Island Escape

Design Report

Mechatronics EN.530.421 (1)
Submission Date: March 25, 2012

Team Hot Rod

Robot Name:
Jonathan Taylor Thomas

Group Members:
Kyle Rohrbach
Adam Merritt
Cassie Tarakajian
Shing Shin Cheng

Introduction

The goal of this project is to build and test a robot capable of completing the “Island Escape” final project for Mechatronics. The course itself is 96” x 144” rectangle which has an 8” “lava” stream running through the middle, separating the start zone from the finish zone. The robot must pass over this lava stream to reach the end of the course. Aspects we will be assessed on are navigation of the course (e.g. staying in bounds, not touching lava, finishing in final zone), ability to pick up and move two cubes laid out on the starting side of the lava, ability to sense color of cubes (light or dark), and ability to deliver light cube into the goal zone first. To accomplish this, we will use knowledge gained from class to help aid in line following, distance measuring, motor control, and state machines. When brainstorming our design for this robot, we must ask ourselves questions such as, “How does the robot find the boxes?” or “Is it possible for the robot to jump the lava pit, or should we design a bridge for the robot to lay down?”

There are many different modes of locomotion for a robot to transport itself from place to place. Two such subsets are wheeled and legged locomotion. In legged locomotion, a robot uses (usually an even number) of legs to move around. These legs are designed in the style of human, mammalian, insect, etc. legs. They can have different gaits as well: alternating tripod, which is similar to the way a horse or dog walks, and wave gait, which is similar to the way a centipede walks. There are also robots designed to slither like a snake, or move like a war tank (wheeled tracked). Wheeled locomotion is good to use because it is simple: when using two wheels, there is a simple equation to convert the two motor velocities to robot position. Controlling legs on a robot is similar to controlling robot arms, except there are a lot of legs that need to be controlled in parallel, which complicates the problem of motion very quickly.

Given the two velocities of the motors, we can calculate the position and orientation of the robot in the following manner:

$$\dot{g} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}\cos(\theta) & \frac{r}{2}\cos(\theta) \\ \frac{r}{2}\sin(\theta) & \frac{r}{2}\sin(\theta) \\ -\frac{r}{L} & \frac{r}{L} \end{pmatrix} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix}$$

Where r is the wheel radius and L is the length of the wheel axle. We can use this to calculate the position *g* of the robot, time step by time step, via:

$$g(t + \Delta t) = g(t) + \dot{g}\Delta t$$

Where *t* is the current time and *t* is the time step interval. Given the motor velocities and position at any time, one can calculate the the robot position at the next time step.

It may seem like the easiest thing to do is figure out the exact path we want the robot to follow in terms of a series of *g*’s as a function of *t*; however, this is a complicated problem involving a lot of difficult calculations. It would be possible to do this calculation ahead of time, and program the robot with the series of $\dot{\phi}_1 \; \dot{\phi}_2$

In the real-world, this is not useful due to environmental errors, and therefore we must use a feedback-control system. We want to have some sensors on the robot in order to have a frame of reference as to where the robot is at any time. One such solution is to use a technique called line-following. In this scheme, a dark line is placed on the light-colored ground (or the colors vice-versa) as the trajectory that one wants the robot to follow. One then attaches infrared sensors to the bottom of the robot to determine where the robot is with respect to the line.

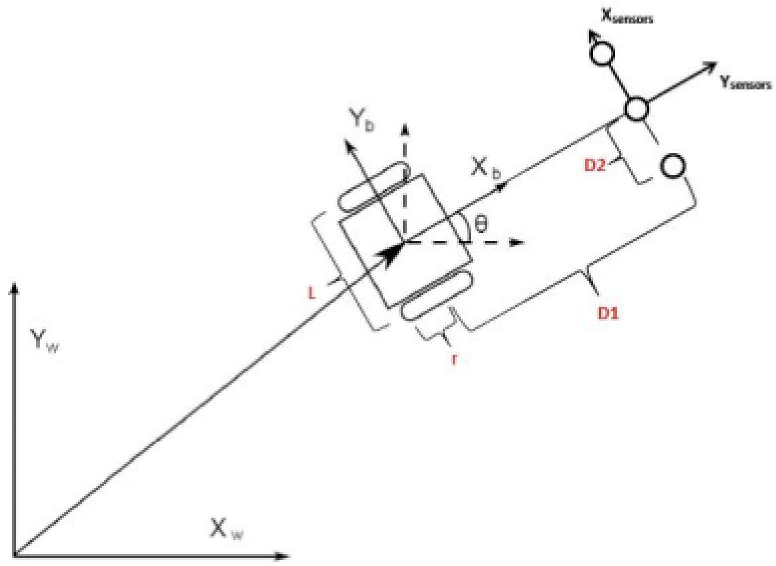

Figure 1: Robot and sensor orientation example, taken from Mechatronics 530.421 Spring 2012 Homework 4: PID Control of a Kinematic Cart by Gregory Chirikjian

Depending on which sensor(s) are over the line, the control system of the robot can determine at any time step whether the robot should keep its same trajectory and not alter the motor velocities because the robot is centered over the line, or change the motor velocities because the robot is moving off the line. The amount we alter the motor velocities is determined by the error: the difference between the desired position of the robot and the current position of the robot with respect to the line. The further the robot is off the line, the larger the error. For example, in a three sensor system as shown in Figure 1, if the middle and left sensor are over the line, the error is smaller than if just the left sensor is over the line.

