Autonomous Laser Locking System

NSF Summer Undergraduate Fellowship in Sensor Technologies Brett Kuprel (Electrical Engineering) – University of Michigan

Advisor: Professor Daniel D. Lee

ABSTRACT

The purpose of this project is to create an autonomous laser locking system for a robot that will be used in the Multi Autonomous Ground-robotic International Challenge (MAGIC). Locking onto a target has many applications in robotics. The laser could be replaced by a flashlight, a camera, a projectile launcher, etc. Solving this problem requires the use of coordinate transformation matrices to deal with multiple reference frames. It also requires sensor analysis to determine positions of both the robot and the target. In this paper I describe an approach to solving this problem.

1. INTRODUCTION

In many robotics applications, it is important to have the ability to lock onto a target. For example, a robot that has a camera on it might be more useful if it could keep the camera pointed at an object of interest while the object is moving. The target locking system could be used for a number of things such as a laser pointer, a flash light, or a missile.

In general, for something to remain pointed at an object in three dimensions, it must have two degrees of rotational freedom. This allows it to remain locked onto the target regardless of where the robot is - or how it is oriented.

This paper presents my results of integrating a laser pointer locking system into an autonomous ground vehicle. The direction of the laser is determined by the angle of each gear. In order to keep the laser locked onto a target, the angle of each gear must be continuously calculated as a function of the robot's orientation and position with respect to the location and motion of the target.

2. BACKGROUND

2.1 Simultaneous Localization and Mapping (SLAM)

SLAM is an algorithm used in robotics to map an unknown environment while at the same time maintaining an estimate of current position. When these processes occur concurrently, the error in estimation converges [1]. This algorithm requires input from sensors, some of which are described in this paper. A 3D map resulting from a SLAM algorithm is shown in Figure 1.



Real-time SLAM visualization by Newman et al [2]

2.1.1 Accelerometer

Accelerometers measure acceleration. Most methods involve a linearly elastic material (F=kx). In order to convert the mechanical motion into an electric signal, components such as piezoresistors or capacitors are used. A piezoresistor's resistance depends on the stress applied to it. The voltage across a capacitor is inversely proportional to the separation of the conducting plates. In robotics applications, three accelerometers – ideally positioned perpendicularly to one another – are required to obtain acceleration in all three dimensions.

2.1.2 Gyroscope

Gyroscopes measure angular velocity. One method of doing this is by taking advantage of angular momentum conservation. A heavy disk spinning at a high frequency resists change in angular momentum. When this disk is mounted on low friction bearings that have freedom to rotate along three linearly independent axes, the external torque of a moving object applied to the

spinning disk is minimized. The result is that the axis of rotation of the disk remains fixed. Another method is a vibrating structure gyroscope. The physical principle is that a vibrating object tends to keep vibrating in the same plane as its support is rotated. A third more precise and also more expensive method takes advantage of light interference in a coil of fiber optic cable.

2.1.3 Light Detection and Ranging (LIDAR)

LIDAR works along the same principles as RADAR with the difference being the wavelength of radiation emitted. Light is sent out and scattered when it reaches an object. Some of this light is scattered exactly back in the direction of the LIDAR sensor. The sensor calculates the distance to the object based on the time it took for the light to return. It does this for many angles in a plane. To give an idea, the LIDAR we used returns an array of distances for 1,080 angles over a range of 270 degrees (90 degrees is cut off from the enclosure) at a rate of 40 Hz. LIDAR has problems with transparent objects such as glass because the light does not reflect.

