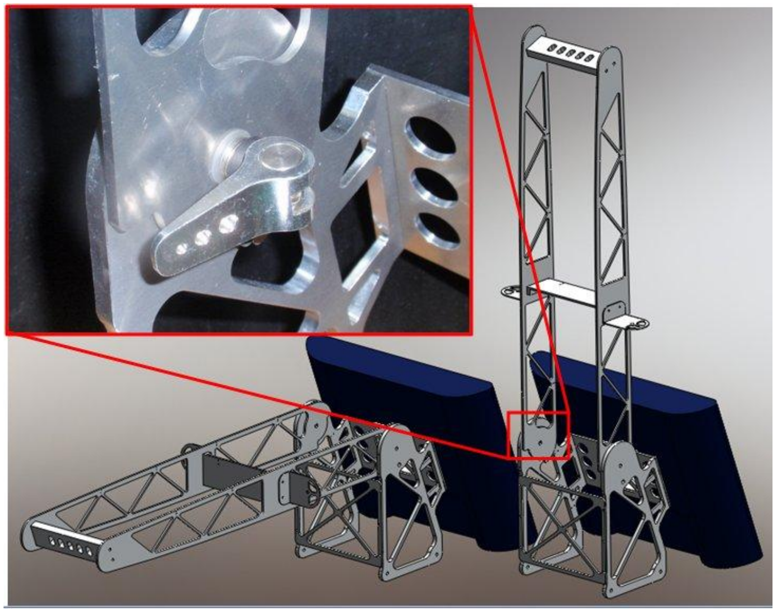

Figure 5: Sensor Mast, Upright and Collapsed

during vibration induced by uneaven terrain.

To make the robot easier to transport, the sensor mast can be folded down over the rest of the robot. When folded the robot 30 inches shorter without requiring electronics to be disconnected or removing any parts. This allows the robot to be shipped fully assembeld and is immediatly ready to go when out of its transport vehicle. Figure 5 shows cam locks that lock the mast in its upright or folded positions.

6.3 Electronics Case

One of the main design considerations for Stark's electronics was the ability to run in mildly wet weather. Because the competition will run regardles sof weather, the robot must be able to function in a somewhat wet environment. The solution was to place all the computer components in a vented case. To create air flow across the components, downward facing fans were placed at the back of the control case. All of these openings are protected by the shape of the control case allowing it to stay water resistant during light or moderate rain.

Another consideration was to allow the electronics to be easily removed from the rest of the chassis. This would allow the electronics to be worked on

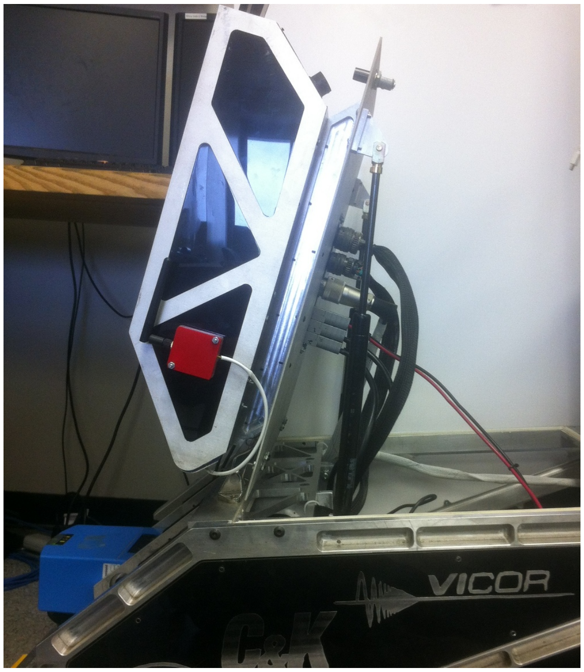

Figure 6: Electronics Case Upright / Open Position

independently of the mechanics of the chassis. Our solution was to mount the box on an access panel attached to the robot. The access panel is attached to the frame using quick release hinges. The hinges allow the panel to be removed yet give a pivot point for the access panel to open. To hold the access panel open, gas shocks were placed within the chassis. Amphenol and Anderson Polwer Pole connectors are used underneath the panel for quick disconnect of electronics. These connectors are water proof giving added protection within the chassis. Together, the pneumatic shocks, hinges, and connectors allow the control panel to be completely removed from the robot in under a minute. Figure 6 shows the electronics case in its open position.

6.4 Laser Mount

One of the design requirements for Stark was that the angle of the laser rangefinder could be adjusted. In previous IGVC attemps, the angle of the laser caused it to report false obstacles by picking up patches of tall grass. Using the SICK Laser's side mounting holes, a mount was made allowing 30 degrees of adjustment. The

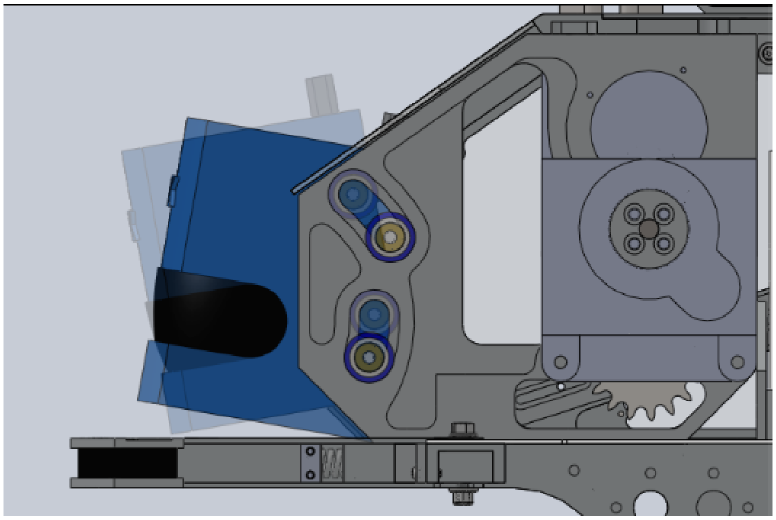

Figure 7: Mount for Laser Rangefinder

pivot point of the laser is centered at about the focal point of the internal lens allowing field adjustment of the sensor. Figure 7 depicts the movement of the device.

7 Conclusions

Over the past three months, the UMass Lowell IGVC team has worked to develop a new control system and to improve our existing Stark robotic platform. We expect that our entry this year will perform better on the Auto-Nav challenge than any previous entry and look forward to competing this year.