

AMOS III Design Report

RIT Multi-Disciplinary Robotics Club



Date Submitted: 5/25/17

Team Lead: William Fanelli

waf6826@rit.edu

Team Members: Sam Feine, Will Kommritz, Chris Culpepper, Evan Gallagher

sif6974@rit.edu, wrk5779@rit.edu, crc3510@rit.edu, ejg5704@rit.edu

Faculty Advisor: Ferat Sahin

feseee@rit.edu

Table of Contents

1. Team Structure and Design Process	3
1.1. Introduction	3
1.2. Organization	3
1.3. Design Process	3
2. Innovations	4
3. Mechanical Design	4
3.1. Overview	4
3.2. Frame Structure	4
4. Electrical Design	5
4.1. Overview	5
4.2. Power System	5
4.3. Motor Controller	5
4.4. Sensors	5
4.5. Safety Devices	6
5. Software	8
5.1. Overview	8
5.2. Position	8
5.3. SLaM / AMCL	8
5.4. Global Path Planning	8
5.5. Local Path Planning	8
5.6. Vision	9
6. Analysis	11
6.1. Cost Analysis	11
6.2. Performance Analysis	11

1 Team Structure and Design Process

1.1 Introduction

The Rochester Institute of Technology (RIT) Multi-Disciplinary Robotics Club (MDRC) is pleased to announce that we will be returning to the 25th annual IGVC competition. As with previous years, we will be using our AMOS III competition platform. We have added several new hardware and software innovations, including a revamped vision algorithm.

1.2 Organization

MDRC is a small, highly focused team of motivated students. Since the organization is multidisciplinary, each team member brings their own unique perspectives and talents. Tasks are distributed to the team members most passionate about working with or learning about them so that each member has a vested interest in the robot's success. This way they remain passionate and productive even through the most arduous tasks. To ensure that each team member's work became integrated with the collective effort, work periods were organized in which team members were encouraged to coordinate individual efforts into their larger modules, discuss design decisions, and to generate and receive feedback on new ideas.

Our main method for training less experienced members is to encourage them to participate in the collaborative work sessions we hold to design and debug the robot. We strive to maintain thorough documentation that is accessible to all members. This makes a strong starting point, enabling work on current issues and helping to provide insight on our design and troubleshooting processes. It also gives practical experience on how individual modules operate and interact with one another. From there, they can begin to provide their unique perspective to help overcome obstacles as they become more acquainted with the systems.

An overall project leader provides guidance throughout the months of development leading up to the competition by setting goals and deadlines for various subsystems. A clear goal for each subsystem is stated and consequent deadlines are set based on achieving this goal in time for testing. This year, our goals focused on simplicity in design and robust operation under widely varied conditions.

1.3 Design Process

Once tasks are assigned to the individual or group most interested in working on it, a Trac system is used to set milestones and assign due dates for the individual tasks. Trac is an online 3

project management system designed to keep software projects on target. Trac integrates with our subversion code repository and allows code check-ins to reference tickets with progress reports as work is being done. This makes it easy to make sure all tasks are progressing as planned, and to add man power to tasks which are falling behind.

2 Innovation

As a continuation from our entry in years past we have developed and improved upon the ROS control system that was implemented. ROS is a powerful tool which simplifies the integration process of sensors and software. Communication between nodes in ROS is greatly simplified over the interface of systems used in the past. Also, because all ROS nodes run as separate processes, a crash in any one node will not bring the whole system down.

Algorithm In previous years, our line detection algorithm has always been the weak spot in our robot's operation. Generally, our vision code ended up being quite complex and difficult to manage. This year our vision algorithm is simple, easy to tune for different conditions, and works quite well.

Experimentation has been performed this year on offloading the vision subsystem to a PIXY cam all in one camera and vision processing system.

3 Mechanical Design

3.1 Overview

The AMOS frame was constructed in-house by the club with the goal of creating a sturdy, modular, and weight-to-power efficient body to compete with. The base houses the main electronics such as the batteries, motors, and motor controller while the upper frame contains the higher-level electronics used for navigating its path. The upper frame also provides space for the required payload as well as for a laptop to rest on the top. It is also removable, allowing for easy access to the base frame.

3.2 Frame Structure

The AMOS base chassis was custom built directly to accommodate our design needs. It is made of aluminum square stock and steel angle stock, with diagonal elements to reduce flexing and custom slots specifically designed for our batteries and motors. Despite heavy rain, rough terrain, and extensive test runs, the platform has remained solid and functional for many years.