2.1.4 Global Positioning System (GPS) Satellite Receiver

A GPS receiver works on the same principles as LIDAR: the time light takes to reach an object determines the distance. One difference is that LIDAR is used as both the source and the receiver of the light. The way GPS works is that a satellite acts as the source and sends timestamp information with the light. One source, however, is not enough to determine a unique position. If all that is known is the distance to the satellite and the position of the satellite, the possible positions of the receiver would include all points on the surface of a sphere centered around the satellite with a radius equal to the distance measured. If one assumes that the receiver is on the surface of the earth, the possible positions would be the intersection of the sphere and the earth, which is roughly a circle (Figure 2, top). The intersection of three spheres would be two points (Figure 2, bottom). If these points are far enough apart, and the position of the receiver is roughly known, this could be enough information. Assuming perfect accuracy and precision, four intersecting spheres would determine a unique position in space. This could be 3 satellites and an ellipsoidal approximation of the earth, or just





4 satellites. As the number of satellites increases, the error due to inaccuracy and imprecision decreases. In general, a GPS receiver is able to receive a signal from at least four satellites and up to as many as twelve.

2.1.5 Odometry

An odometer measures distance traveled by a wheel. This can be accomplished through optical or electromagnetic methods. The electromagnetic method involves two sensors attached to fixed

points on the wheel with one magnet on the spinning electric motor. When the magnet on the motor passes one of the sensors, it causes a spike in voltage. The frequency of these spikes determines the angular speed of rotation. The second magnet is used to determine direction of rotation, which is determined by the order of spikes (i.e. magnet 1 then magnet 2, or magnet 2 then magnet 1). The optical method involves a light source, a photo sensor, and a perforated disk mounted to the motor. The frequency of spikes in the photo sensor is directly proportional to the angular speed of the wheel. Both techniques return an angular velocity of the wheel, but this alone is not enough to determine distance traveled. The diameter of the wheel must also be known. The diameter and the time integrated angular velocity of the wheel are enough to determine the distance travelled by a single wheel.

2.2 Differential Drive

Knowing the distance traveled by one wheel of a vehicle is not sufficient for determining the change in position of the vehicle. If one wheel is travelling faster than the other, the resulting action of the vehicle will be to turn. If the vehicle has two wheels, the motion of the vehicle can be modeled as the center arc of the two arcs travelled by the wheels (Figure 3). If the vehicle has four wheels, it is necessary that they will slip, and so the two-wheel approximation is not as accurate.



Modelling the motion of a two-wheeled robot based on the distance travelled by each wheel

2.3 Motion Planning

In robotics it is important to get from point A to point B in the most efficient way. Efficiency is determined by the cost of travelling the path. The cost can be a function of time, distance, terrain, etc. One algorithm of approaching this problem is called the A* Search Algorithm [3]. It works by finding the path that minimizes the cost. In order to do this, it must have a cost map to work with. This is obtained by SLAM, and any other cost modification relevant to the application.

2.4 Homogenous Transformations

Often it is convenient to describe various locations in different coordinate systems. An example is converting between a global coordinate system and individual local coordinate systems. To convert points between coordinate systems, transformation matrices can be used [4].

2.5 Proportional-Integral-Derivative (PID) Controller

A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting control parameters. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been

changing. The weighted sum of these three actions is used to adjust the process via a control element



Figure 4

A block diagram of a PID controller

3. APPROACH

In order for the robot to point a laser at a target, the robot must know its own position and orientation. This task is more difficult than it might seem. A naïve approach would be to use only GPS. The problem with this is that GPS is only precise within about a half meter. Also, when a GPS receiver is near a building, it will report a false position due to multipathing which means that the satellite signal was reflected before it reached the receiver. A better method is to simultaneously map the robot's environment and localize it within that map also known as Simultaneous Localization and Mapping (SLAM). SLAM synthesizes various sensors including lidar, gyroscopes, and wheel encoders. The primary sensor is the lidar. An example of a typical lidar output is shown below:







SLAM works by adding each new lidar output to a collaborative map. If a robot were to drive inside a building, SLAM would build a map of the interior. In order to do this, SLAM must know how to add consecutive lidar maps. For example, if the robot turns, then the new map will different from the old one by a rotation. If the robot goes straight, the new map will be shifted backward compared to the old one. SLAM determines what the robot did between maps by using encoder and gyroscope data. Encoders give information about translation, while the gyroscopes give information about rotation. This information, however, is not absolute. SLAM makes small variations in rotation and translation to find a least squares fit.