One highly respected way to control the motor velocities is called PID (Proportional, Integral, Derivative) control. In this method, the motor velocities are altered each time step with a term $u(t)$:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Where $e(t)$ is the error, Kp is the proportional error constant, Ki is the integral gain constant, and Kd is the derivative gain constant. The Kp$e(t)$ term simply adds or subtracts a scaled error to the motor velocities. The integral term keeps track of all past errors, sums them and scales this term with Ki. The derivative term predicts the future error, by subtracting the last error from the current one, and then scales this with Kd. This allows $u(t)$, the adjustment term, to be determined not only by the current error, but by past and future errors as well. This means that the motor velocities are adjusted more “intelligently,” and therefore the robot is able to follow the line more closely by not overshooting or undershooting as much.

Another aspect of this robot design is the method to pick up the two boxes. One way to do this is using a robot arm. Robot arms can be made up of multiple bars and linkages, but in our case, one degree of freedom is suitable. On the robot arm would need to be a sensor to tell how close the arm is to the boxes (such as an IR sensor), and also a method to determine the color of the boxes (such as a light sensor). Because the boxes have magnets on each face, there are two methods for picking up the boxes: gripping like a human hand, or using a magnetic material. When using the magnetic material, it is a simple task to get the box attached to the arm, but to get it off, one needs a method to push the box off the arm, overcoming the magnetic force of the magnets on the box.

The last aspect in the design of this robot is the method to transport the robot across the lava stream. Since there is a ramp leading up to the lava pit, one method is to have the robot move fast enough to jump the pit. Another safer method is to drop a bridge down across the pit. This bridge would need to be able to be picked up by the same robot arm that picks up the boxes.

Methods and Materials

We are using wheeled locomotion as the method to move our robot around. Therefore, we are using two Pololu DC motors, one for each wheel. Each wheel is connected directly to each motor. To balance the robot, we will also have ball caster wheel, not connected to any motor so that it moves freely. It will be a ¾’’ diameter ball caster from Pololu, which is bigger than the one provided.

In terms of our line following control, we need four reflectance sensors. We need three reflectance sensors in the front of the robot that would be over the line at all times, as a reference to the position of the robot. We also need one reflectance sensor in line with the wheel axle, off center to the left or right wheel in order to make 90° turns.

The robot arm is the most complicated part of our robot. The position of the robot is moved using Power HD High-Torque Servo 1501MG that generates a torque of 17 kg cm, which after our calculation, is just enough for the lifting motion of our arm. We then have a hand design consisting of two plates, the front one with holes, the back one with steel plates that can stick through the holes on the front plate. The back plate is retracted using wire and a servo. The robot arm and hand will be made from laser cut ABS plastic. Also mounted on the robot arm is a color sensor, that can determine the color of the box being picked up.

The bridge design is included in our discussion section. It will be laser cut and made of ABS plastic. In the bridge will also be magnets so that the robot arm can pick up the robot.

The base of the robot will also be made of ABS laser cut plastic. On the base will also be mounted our controller, an Arduino Uno, our breadboard that contains all electrical connections to the sensors and motors, and our Li-Ion batteries. Our IR distance sensor will be mounted under the robot, to determine when the robot has reached each of the boxes.

<u>**Discussion**</u>

Our robot will implement line following using 3 reflectance sensors on a bar protruding out the front of the robot. The robot will be driven by two wheels (80mm - provided) connected to DC motors, which are driven by our Arduino through a separate power/driver circuit. There will be a third 'ball-caster' wheel that stabilizes and balances the robot located toward the front. In order for the robot to locate the areas on the ground where the lines cross each other perpendicularly, there will be a fourth reflectance sensor located by the wheels. When the sensor bar reaches the crossroads, the robot will drive forward until the fourth sensor is above the crossing lines. The robot will then rotate 90° until the three sensors on the bar line up with the new line. We designed the robot so that the two driven wheels are located in the rear so that we maximize the distance between the sensors and the wheel axis. Maximizing this distance improves the accuracy of the line following algorithm. The width of our robot was chosen to allow space for the DC motors to fit.