The upper frame was specifically designed with modularity in mind: the upper layers can be easily reconfigured for different purposes. For example, if the payload was to increase in height, we would only have to unscrew 4 bolts and move it up a few inches on the graduated support

poles. The lower chassis was designed with a slightly more rigid framework in mind as the components from the motor layer are rarely changed. The LIDAR mount is also modular in that it can be moved vertically and tilted with simple tools. We have used this ability to tweak the trade-off between range (by adjusting the angle) and angular resolution at small distances (by adjusting height). The upper and lower layers can be separated, which greatly facilitates transportation in vehicles. The only communications connection between the lower motor layer and the above layers is a USB cable, which uses a highly noise immune differential signal. The benefit of minimizing interfaces between these layers is that it allows simple physical disconnection as well as an abstracted, reusable motor layer for other projects.

4 Electrical Design

4.1 Overview

This year's entry of AMOS includes a simplified version of the electrical layout of years past. This allows the robot to function more reliably and can easily be fixed if any problems occur on-site.

4.2 Power System

The power subsystem is extremely simple and self-contained by design: it consists of four batteries, fuses, a relay for the E-Stop, and two switches controlling the computer and motor layer power. All battery charging is done off-board; this ensures we can quickly swap out batteries with freshly charged ones. All batteries are enclosed in the motor layer, pushing the center of gravity down towards the ground. Four batteries are used to power the robot: two are used for driving the motors and two are used for the top computing layers. Each battery is an AGM sealed lead acid at 12 volts. See the Performance Analysis section for battery life estimates.

4.3 Motor Controller

The motor controller used is a 2x60 RoboClaw which handles the low-level control of the robot's movement. Closed loop motor control is accomplished using PID and high resolution optical encoders on both wheels.

This controller, along with motors and control circuitry is in the lowest section of the robot. In keeping with the hardware layer isolation approach, the only connection between the motor layer and the above layers is a USB cable. The benefit of minimizing interfaces between these layers is that it allows simple physical disconnection as well as an abstracted, reusable motor layer for many projects.

Experience has shown that high resolution encoders provide much smoother PID control when properly tuned. Low speeds can cause problems with low resolution encoders due to lack

of data while the robot is slowly rolling. Our new encoder selection offers very high precision:

- Direct Angular Resolution: 180 ticks / revolution
- Linear Resolution: 5mm / tick

Though much of the linear precision is lost due to slippage on the terrain or slack in the motor assembly, the high direct angular resolution is beneficial for our control system.

4.4 Sensors

For near-field obstacle detection and avoidance we make use of a Light Detecting and Ranging module, made by SICK. A spinning mirror inside the device directs the laser over a 360-degree path (180 of which is visible through the device's shield). At each finite degree, the distance the laser travels is measured and the angle plus distance is returned as part of a sweep. This is done many times per second to produce a horizontal line of points out to a distance of 8 meters.

For absolute positioning, AMOS uses an off the shelf nmea GPS module. It receives signals from the standard GPS satellites, as well as Glonass. The gps can achieve decent accuracy anywhere in the continental United States.

For inertial measurements of the robot chassis, AMOS is equipped with a Sparkfun 9DOF Razor IMU. The IMU uses an ITG-3200 MEMS based three axis gyro, an ADXL345 three axis accelerometer, and HMC5883L three axis magnetometer. The unit runs a ROS compatible firmware that reports filtered measurements to the system.

AMOS is outfitted with a single commercial USB webcam. This camera setup utilizes a high-quality webcam to provide detailed frontal images for use in mapping. The webcam is used to get the images which are processed to detect the lines of the course.

4.5 Safety Devices

Our E-stop system was custom designed in a previous year.

Due to the similarities between the remote and local E-stop modules, a single PCB design is used in both. Both modules also share the same ATmega168 microcontroller and XBeePRO 900 radio modem. The remote module (handheld unit) is powered by a rechargeable RC battery which will last for approximately 4 hours of continuous use, while the local module is powered by the 12V power-rail from the robot's computer battery. Both units have a pair of LEDs for monitoring current state and observing error conditions.

The E-Stop system is power failure safe, if the traction / motor power source is interrupted the motor driver and contactor will both lose power and the robot will stop. bIf the computer / sensor power is interrupted then the power to the wireless receiver is also lost in turn

deenergizing the contactor coil cutting power flow to the motor driver.

