Lab Report IV

S.R. Sinha and B. Zaitlen

November 10, 2011

1 Report

1.1 Purpose

Our objective in this lab was to implement an object avoidance algorithm. Our implementation was to enable autonomous navigation between the nine different predefined waypoints (PWPs) while staying within the course corridors and avoid objects placed within the course. The goal was to first verify our implementation using the Cart Visualizer simulator and then field test our program.

1.2 Location

ERTS Test Field, Indiana University parking lot north of indoor track

1.3 Path Planning

1.3.1 Procedure

We introduced enhancements to our two-state navigation system introduced in Lab 2. That system implemented turning and straight states that permitted navigation between predefined waypoints (PWPs). Our current path planning routine adds additional waypoints (AWPs) and positions these AWPs to allow us to better position the cart coming into and out of turns. As a result the cart stays within the corridors over a large fraction of the course.

Our first step was to introduce AWPs to reduce oversteering in turns. The number of additional waypoints was variable and we generally chose three to nine (Figure 1, right, green dots). The AWPs were positioned at different fractions of the distance, d, between each pair of PWPs. We routinely placed one AWP was at (0.1 x d), a second at (0.9 x d), and the remaining equally spaced between.

The AWP were strategically positioned on a line that was rotated with respect to the the center line of a corridor (Figure 1, right, black dashed line). The direction of rotation depended on three PWPs:

1. The waypoint just departed,

2. The waypoint the cart was navigating towards, the target, and

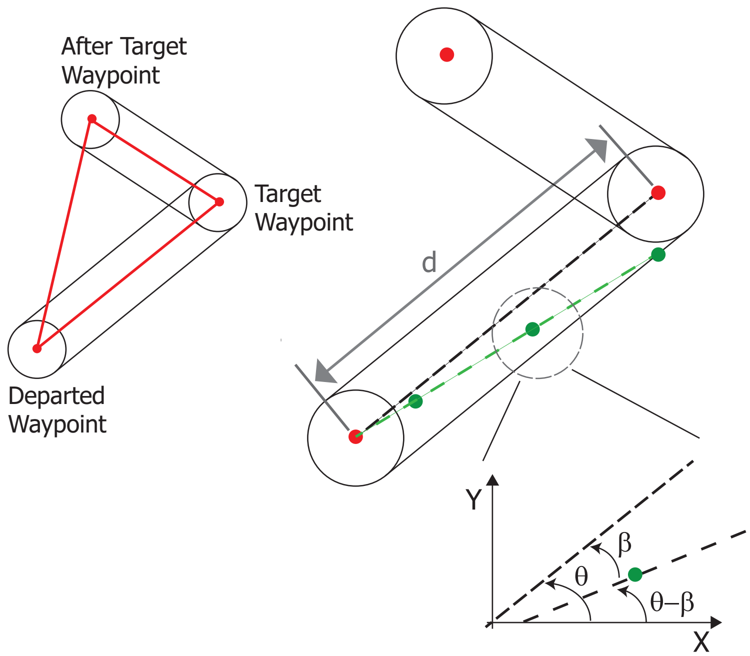

Figure 1: Schematic illustrating additional waypoints for path planning. Red dots represent the original predefined waypoints. Green dots represent added waypoints. shown too are the center line down the corridor center (black dashed line) and the line connecting the additional waypoints (green dashed line).

3. The waypoint after the target waypoint.

These three waypoints form the vertices of a triangle (Figure 1, left). We calculated the area of the triangle formed by these waypoints using Equation 1.

$$triangle\_area = .5 * (x1 * (y2 - y3) + x3 * (y1 - y2) + x2 * (y3 - y1)), \quad (1)$$

where (x1, y1), (x2, y2) and (x3, y3) respectively represent the coordinates for the departed waypoint, the target waypoint, and the waypoint after the target.

The area of this triangle could be positive or negative depending on the relative positions of the three PWPs. If the calculated area of the triangle was positive (negative) we rotated the line for AWPs clockwise (counterclockwise) relative to line joining the departed and target waypoints.

The next step was to determine how much to rotate the line along which to position AWP. We introduced a Cartesian coordinate system centered on the departed waypoint (Figure 1, right, inset). The straight line formed between the departed and the target waypoints forms an angle, θ, with respect to the X-axis. The angle, β, we determined using the departed waypoint, the AWP at (0.9 x d), and the fact that this AWP was positioned on the corridor boundary, i.e. an LBO distance away from the corridor center line. Therefore,

$\beta = atan(LBO/(0.9 \times d))$. The resulting angle, $(\theta - \beta)$, is the angle used to determine the coordinates of an AWP.