In order to carry objects, our robot will have an arm actuated by a servo motor. On the end of the arm there will be two plates that are forced together by springs. A second servo motor will be attached to the back plate via a wire that runs through the arm and around a pulley on the motor. When this motor rotates, the wire will pull the back plate backwards thus separating the two plates. The back plate will have steel rectangles that will stick through openings on the front plate so they can attach to the magnets on the boxes. Although steel is heavy, we feel this will be more effective than using electromagnets which consume large amounts of power. When the plates are forced to separate, the steel rectangles will pull back, increasing its distance from the magnet (until magnetic force is small enough that the object falls off). The plates will also be fitted with a distance sensor, light sensor, and a contact switch in order to facilitate the identification and retrieval of the boxes. The arm length was chosen to be as short as possible in order to reduce the moment arm on the motor. Since the provided motor would still not be capable of providing the necessary torque, we will be ordering a higher torque servo motor.

One problem that we determined was the servo motor we were given was not powerful enough to lift the box given the length of our arm. Considering our arm length is roughly 2", and a cube is 5" per side, the center of mass of the cube to the center of rotation on the moment arm is roughly 4.5" or about 11.5cm. Thus the torque required to lift the .5kg cube is about 6kg-cm, already much greater than the motors 3kg-cm torque rating. We decided to order the 1501MG motor giving us plenty of additional torque at 17kg-cm.

Although the IR sensor given to us is accurate to within 15 cm, our design requires the robot know when it is in fact touching the block. Thus the contact switch will be used to allow the robot to know it is touching something (in our case this will ideally be the blocks). The light sensor will be used to gather information about the color of the box (either light or dark). By first comparing the two blocks colors, the robot will be able to then determine which box it should move first. To save Arduino pins, we will ideally consider the first box the robot moves to be the light box (have not confirmed this counts with Professor). If not we plan to use an LED that will turn on briefly when the robot makes contact with the light box.

The robot will start the course holding onto a bridge part with its arm. The bridge will consist of a length of plastic with notches (to hold it in place) and a hinged section that allows it to fall into place perpendicularly. The hinged section will also have magnets in the same configuration as those on the boxes so it can be held in the same way. The first thing the robot will do is drive up the ramp and lower the bridge into place. It will accomplish this by driving to the edge of the top of the ramp, and lowering the bridge onto the other side of the ramp (10" bridge, 8" gap, the bridge will actually be about 1" past the proper position of 1" on either side of the ramp). Then, while still holding onto the bridge, it will slowly back up until the bridge "locks" into place (due to the notches made on the bottom of the bridge). We plan to tell the robot to simply back up until it is off the ramp, using the force of backing up to pull the bridge off the arm.

Some circuit diagrams that we will use in our project include the motor speed control, servo control and line following using reflectance sensors. Figure 2 describes the physical setup for controlling the speed of our motors that are attached to each of the the two wheels

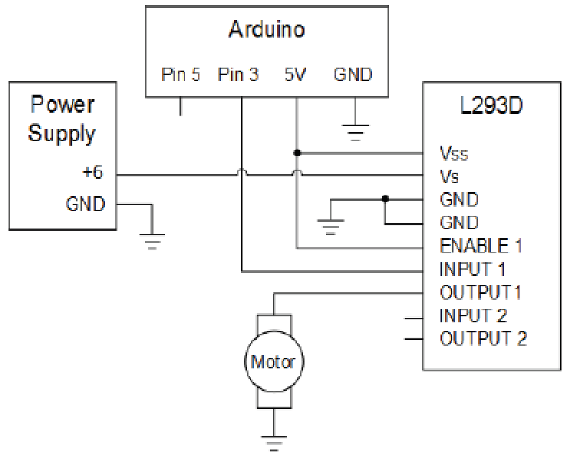

Figure 2: Motor Speed Control Schematic

Figure 3 describes the physical setup for controlling servo motor that will be used to control the arm motion.

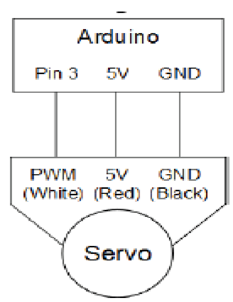

Figure 3: Servo Connections

These two diagrams only show how servo and DC motor normally interact with the Arduino. We might have different circuit diagrams for our project but the idea should stay the same.

Figure 4 describes the physical circuit setup for line following using three reflectance sensors placed in front of the robot.

