

Illinois Tech Robotics

7th Annual Robotic Lawnmower Competition Report

I. Introduction

- A. Design
- B. Team Organization
 - i. Roster

II. Software

- A. Motor Driver
- B. Sensors
 - i. Ultrasonic Range Finders
 - ii. Accelerometer
 - iii. Rotary Encoders
 - iv. Compass
- C. Sensor Integration
- D. Decision Making
 - i. Map Updating
 - ii. Pathfinding
 - iii. Motor Control

III. Overall System Specifications

- A. Costs
- B. Kinematics
- C. Dimensions
- D. Cutting Width
- E. Energy Source
- F. Safety, Reliability, Durability
 - i. Weather Resistance
- G. Autonomous

I. Introduction

IA. Design

Our design goals for the robot were focused on simplicity, efficiency, maneuverability, and cost. To keep the robot simple we based the frame of the robot around the frame of a push lawnmower and used a basic rectangular framework constructed out of welded steel to mount the wheels, drive motors, electronics, and sensors. We designed the robot to be as close to the width of the mower housing as possible so the robot could cut as close as possible to obstacles. To make the robot more maneuverable we used caster wheels for the front wheels. To keep the design of the robot within cost restraints we chose to use mostly inexpensive sensors instead of higher priced options such as LIDAR and GPS.

IB. Team Organization

- Mechanical team - Designed and built the frame and assembled the robot
- Electrical team - Mounted electronics and wired everything together
- Sensors team - Determined which sensors to use, determined where to place the sensors on the robot, wrote the code to read the sensor data
- Camera team - Developed the image processing code to convert web camera data to useful information
- High-level code team - Developed and implemented the high level algorithm and code to control the robot and react to sensor and camera events

IBi. Roster

- Marcos Arroyo - Sensors
- Allen Baker - Sensors, Camera
- Adrian Birylo - Mechanical, Electrical
- Zack Cornelius - Sensors
- Earl Fairall - Mechanical

- Allen Flavell - High Level Code, Mechanical, Electrical
- David Hann - Mechanical
- Chris Jones - Sensors
- Michael Ng - Sensors, High Level Code, Electrical
- Jesse Taylor - Team Captain, Camera, High Level Code
- Jesse Young - High Level Code

