4/22/08
Eduardo Beltran
TAs: Adam Barnett
Mike Pridgen
Instructors: Dr. Antonio Arroyo
Dr. Eric Schwartz

University of Florida

Department of Electrical and Computer Engineering

EEL 5666

Intelligent Machines Design Laboratory

Final Report

GIR

Table of Contents

Abstract	3
Executive Summary	3
Introduction	3, 4
Integrated System	4, 5
Mobile Platform	5, 6
Actuation	6
Sensors	6, 7
Behaviors	7, 8
Experimental Layout and Results	8
Conclusion	8, 9
Documentation	9
Appendix	10…

ABSTRACT

This report covers the planning, design, construction, and testing of an autonomous robot. The robot, named GIR, takes objects and moves them around a building based on the weight of each object. GIR accomplishes this by sensing a weight and following the corresponding line that leads to its destination. Furthermore, GIR avoids obstacles in the path of the line and returns to home base after delivering the object.

EXECUTIVE SUMMARY

GIR delivers objects around a building through several behaviors. First, GIR senses the weight of the object that is placed on him. This is done through a load sensor. This strip of material is a force sensing resistor. A proper drive circuit allows for a reading of an analog voltage. The high resistance is decreased by weight, therefore a heavier objects creates a higher voltage. GIR's other behaviors are activated by the weight of the object.

GIR then delivers the object to another location in the building via line following. GIR does this with a line tracker composed of 3 IR sensors that face the ground. When the each IR senses a white color, its LED illuminates. The LED turns off when the IR senses black. The line tracker is used by coding the robot so that the middle LED stays off (middle IR over the black line). The robot moves to the right when the right LED turns off (the sensor is to the left of the line), turns left when the left LED turns off, and moves forward in a straight line if the middle LED is off. The robot looks like it wobbles around the line, constantly adjusting its position as it follows the line.

GIR also includes the behavior of obstacle avoidance. When an object is in GIR's path, GIR stops. It stays in the stopped position until the object is removed. This guarantees that GIR stays on the line. GIR avoids obstacles with the help of sonar sensors. With sound, these ultrasonic sensors determine the distance of an object in front of it. The two sonars are located on the front of the robot. The robot is programmed so that it stops when the sonars detect something roughly within 7 inches of them.

GIR is not only programmed to deliver an object, but choose its path based on the weight. GIR is programmed to "know what goes where". When GIR reaches its destination, it will pause for a few seconds so that the object can be removed. GIR then turns around to follow the same line back to home base. Therefore, aside from following a line, GIR turns in place in the absence of a line (area of just white – all IRs are on) and stops at a large black spot (are of black – all IRs are off).