1.3.2 Simulated Results

Our initial simulation results (Figure 2) appeared promising. Simulations showed that we were able to stay largely within the course corridors. With fixed speed and braking we were able to stay within 2.5 meter corridors, even in regions where consecutive PWPs (7-9, left of figure) were closely clustered. The simulation shown in Figure 2 shows eight AWP between each pair of PWP. Introduction of these AWP by itself shows that the cart is better able than in Lab 2 to stay within the course corridors. Figure 3 shows a similar simulation result, but suppresses the AWP and shows the LBO used in field test, namely an LBO of 2.5 meters. The cart leaves the corridor only at the starting position, therefore as a result of the initial starting position.

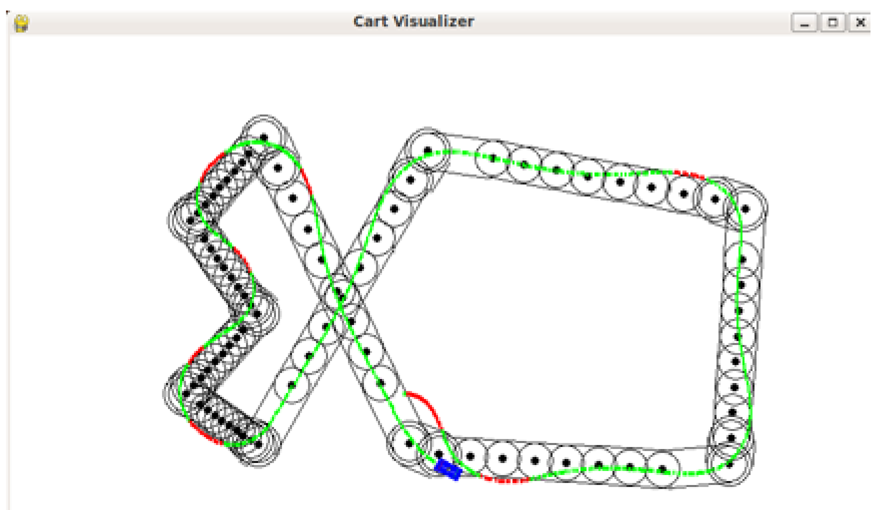

Figure 2: Simulation results from the implemented path planning routine. In this example simulation we added eight additional waypoints between each pair of waypoints. An additional waypoint was added at 10% - 80% in 10% intervals of the distance between each pair of waypoints. Each additional waypoint has the same LBO as the waypoints provided.

After closer examination of the simulations we recognized deviations in path planning that were inconsistent with our intended navigation goals. For instance, in Figure 2 the AWPs all fall on a line that overlaps with the corridor centers. In addition, we expected the first AWP after a PWP to be close to the center of the corridor and the last AWP to be closer to the corridor boundary. This is clearly not the case, as is obvious in Figure 3. The discrepancies were a result of incorrectly calculating our rotation angle θ. See below for more details.

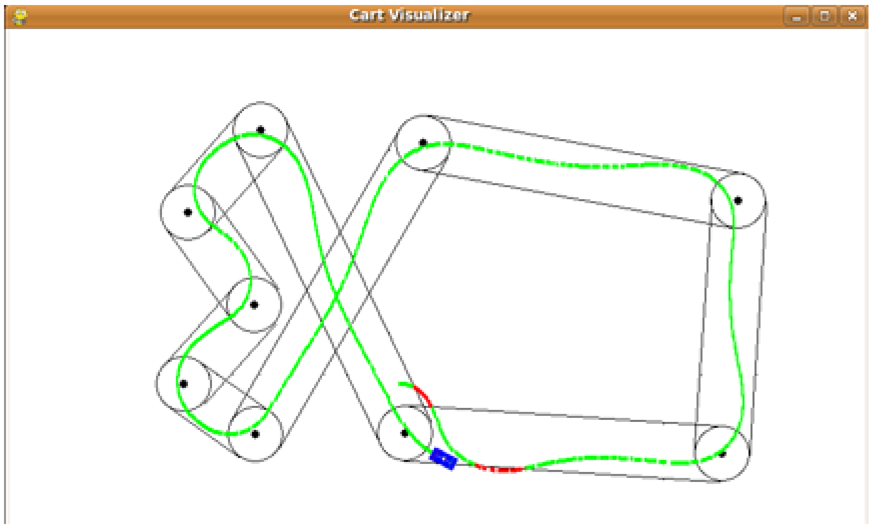

Figure 3: Simulation results using the Cart Visualizer. As in Figure 2, eight additional waypoints were included spaced equally between 10% and 80% of the distance between original waypoints. The added waypoints are not shown here, but have an LBO of 1.5 meters, versus the original waypoints which have an LBO of 2.5 meters. The cart stays fully within the corridors, the only exception being when it first starts navigating the course.