II. Software

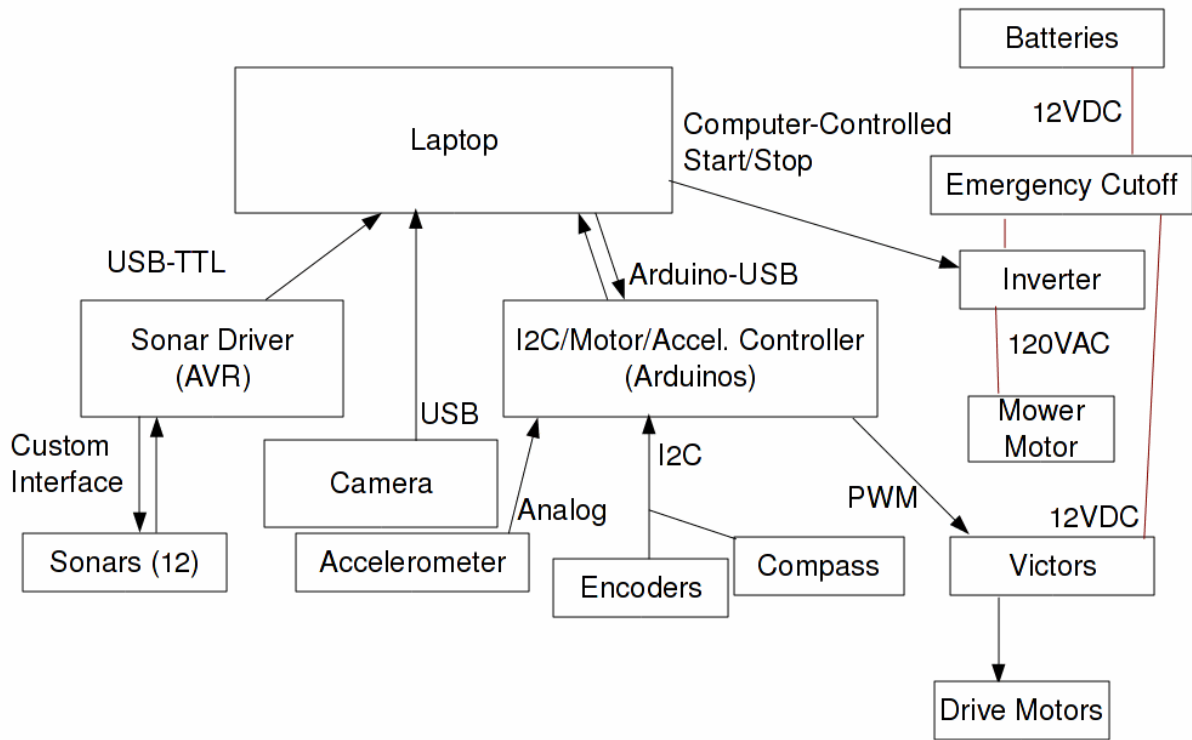


Figure 1: Hardware Control Block Diagram

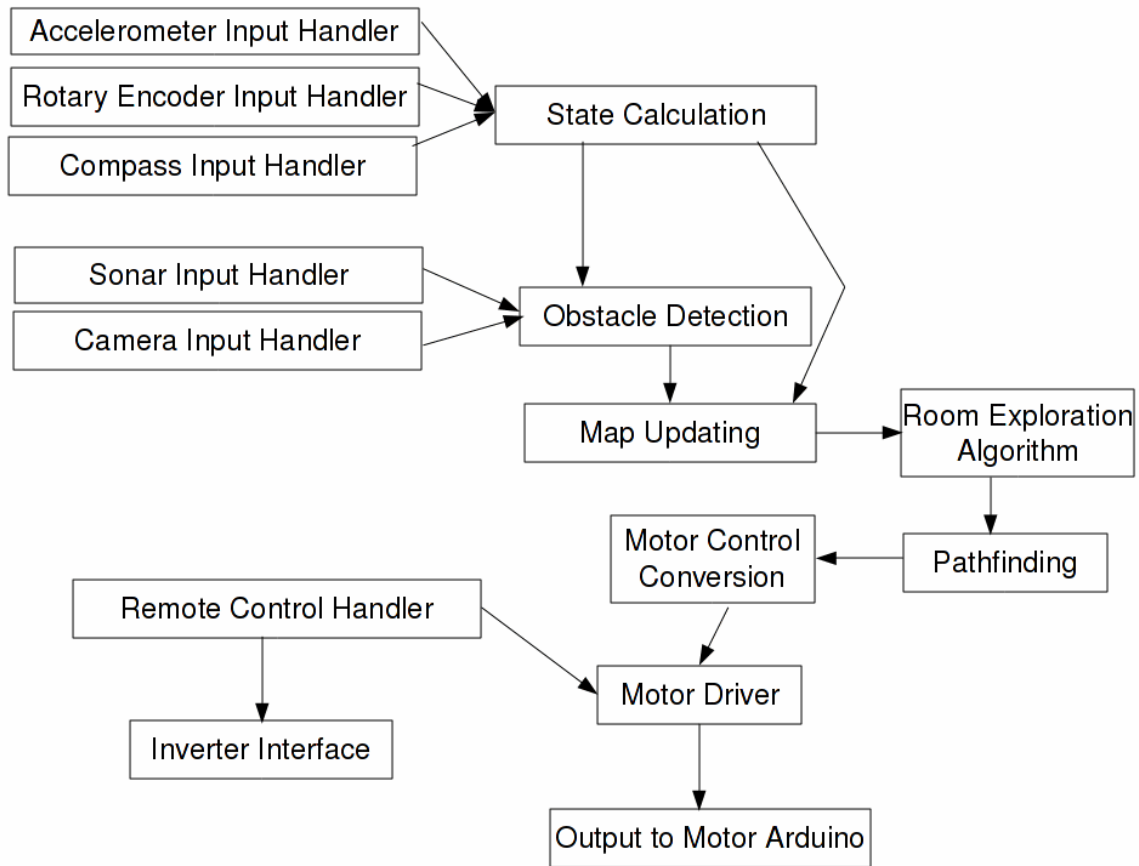


Figure 2: Software Block Diagram

IIA. Motor Driver

Driving our motors are the Victor 883 speed controller chips. These work by taking in a PWM signal and modulating it so that it is capable of powering and controlling an electric motor. We abstract this by having an Arduino generate these PWM signals from two integer inputs, one for the left motors and one of the right motors. On the laptop end, our motor driver detects changes in a pair of variables that are dedicated to the motor speeds, and outputs to the motor controller Arduino as needed. Thus, to control the motors, we only need to change these two variables.

IIB. Sensors

IIBi. Ultrasonic Range Finders

To detect obstacles before the robot collides with them we are using ultrasonic range finders positioned to face in each direction around the robot to maximize obstacle detection coverage. The ultrasonic range finders are controlled by a single AVR ATmega328 microcontroller. The ultrasonic range finders are fired in groups to avoid overlapping pulses.

IIBii. Accelerometers

We are using an ADXL330 Accelerometer to measure the robot's acceleration. This device is configured during wiring, and does not require any special interface. It simply provides analog output on its output pins.