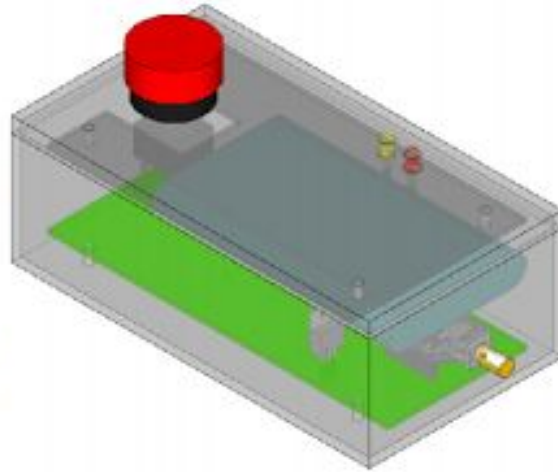


Figure 1: Emergency Stop Module 3D Mockup

When turned on, the remote immediately begins broadcasting the state of the E-Stop button many times per second. The XBee radio modem transmits these messages to the robot using a Frequency Hopping Spread Spectrum algorithm. Upon receiving these broadcasts, the robot checks the CRC provided by the XBee to ensure a complete message. If the message passes the CRC check, it notes the current time and checks the contents of the message. If the message contains the “STOP” command, or no valid message has been received for more than the 2.1 times the inter-broadcast delay, the robot will immediately disable the motors bringing it to a halt. Once halted, the robot will not re-enable itself until it has received the “GO” signal for more than 5 seconds straight without error.

The communication protocol used in our implementation is unidirectional, and as such the status display on both devices can only show the current “expected” state of the robot, rather than the actual state of the E-Stop relays.

In some of our implementations from previous year’s we used standard XBee-PRO modules which worked fine around our college campus, but failed on the competition grounds. After evaluating those failures, we identified several key features that must be present in our choice of radio-modem. The XBee-PRO 900 meets every one of our design requirements:

- Operates in a non-2.4GHz spectrum
- Provides good resistance to nearby signals and EM noise
- Reliable data-transport

The XBee-PRO 900, as the name suggests, operates in the 900MHz ISM band, and keeps us out of the Wi-Fi frequency range. Unlike most of the rest of the product line, this model 7

employs a frequency-hopping spread-spectrum algorithm which allows it to use a more powerful transmitter due to limited use of each individual frequency. This also provides a reasonable level of protection against noise in specific frequencies. Finally, the XBee-PRO 900 also provides an API which includes a checksum for all transmitted and received data, which allows the local module to trust every bit in the received message.

Some small improvements were made to this existing E-stop system. Previously, the custom PCB which was designed for the E-stop had a direct connection between the external antenna which provided our extended range capabilities and the Xbee. In our last competition, having the antenna connected to the Xbee resulted in a crack in the Xbee board after a few days of competition use, rendering our E-stop almost ineffective, except for a between runs repair. This year, the hand-held board was modified so that a flex connector could be inserted between the Xbee and the external antenna. This helps to ensure that any torque on the external antenna isn't transferred to directly to the Xbee.

5 Software

5.1 Overview

Our code base uses the open source ROS software. ROS is an arbitrated peer-to-peer architecture, this allows us to run algorithms on separate processes or even separate machines to offload computations as needed. Using already designed and built software allows the software team to focus on the algorithms involved with solving the challenges. Separate ROS programs are organized into packages, collections of related packages are organized into stacks.

5.2 Position

The ROS tf package is used to provide frame transforms for the robot. To allow for SLAM, and both local and global positions, four main frames are used: world, map, odom, and baselink. World is a stationary frame placed at the global origin. Map is the origin of the SLAM map and is placed at the starting position of the robot, the local origin. Map modifies Odom to perform localization, it is the origin of the positioning by the encoders. Finally, base_link is the static frame rigidly affixed to the chassis of the robot. Specifically the base_link frame exists on the axis that the wheels spin at the center of the chassis.

By asking for the transform from world \rightarrow base_link, the global position of the robot can be obtained. Similarly, by asking for the transform from map \rightarrow base_link the local position with localization is obtained. Finally, by asking for the transform from odom \rightarrow base_link the raw encoder orientation is obtained.

5.3 SLaM / AMCL

A ROS stack called gmapping was used to provide the SLaM capabilities of the system. The slam gmapping package takes the lidar output along with the robots odom \rightarrow base_link frame transformation to build the map. Localization is performed in real time by gmapping to keep the mapping algorithm localized. During normal operation the SLaM map is reloaded and requested by the adaptive monte-carlo localization node. This uses the same sensor and tf data as SLaM does to provide the most probable location of the robot in the map frame.

5.4 Global Path Planning

The global path planning system for the GPS portion of the course utilizes A* along with an exploded version of the SLaM map build by gmapping. The exploded map contains probabilities that ramp down from 1 \rightarrow 0 as you move away from the obstacles. The obstacles are also expanded so that in the new map the robot can be treated as a point to simplify the A* algorithm.

Waypoints are handled at the global level and are input to the navigation stack by the control node as navigation goals.

5.5 Local Path Planning

Local planning is achieved using the teb_local_planner package. This planner uses the timed elastic band method (TEB) of planning a path. Multiple paths are planned and the shortest time one is chosen. The planner acts differently than most planners which will dive in straight lines and turn on a point. The TEB planner will plan in smooth curves eliminating quick transitions in angular velocity which acts to stress the localization subsystem less.