For a laser to be able to lock onto a target that can move in three spatial dimensions relative to the laser, the laser must have two degrees of rotational freedom. This was accomplished by using 2 servos. A servo is basically a gear that can rotate to a specific angle. The angle is determined by the voltage applied to it. The servos we used can rotate between 0 and 300 degrees. When oriented so that one servo controls yaw and the other controls pitch, the resulting field of view is all points in space except for a 60 degree pyramid. It is shown in Figure 4.



Figure 6

Left: Field of view of laser Right: roll, pitch, and yaw w.r.t. the robot

The direction of the laser is determined by the angles of the servos. These angles need to be continuously calculated as a function of the robot's position and orientation, as well as the target's position and orientation. The first step I took was to find the target in robot coordinates, which is initially given in global coordinates.



A straightforward way of converting points between global and robot coordinates is by using a transformation matrix. The way a transformation matrix works is that one first determines the series of rotations and translations of the original coordinate system required to match it up with the other coordinate system. Each rotation/translation corresponds to its own transformation matrix. The final matrix is calculated by multiplying the individual matrices in order. To match the global coordinate system up with the robot's coordinate system, the following transformations have to happen in order (use Figure 5 as a reference):

- 1. Translate along x_G axis to x coordinate of robot, along y_G axis to y coordinate of robot, and along z_G axis to z coordinate of robot
- 2. Rotate around the z_G axis the yaw of the robot
- 3. Rotate around the y_G axis the pitch of the robot
- 4. Rotate around the x_G axis the roll of the robot
- 5. Translate along x_G axis to x coordinate of the laser, along y_G axis to y coordinate of the laser, and along z_G axis to z coordinate of the laser

Multiplying these transformations together results in a matrix that will convert robot coordinates to global coordinates. The goal, though, is to do the opposite: convert global coordinates to robot coordinates. This matrix can be found by simply finding the inverse of the other matrix. The transformation matrix is shown in Equation 3.

$$T_{G \to R} = [Trans(RobotPosition) * Rot_Z(yaw) * Rot_Y(pitch) * Rot_X(roll) * Trans(LaserOffest)]^{-1}$$
[1]

$$T_{G \to R} = \begin{pmatrix} \begin{bmatrix} 1 & 0 & 0 & x_R \\ 0 & 1 & 0 & y_R \\ 0 & 0 & 1 & z_R \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(yaw) & -\sin(yaw) & 0 & 0 \\ \sin(yaw) & \cos(yaw) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(pitch) & 0 & \sin(pitch) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(pitch) & 0 & \cos(pitch) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(roll) & -\sin(roll) & 0 \\ 0 & \sin(roll) & \cos(roll) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & x_L \\ 0 & 1 & 0 & y_L \\ 0 & 0 & 1 & z_L \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1}$$
 [2]

Although calculating this matrix is complicated, the application is quite simple. To convert a point in global coordinates to robot coordinates, one simply needs to multiply the global coordinates by the transformation matrix.

$$\begin{bmatrix} x_{R} \\ y_{R} \\ z_{R} \\ 1 \end{bmatrix} = T_{G \to R} * \begin{bmatrix} x_{G} \\ y_{G} \\ z_{G} \\ 1 \end{bmatrix}$$
[3]
$$R = \text{robot} \\ G = \text{global}$$

To switch from robot coordinates to global coordinates, multiply by the inverse:

$$\begin{bmatrix} x_G \\ y_G \\ z_G \\ 1 \end{bmatrix} = (T_{G \to R})^{-1} * \begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix}$$
 [4]

Notice that when the robot moves the transformation matrix changes. The matrix must be recalculated every time new information about the robot's position and orientation are received.

Once the target is known in robot coordinates, the angle of the horizontal/yaw servo can be calculated. The requirement is that the plane of the vertical/pitch servo be lined up with the target.



Figure 8

The sensors do no rotate around the same point, and so calculating the angle of each requires the offset from one servo to the other. The horizontal angle can be found by first calculating the angle as if there were no offset, and then making a correction to compensate for the offset.