We send this data to an Arduino, making use of its onboard ADCs. The Arduino then forwards the data to the appropriate input handler on the laptop. The input handler performs unit conversions from Gs to m/s^2 and also performs numerical integration. The output to the higher level code will be a vector representing the robot's velocity in m/s.

IIBiii. Encoders

We are using two Omeron encoders to measure the rotation of the robot's drive wheels. Each encoder is driven by a secondary wheel which are passively driven by the drive wheels. The rotation of the wheels measured by the encoders is converted into the linear distance traveled by each wheel based on the circumference of the drive wheels and ratio between the circumferences of the drive wheels and the encoder wheels.

IIBiv. Compass

We are using an HMC6352 compass module to measure the robot's heading. The compass module is connected to an Arduino and sends data to the Arduino using the I2C protocol when polled.

IIC. Sensor Integration

IICi. Camera Filtering

The image data from the cameras is passed through a custom color filter to mask out the grass and highlight lines and obstacles. Next an edge detection filter is applied. Then a Hough Transform is applied to correctly identify the painted lines on the field.

IICii. State Calculation

Our sensors contain a small degree of redundancy. This is intentional, as it allows for the sensors to correct each other (and act as a backup in case some of them fail). One of the challenges to overcome is how to aggregate all of the information together, and to decide which sensors have priority in case of conflict. The decision was made to use Bayesian Filtering to accomplish this task.

The general assumption is that the vast majority of the sensors' error will come from white noise. This greatly simplifies calculations, as almost all probabilities will be normally distributed. The difficulty with using Bayesian Filtering with our setup is with the rotary encoders. There will be a considerable degree of wheel slippage that should be taken into account with modeling. Furthermore, the relationships between much of our sensor data and the robot's physical state (position, velocity, acceleration) will be nonlinear.

IICiii. Obstacle detection

It is not enough for our sensors to detect obstacles, our robot must also know where these obstacles are located, compared to its own position. For the ultrasonic rangefinders, this consists of taking the sensor's position and orientation on the robot into account. For the camera, this is much more subtle and requires an estimate of the size of the obstacles.

IID. Decision Making

IIDi. Map Updating

As the robot operates, we will be constructing a map of the field. We begin with a blank map, which consists of a grid of cells centered at the robot's starting position. With every major sensor tick or milestone reached, the map gets updated. A map update consists of marking the region enclosed by the robot as having been "explored," as well as placing down any newly detected obstacles from the range finders or camera.

Because our instruments are not precise, obstacles are marked as regions where the obstacle could be. Furthermore, a given cell's state of being blocked is not a simple matter of being blocked or not blocked; there is a level of confidence associated with the status of a given cell. Multiple sensed blockages covering a single cell will cause that cell's confidence level to rise very fast. As the robot will no doubt be in motion, this allows us to build a picture of what the obstacles look like on the map, albeit a blurry one.

Each map update also causes the obstacle confidences to decay. High confidences decay much more slowly than low confidences, so regions where we are certain that there is a static obstacle are likely to stay that way.

IIDii. Pathfinding

After the room exploration phase, the robot has a new "goal" cell that the robot will have to cover next. The objective of the pathfinding phase will be to construct a path from the robot's current location to this "goal" cell. The difficulty arises in that there are obstacles on the field, and our robot must maneuver around them effectively. The way this is accomplished is by constructing a map overlay with an extra zone around every detected

obstacle. The width of this zone will be approximately half of the largest measurement of the robot. This overlay is updated every time the pathfinding phase is entered. If the goal cell is located within the overlay region, the goal cell is moved to the closest cell outside of the region.

On this new map, a standard pathfinding algorithm is applied to obtain the desired path. A* has been selected for this task¹. A side effect of using a standard pathfinding algorithm in this way allows for the previously-mowed cells to be given a larger weight, thereby making the robot more efficient in its task.

IIDiii. Motor Control

After the pathfinding phase, our robot's future motion is described in terms of a list of displacement vectors which, if followed, we are assured that the robot will not collide with any of the sensed obstacles. The motor control subroutine takes the current desired displacement vector, finds the relative angle, and converts this into values to send to the left and right motors using the appropriate trigonometric relationships and the desired "top speed," which at the time of this writing is set at 25% of the motors' maximum output.