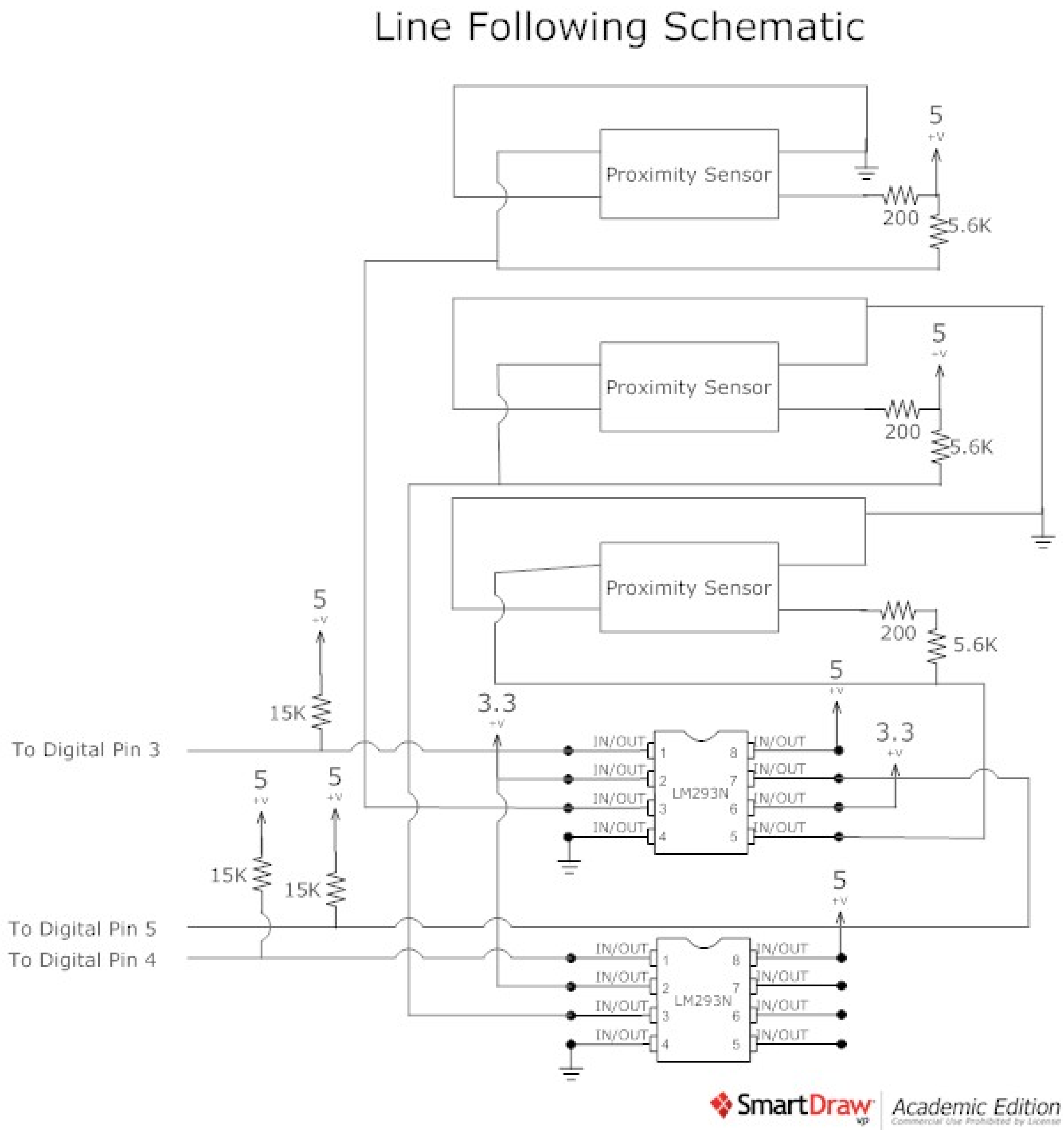

Figure 4: Line Following Schematic

Note: The proximity sensors in the diagram refer to the reflectance sensors.

A few pictures/sketches that show the dimensions and other details of our robot can be found in Designpictures.pdf.

Conclusion

The objective of our Mechatronics robotics project is to design a robot that autonomously navigates through the “Island Escape” course. The robot must be able to follow a line, pick up boxes, determine the color of the boxes, move itself and the boxes across the lava river, and place both boxes and itself in the end zone. Given free reign in our robot design, we decided upon a robot using two motorized wheels to allow the robot to move around, and three reflectance sensors to allow the robot to follow the line. To adjust the position of the robot with respect to the lines on the course, we are using PID control. In order to pick up the two boxes, we are using a robot arm, controlled by two servo motors that move the position of the robot arm and that move the back plate of the robot hand. The robot also will carry a bridge, which it can place over the lava river, and then drive over to get into the end zone.

Though our robot may seem like it’s designed for a very specific purpose, there are actually many real-world applications. Line-following robots are useful in the future of autonomous cars that could follow highway lines to navigate. In terms of picking up specific objects, our robot arm design could be used to pick up hazardous materials, stored in specifically designed boxes. Even though our robot is designed to navigate a specific course, the elements in its design are useful in many other applications.