5.6 Vision

A new vision algorithm was used this year. It is based on the very simple concept of combining different color channels into one gray-scale image. The gray-scale image is constructed as follows:

$$gray_{ij} = 2 * blue_{ij} - green_{ij} + hue_{ij}$$

The gray-scale image is then normalized to the range [0, 1].

The combination of the blue and green color channels allows for robust detection of the lines in the course. However, it lacked the ability to differentiate between the speed bump and the lines, the addition of the hue channel helps to alleviate this issue, because the hue of the white line and the yellow speed bump are quite different.



Figure 2: Initial Image Frame

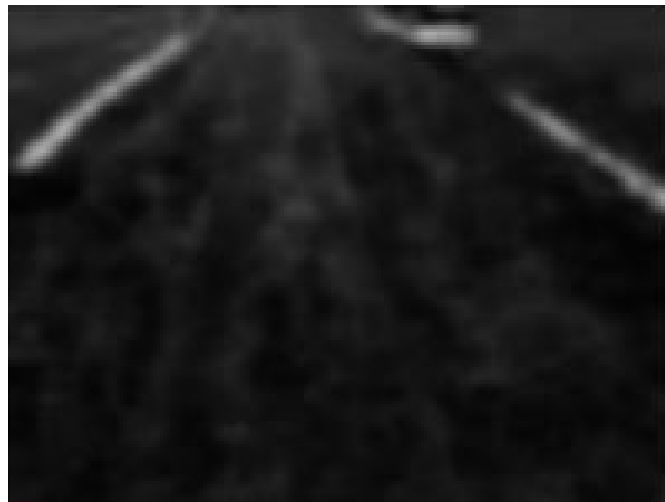


Figure 3: Gray-Scale Image Frame

The gray scale image is then passed through a Gaussian filter to eliminate small areas of noise.

The image is then split into a left and a right half to allow for the detection of lines on both sides of the image, the top part of the image is also cut off to avoid finding and responding to things too far into the distance. Each of these half images are run through an algorithm that marks the brightest pixel in each row, then in each column and builds a new binary image out of the data.

At this stage, the lines in the image are clearly visible and the Hough Lines algorithm can be run on each half to pick out the lines.

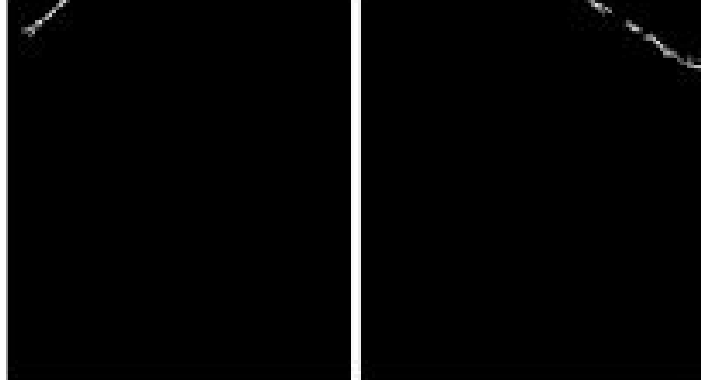


Figure 4: Left and Right Halves of the Image



Figure 5: Initial Frame with Lines Drawn

Finally, these lines are converted into a format usable by the local navigation system and SLAM and passed out to the rest of the system as a ROS message.

6 Analysis

6.1 Performance Analysis

Attribute	Design Goal	Final Product
Speed	5 mph	4.5 mph
Reaction Time	Near Instantaneous	100 ms
Battery Life	2 Hours (normal operation)	2 Hours
Obstacle Detection Distance	8 Meters	8 Meters
Vehicle Performance with Obstacles	Perfect	Acceptable
Navigation Point Accuracy	.2 Meter	1 Meter

6.2 Cost Analysis

Part	Vendor	Part #	Quantity	Cost (actual)	IGVC Cost
Frame Materials	Metal Super-Market	-	30ft	\$100	\$100
Frame Materials	Metal Super-Market	-	15ft	\$75	\$75
Misc. Mechanical	-	-	-	\$75	\$75
LIDAR	SICK	LMS-291	1	\$3000	-
dGPS	AgGPS132	1	1	\$500	-
Digital Compass	-	CMPS03	1	\$60	\$60
Motors	-	-	2	\$500	-
Main-board			1		
Misc. Connectors	-	-	-	\$40	\$40
Solid State Storage	Newegg.com	4 Gb Flash Card	1	\$35	-
Webcam	Newegg.com	-	1	\$50	\$50
Robo-Claw		2X60	1	\$200	\$200
PIXI-CAM			1	\$69	-
Total				4704	600