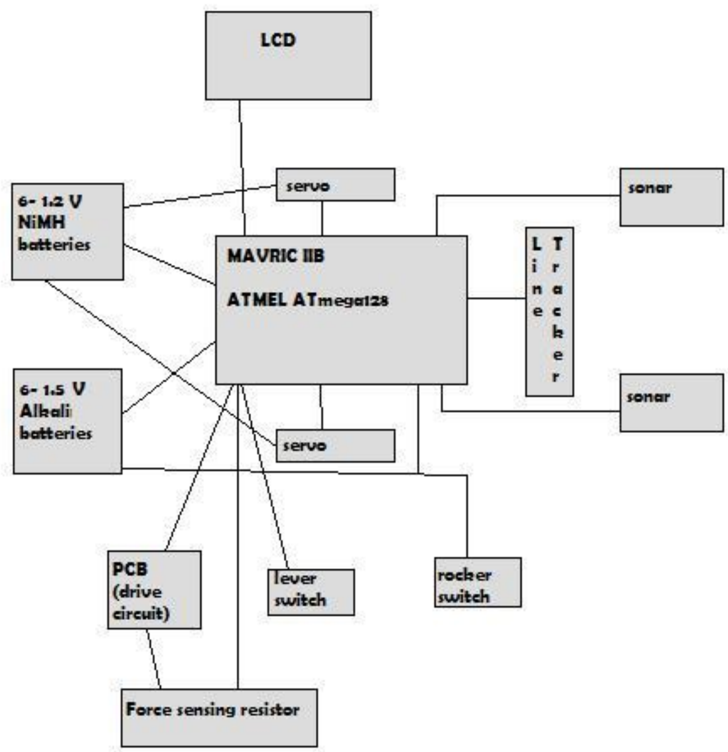
INTRODUCTION

In today's world, people seem to live on the go. People multitask and have busy lifestyles. A little help is needed once in a while. A mother needs help when she is busy with work and family. A studious student needs help when he or she is busy with homework and projects. The everyday couch potato needs a hand when he or she is too lazy to pass the chips or reach for the remote.

This is where GIR comes in handy. GIR takes the trouble out of sending things around a building. GIR can help hold things that a person cannot (or choose not to) carry and relocate. Businesses can use GIR for sending memos, packages, and many other items around the office.

GIR takes one less stress out of life and is perfect for the quick, on-the-go, give-me-now world.
GIR will take an object and take it to its destination. GIR will recognize the objects weight and follow its corresponding path. GIR will then come back home, ready to send out another object.
The following report describes the robot and its construction. The report includes details of its parts, behaviors, and results.

INTEGRATED SYSTEM

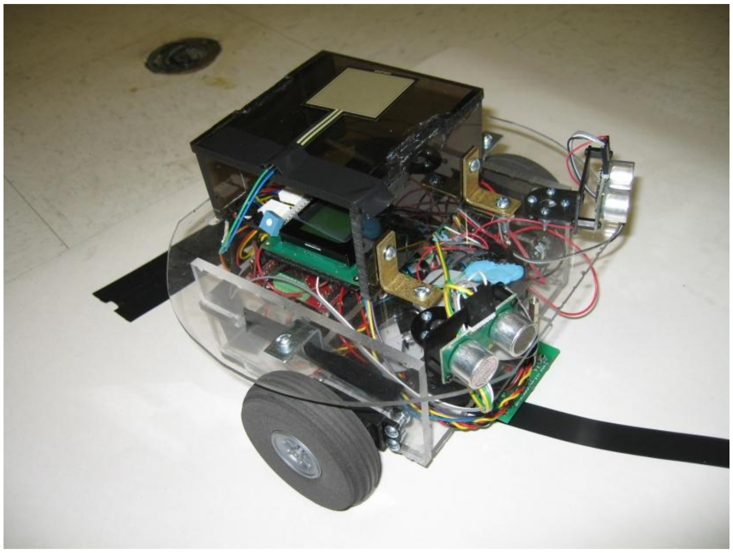


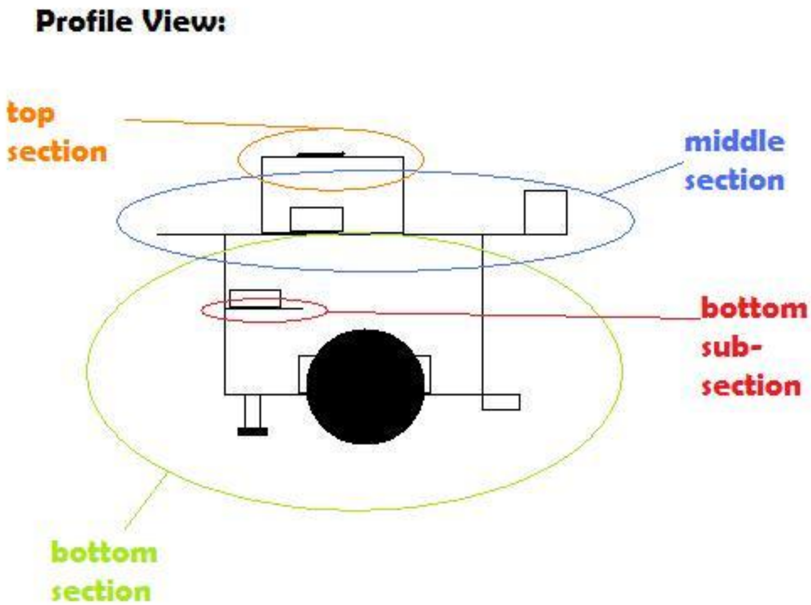
GIR's microcontroller is the ATMEL ATmega128 on a MAVRIC IIB board. The electrical components of GIR include the following:

- 1 MAVRIC IIB board with Atmel Atmega128 microprocessor
- 2 battery packs; one for the board and one for the servos (each holding 6 batteries)
- 2 HITECH servos HS-475HB
- 2 Devantech SRF05 sonar sensors
- 1 LCD display
- 1 force sensing resistor with PCB for drive circuit
- 1 Lynxmotion 3-IR line tracker
- 1 rocker switch (on/off switch)

- 1 lever switch (reset)

MOBILE PLATFORM





The mobile platform is made of Plexiglas. The platform is divided into three main sections. The first, or bottom, section is the area where about half of components and materials are housed. The line tracker, two servos, two chair gliders, and the two battery packs are located here. The bottom section is divided into sub-sections. The other sub section is a slide-in piece of plastic that holds the MAVRIC board. The slide-in (and out) feature was created so that the board can be easily removed for debugging without disassembling the rest of the robot. The top portion, the circular piece of plastic, houses the switches and SRF05's. The LCD is also located here as well, but is located under a platform of smoke-colored Plexiglas. This plastic portion makes up the third, top section. This plastic is used to not only protect the LCD, but also to create a flat surface to place the force sensor. This way the object for which the weight is measured can be measured more accurately and can stay away from the other components of the robot. The other materials used to construct the mobile platform include metal (i.e. angle brackets), and glue/epoxy (for binding plastics).

ACTUATION

The robot moves around using 2 HiTech HS-475HB servos attached to 3" Du-Bro Super Lite wheels. The wheels are closed cell treaded foam with nylon hubs. The servos provide more speed and torque than standard servos. This essential for the purpose of GIR, for it has hold objects as well as its own weight. The 3" wheels were chosen to create for speed than smaller wheels and to have a certain aesthetic proportion with the rest of the mobile platform. Two chair gliders that are attached at the bottom of the robot are used to provide stability and allow the robot to turn in any direction (like casters). The servos are powered by a battery pack of 6 – 1.2V rechargeable nickel metal hydride batteries.

SENSORS

The sensor that will be used for line following is the Tracker Ver 3.0 line tracking sensor from Lynxmotion, Inc. The sensor is made up of three reflective IR sensors. Each IR LED is paired with an IR sensor. Each LED is illuminated and directed to the surface where the line is to be detected. The LEDs on the tracker are initially on. Each output will go high when positioned over white and low when positioned over black. Each LED will turn off when it senses a dark spot, which in this case is the black line that the robot will follow. A comparator is used to clean up the signal. The sensor can be used from about .5" from the floor to almost touching the floor. The sensor seems to works well in normal ambient lighting, so a shield in not necessary (unless needed in the future after further testing).

The way the line tracker is used is to keep the center sensor on the line (the center LED should stay off). When the robot runs off track, the sensor will detect the line on one of the other IRs. The robot will be programmed so to move so that it keeps, or tries to keep, the center IR on the line. If the middle LED is off, then the robot is running in the right direction. If the left LED goes out, then the robot has moved off to the right, so the robot will go left. If the right LED is off, then the robot is on the left side of the line and must move to the right. The robot will be also programmed for two other situations: (a) when the robot loses the line (all LEDs are on) and (b) when the robot finally reaches its destination. When the robot gets lost, the robot will go into a "random movement" mode where the robot will move randomly, preferably in circles, until it finds the line again. The robot will also be programmed so that it will stop at the destination. The robot will be programmed so that when all the IRs sense a black surface, the robot will stop and wait for the object to be removed. A large black circle will be made at the end of the path to ensure this.