1. Rabin, Steve. *AI Game Programming Wisdom*. Hingham, MA: Charles River Media, Inc. 2002. p.105-152

III. Overall System Specifications

IIIA. Costs

Part	Unit Cost	Quantity	Actual Cost	Retail Cost
Arduino	\$40	2	\$80	\$80
Battery Connectors	\$30	2	\$60	\$60
Battery Selector	\$34.56	1	\$34.56	\$34.56
Bondo	\$8.06	1	\$8.06	\$8.06
Cameras	\$31	1	\$31	\$31
Caster Wheels	\$23.36	2	\$46.72	\$46.72
Circuit Components	\$22.26	1	\$22.26	\$22.26
Compass	\$34.95	1	\$34.95	\$34.95
Computer	\$650	1	\$650	\$650
Craftsman Push Mower	\$214.99	1	\$0	\$214.99
Drive Motors	\$210.98	2	\$421.96	\$421.96
Electric Lawn Mower	\$249.99	1	\$249.99	\$249.99
Emergency Stop	\$44.83	2	\$89.66	\$89.66
Inertial Sensor	\$113.46	1	\$113.46	\$113.46
Motorcycle Batteries	\$65	4	\$269.57	\$269.57
Screw Terminal Shields	\$9.95	2	\$19.9	\$19.9
Tires	\$13.30	2	\$26.60	\$26.60

Ultrasonic Range Finders	\$33.96	5	\$141.31	\$169.80
USB to Serial Adapter	\$8.99	1	\$8.99	\$8.99
Victor 883	\$107.24	4	\$428.96	\$428.96
Wheel Bushings	\$9.51	2	\$19.02	\$19.02
Wire	\$11.98	3	\$35.94	\$35.94

IIIB. Kinematics

Our original specification was that the robot should go fast enough to be able to cover the entire field within the time limit. In addition, the rules specify that the robot's top speed shall not exceed 10km/hr.

Stress testing has shown that the top speed of our robot on asphalt is 10.8 km/hr, with a 167°/s turning speed. Since our robot is capable of exceeding the top speed for the contest, we believe that it is well within the minimum speed and turning speed specs.

Ensuring that the robot does not exceed the top speed specification is done in software during the motor control phase.

IIIC. Dimensions

Our robot measures 91cm long, 58cm wide, and 66cm high. The rules specify that the robot shall not exceed 2m (that is, 200cm) in any dimension. The robot meets this specification.

IIID. Cutting

Nearly a third of the cutting score our robot will receive comes from Zone 3 (the regions near obstacles). One of our specifications was that the robot's footprint in at least one direction should be very close to the size of our blade, so that we may be able to cut a large portion of Zone 3.

The safety of the blade is also a concern. Although it is a safe assumption that the field shall be reasonably free of rocks, this should not be taken for granted. We had to balance the safety of the blade with the quality of the cut, and decided that the blade shall be offset from the ground by about 2 inches.

The exact measurements for the blade are 48cm long, and it is offset from the ground by 5cm. So, the robot is capable of cutting up to a distance of 5cm from any obstacle, and the grass that it cuts will be about 5cm long. This meets our specification.

IIIE. Energy Source

The rules specify that the robot be powered exclusively by gasoline, batteries, or both. Our robot's motors are powered by several 12V Sealed Lead Acid motorcycle batteries, and everything else is powered by a Lithium-Ion laptop battery. Power for the blade motor is passed through a DC to AC power inverter since we are using a motor from an AC electric lawnmower. The energy sources meet the specification.

IIIF. Safety, Reliability, Durability

The rules specify that there be two means of stopping the robot in an emergency: remotely via wireless connection, and a 40cm in diameter red button. It is further specified that the robot shall stop within 3s and within 2m from the stop time.

Our robot includes a 40cm in diameter red button located on the rear of the craft, next to the batteries. Pressing it cuts all power to the motors. Testing has shown that the drive motors spin down in approximately 2 seconds and the blade motor spins down in approximately 8 seconds.

Our robot also includes a "stop" command in software. This command may be initiated wirelessly via remote connection from another computer. Initiating the stop command causes the drive motors to brake and the mowing motor to

lose power. Testing has shown that the drive motors stop within half a second and the blade motor spins down in approximately 8 seconds.

At the speeds the robot will be expected to be traveling, the robot will be expected to stop within 1.85 m or 0.46 m, depending on the mode of shutoff. This meets the specification.

In addition, our robot is safeguarded against collisions and mistreatment by its durable construction. All of the major structural components are attached to a welded solid steel frame.

IIIFi. Weather Resistance

Our robot is supposed to be "designed to operate in any weather condition." Our robot's most sensitive electronics components are insulated inside of a plastic shell. This not only helps protect the robot from shocks, but it also prevents adverse weather conditions for interfering with the robot's operation. Our robot meets this specification.

IIIG. Autonomous

One of the goals of the competition is to develop an unmanned craft. Although our robot has an alternate mode of operation which allows for joystick control, it is not part of the core design and this alternate mode is not active when the robot's autonomous code is running. Our robot meets this specification.