Lab 4: Obstacle Avoidance

So far we have navigated the test track with GPS and with a path planning algorithm. Finally, in this lab we used ERTS' front-mounted laser range finder to avoid obstacles placed around the test track.

For the simulations, a pre-built array of obstacles containing 9 more arrays containing the name of the cone, the GPS coordinates for it, and the size of the obstacle. The obstacle_list array is fed to the synlaser_s class so that it may know where the obstacles are. A variable called obs_found is intiaialised as false and the integer obs_count is initialized to 0, but will switch to "turn_m" when an obstacle is found and will increase the count according to how many are seen.

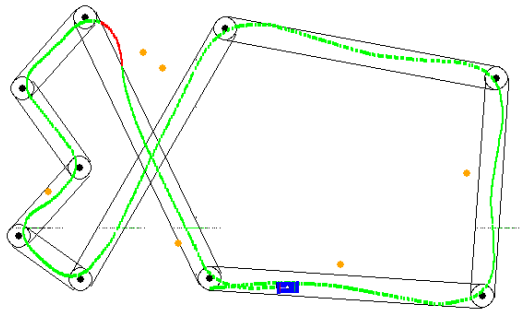

Figure 1: Orange dots are obstacles

When an obstacle is spotted, first the angle and distance to the obstacle must be computed. This is achieved by using elements from another class called geopy. Geopy's gps_s is used to calculate the distance to the obstacle by taking the GPS coordinates of both the obstacle and ERTS and comparing them using the distance formula. The difference of angle is calculated in two parts: angle1 and angle2. Angle2 uses geopy's distance function to find the forward azimuth of the obstacle. Angle1 is found in a similar way, but used the forward azimuth of the next way. Angle2 is subtracted from angle1 to find the angle to the obstacle. Using all of these components, the distance is calculated. If the distance is calculated to less than or equal to .00009 and the absolute value of the difference of angles is less than or equal to 15, an obstacle is counted as detected obs_found switches from false to turn_m.

Next the best way to avoid the obstacle must be calculated. The same method of path planning using synthetic points (refer to lab 3) is used except with the added use of the compass. The current heading is calculated using the forward azimuth of ERTS and the heading of the next waypoint is calculated the same. Depending on the compass headings, the fourth synthetic point moves causing the cart to veer to the left or right. If the obstacle is no longer seen, it continues on as normal. If the obstacle is still in sight, the synthetic points will continue to shift.

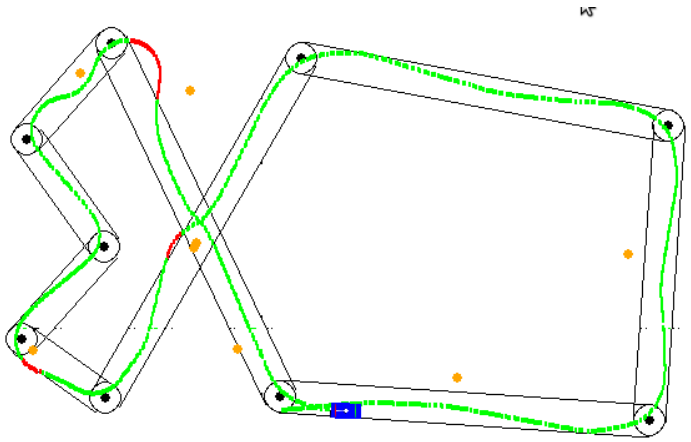

Figure 2: Alternate obstacles

Just to be sure that everything worked as it should, the simulator was run again, but this time with a slightly altered list of obstacles. Everything performed as hoped.