The sonar sensors that will be used for GIR are Devantech SRF05 Ultrasonic Range Finders. The ranger works by transmitting a pulse of sound that is outside of the range of which a human can hear. The pulse travels at the speed of sound (about 900 ft/s) away from the ranger in a cone shape. The sound travels back to the ranger after it reflects off an object in its path. When the sonic wave is transmitted, the ranger pauses for a brief moment and waits for the sound to be reflected in the form of an echo. The controller driving the ranger requests a ping, the ranger creates the sound pulse, and waits for the echo. If received, the ranger reports this echo to the controller and the controller can then compute the distance to the object based on the elapsed time.

The SRF05s require some calibration, for the listed range is not accurate. The datasheet states that the SRF05 has a range of 1-4000 cm. The true range falls somewhere in between but falls short of the maximum. Also, when detecting items at a distance close to the maximum range, the sensor loses a lot of accuracy. This is not only due to the built in range but also the other factors the come into play when working at 40 m, including interference from other objects, the angle of the surface relative to the sensor (objects not perpendicular to the sonar will not reflect properly), and the change in the sound wave (larger and weaker). To acquire the most accurate results, one must initially test the range of the sensor. To do this, one must an object at different distance from the sensors (preferably in constant increments). After measuring the output of the sensor, one can graph the relationship of the data and use it for calibrating the sensor.

The final sensor is a force sensing resistor. This sensor is a component of variable resistance. It initially holds a very high resistance, but a load on the sensing area reduces the resistance of the sensor. A drive circuit, for which there are numerous, is used to convert the resistance into analog voltage. A load increases the output voltage of the sensor. Therefore, a higher load produces a higher analog voltage. This way, there is a direct proportion between voltage and load. Calibration is needed for this sensor, especially because of the amount of noise in the reading. This sensor needs to be “broken in” to read some of the higher loads.

BEHAVIORS

GIR features 3 basic behaviors: line following, obstacle avoidance, and weight sensing. Using these behaviors, GIR will sense the weight of an object and, based on the range the weight falls on, send the object to its destination. There are 3 paths. The paths correspond to the lightest weight, the middle weight, and the heaviest weight. Two or three objects can be used, depending on the code. The robot can be programmed so that the lightest weight is no weight, where no object is placed, and therefore only 2 objects are used.

The way GIR works is best compared to a driver that runs into a fork in the road. In this case there are three possible routes, or lines, which GIR can follow. Based on the object placed on GIR, GIR will decide which route to take. A moving command with a time delay is used so that GIR makes the correct initial movement before following a line. For a short moment, GIR moves in the direction of the correct route. After this movement, GIR goes directly into the line following mode.

GIR is programmed so that it stops when the line tracker detects black on all three IR’s. Therefore, GIR stops and starts at the same spot (a.k.a. home base). When an object is placed on

GIR, the force sensor sends an analog voltage to the MAVRIC. The board reads this output and the LCD displays which path this output corresponds to. For example, if the heaviest object is placed, the LCD delays for a few seconds and then displays "Path 3". GIR then makes the initial movement in the route's direction. In the case of Path 3, which is to the left of the starting point, GIR turns momentarily to the left in order to leave the starting point and line up with the third path. GIR follows the line until it reaches the end of the line, where the line tracker detects an all white surface. GIR stops at this moment for a few seconds so that the person at the end on this line can remove the object from GIR. After this short delay, GIR turns in place until it detects the line again. GIR follows the same line but this time towards the start position instead of away. When GIR detects the all black spot, it stops. The task is done here. This procedure can be repeated for the other objects.

EXPERIMENTAL LAYOUT AND RESULTS

Before testing the main program, each individual behavior was tested. The easiest to test were the sensors and the LCD, for almost any output was a sign of a working component. The hardest part was testing the components with the servos. Most of the problems encountered came from this part of project. Constant adjustments had to be made in the code and the demo area.