Figure 9

Calculating the required angle of the horizontal/yaw servo to align the pitch servo's plane with the target. It requires a small correction $d\theta_H$

$$\theta_H = \arctan\left(\frac{y_T}{x_T}\right)$$
[4]

$$d\theta_H = \arcsin\left(\frac{o_y}{\sqrt{x_T^2 + y_T^2}}\right)$$
 [5]

$$\theta_H = \theta_H - d\theta_H \tag{6}$$

Notice that if the y offset is zero, then no correction needs to be made. Also there is no dependence on the offset which is in the direction of the laser (x offset).

Now, the vertical angle needs to be calculated. The first step I took in doing this was to create another coordinate system centered on the pitch servo. This means another transformation matrix is required to find the coordinates of the target in this new coordinate system.



Figure 10

The new coordinate system is different from the old one by a translation equal to the distance between the center of the horizontal/yaw servo and the center of the vertical/pitch servo. Also there is a rotation equal to the angle of the horizontal servo $\theta_{\rm H}$

$$T = [Trans(ServoOffset) * Rot_{Z}(\theta_{H})]^{-1}$$
[7]

Finding the angle of the vertical/pitch servo now is easy.



Although these equations allow the laser to accurately point at a target, they do not take into account the relative motion between the target and the robot. There are two problems that occur with setting the angle of each servo.

- 1. The angles of the servos are updated at discrete time intervals which causes the laser point to be unsteady
- 2. The servos are designed such that when a target angle is set, the servo comes to a complete stop at the target angle. When the target angle is constantly changing, the servo never reaches it.

The solution to the first problem is to make the servo's motion more continuous. There are two ways of approaching this problem:

- 1. Increase the frequency at which the target command angle is sent
- 2. Send angular velocity commands instead

The first option will not work in our case due to the fact that the position and orientation information updated at 40 Hz, and this is limited by the angular velocity of the lidar. Most often, the target angle changes because the position and orientation of the robot change. Increasing the target angle command rate to 400 Hz would mean that the same target angle command would be sent 10 times. The second option is reasonable and it also solves the second problem. Sending angular velocity commands requires calculating the desired angular velocity or angular acceleration. I wrote a PID controller to do this. The controller looks at the difference between the desired angle and the target angle and sets the angular velocity accordingly.

4. RESULTS

The autonomous laser locking system was tested with a robotics simulator.



Figure 11 (left)

The autonomous laser locking system was tested with a physics simulator. The model of the robot was imported from SolidWorks. In the simulator, the yellow ball moves while the robot drives around. The laser remains pointed at the target.

Figure 12 (right)

The laser is targeting the center of the red bin.



5. CONCLUSIONS

In order to create a laser locking system, the following steps were taken:

- Estimate position
- Receive target in global coordinates
- Calculate target in robot coordinates
- Calculate angles of servos
- Do PID control on angle error to obtain an angular velocity
- Repeat

6. RECOMMENDATIONS

The servos are currently controlled with angular velocity commands. It would make more sense to control them with angular acceleration commands because acceleration can change almost instantaneously whereas velocity cannot.

7. ACKNOWLEDGEMENTS

I would like to thank:

- Professor Dan Lee (research advisor)
- Alex Kushleyev (graduate student in lab)
- Dr. Jan Van der Spiegel (Sunfest coordinator)
- National Science Foundation

8. REFERENCES

- [1] Durrant-Whyte, H. and Bailey, T., "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms," *Robotics and Automation Magazine*, 2006
- [2] P.M. Newman, J.J. Leonard, J. Neira, and J. Tardos. "Explore and return: Experimental validation of real time concurrent mapping and localization". *Proc. IEEE Int. Conf. Robotics and Automation*, 2002
- [3] PE Hart, NJ Nilsson, B Raphael, "A formal basis for the heuristic determination of minimum cost paths". *IEEE transactions on Systems Science*, 1968
- [4] Jennifer Kay. "Introduction to Homogeneous Transformations & Robot Kinematics." *Rowan University Computer Science Department*. January 2005