The line tracker worked well at first, but soon GIR encountered problems when following sharp turns. The servos were spinning too fast. The speed of the servos was reduced to fix this. Also, the initial code for the obstacle avoidance did not work well. At first, GIR was programmed to avoid obstacles when outside of the line. This caused contradictions in the code and the line following behavior could not work either. Therefore, the code was changed so that GIR would avoid obstacles around the line while following the line. As GIR follows the route, if an object is in its way, GIR will stop until that object is removed. The newly fixed obstacle avoidance worked perfectly with the line following behavior.

The other major change was made with the force sensor. At first, a Flexiforce sensor was used until results showed the inconsistency of the sensor. Not only was it not consistent, but it could not hold the value of the force. This was mostly due to the small sensing area of the Flexiforce (a circle of a diameter .375"). Most, if not all, objects with considerable weight were much bigger than the sensing area, so the weight could not be focused on the sensing area. The Flexiforce was replaced with an Interlink sensor; a sensor with a 1.75"x1.5" rectangular sensing area. This sensor had more consistent results and was easier to calibrate.

When first testing the robot, several white sheets of poster paper were used with electrical tape. This cause many problems, including the gilders accidentally catching the edges of each new sheet. The main problem was that the paper would fold and crease. GIR would mistakenly see these areas as all black areas. Therefore, a PVC board was used for testing as well as cardboard display boards.

CONCLUSION

The most successful parts of GIR are the sonars and the line tracker. The line tracker had constant results. The sonars had a lot of noise but did very well in detecting and object that was close to the robot. Since obstacle avoidance did not depend on pinpointing the specific distance of the obstacle, the sonars were enough to detect an object at a relative distance.

Some limitations of the robot came from the servos. Since the servos were hacked, they were not identical in application. The speed had to be adjusted constantly in the code to find each servo's stopping point as well as the value to produce the same speeds in the same direction. The limitations from the servos also affect the behaviors of the robot. Line following was affected the greatest by the servos. Too fast a speed does not allow the robot to accurately and precisely follow a line, especially big curves. The sharper the curve, the slower the servo speed would have to be if the robot were to successfully track it.

The greatest limitation came from the force sensor. The first sensor used – Flexiforce – had too much noise and needed constant calibration. Also, the sensing area was inconveniently small. It was not accurate in reading the weights of objects with much greater cross-sectional areas. Aside from the calibration, a piece of plastic was needed to be placed on top of the Flexiforce sensor to increase accuracy (it helps focus the load to the sensing area). For these reasons, the Flexiforce was replaced with an Interlink FSR (Force Sensing Resistor). It worked the same way as the Flexiforce, but provided a much greater sensing area. It also had much less noise when it came to data output. Overall, for either sensor, the output values were never consistent. The limits made in the coding had to be changed several times to have the robot successfully complete all its behaviors. This is because of the type of sensor used. It is best to imagine these sensors as a foam pad (like memory foam). If not enough time is given for the object to sit on the sensor, the sensor will not fully sense the load. Also, if another object is placed too soon after an object, the sensor is still trying to return to its zero position. Even if calibrated and used perfectly, the sensor still holds up to a 10% error and is read on noisy analog values.

If I were to remake GIR, I would try using a different force sensor and add more components. The type of sensor I would use would be a load cell, like the ones used for digital scales. This sensor seems to be a more widely used in industry. As for other components, I would try using an mp3 playback unit to play sounds to alert the person at the end of the path to pick up the object. I would also try using some IR's for more advanced obstacle avoidance. Furthermore, aside from electrical components, I would try using some other material in making the mobile platform (i.e. Lexan, metal, etc.).

Advice for future students would include wiring and soldering correctly and cleanly, planning a descriptive design, and following datasheets.

DOCUMENTATION

Tracker Ver 3.0 Users Manual TRA-01 Ver 6.0 from Lynxmotion, Inc.

SRF05 users manual from Acroname, Inc.

APPENDIX A: Source Code

The code for the main program was made with the help (either in person or through sample code) of Dr. Arroyo, Adam Barnett, Mike Pridgen, and Angel Nunez.