After carefully examining our code, we identified the sources of our error. The improved simulation results are exemplified in Figure 5.

1.3.3 Field Results

Our field test of the path planning program had mixed results (Figure 4). [This was prior to our identifying our algebraic errors in positioning AWPs.]. The telemetry data we collected made clear unanticipated cart navigation problems. The path followed by the cart was similar to the simulation results shown in Figure 3.

Ultimately we tracked down the source of our error to be in the calculation of the rotation angle, θ. The choice of coordinate frames, and position of the PWPs (exemplified in Figure 1) defines the angle, θ. The second PWP is effectively in the positive portion of the X-Y plane. However, if the second PWP is in a different quadrant a corrected version of θ needs to be used. For example, if the second PWP is in quadrant II, the angle $(\pi/2 - \theta)$ should be utilized in determining the position of AWPs.

After correction for this error, we ran simulations to examine the path planning (Figure 5). The results were completely in line with our expectations. However, due to time constraints, we chose to move onto developing our obstacle avoidance program, and delayed field testing of our improved path planning

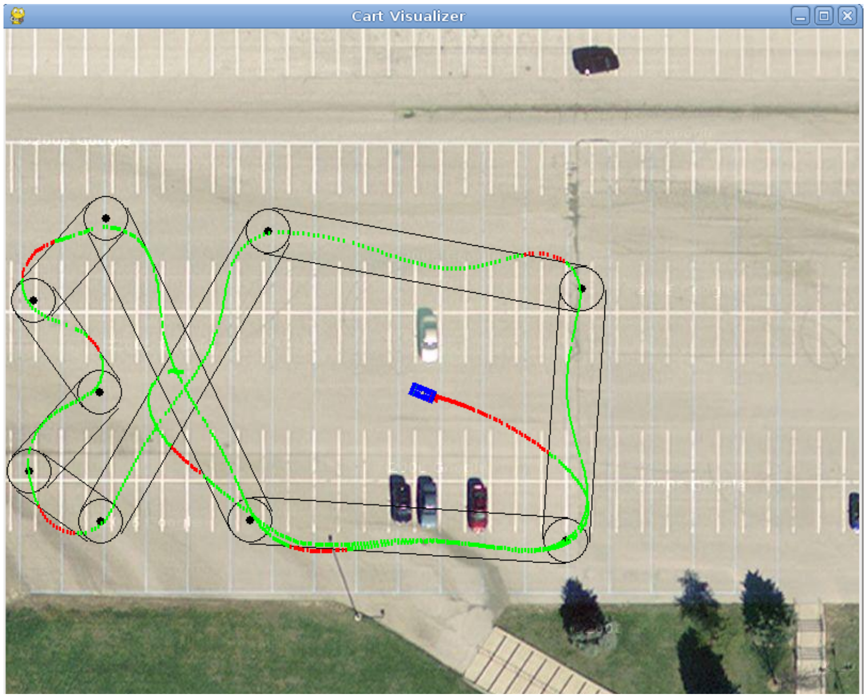

Figure 4: Mixed results during field testing of our initial path planning program. the cart is moving in a counter clockwise direction. The movements toward the AWP positioned at $(0.9 \times d)$ were clearly deviating from our design. AWP at these positions are just before a predefined waypoint. d represents the distance between each pair of waypoints.

program.

1.3.4 Field Observations

Overall we observed what we initially thought was good performance on the course. We largely stayed within the corridors when the LBOs were 2.5 meters. However, we had far greater difficulty with an LBO of 1.5 meters. After initially looking at the trajectory of the cart (Figure 4) and later analyzing telemetry, we observed problems in navigation. All in all, even with errors in our calculations, the addition of waypoints kept the cart within the course corridors (though not to any sophistication in our algorithm).

For the future, we foresee three possible enhancements to improve our performance in cases where the LBO is smaller. First, instead of placing the additional waypoints along a line, better results may be achieved if an arced path is chosen. Second, as opposed to placing additional waypoints equally space between waypoints, better performance may be achieved with more waypoints close to turns, or around obstacles, and fewer along straightaways. Last, while our per-

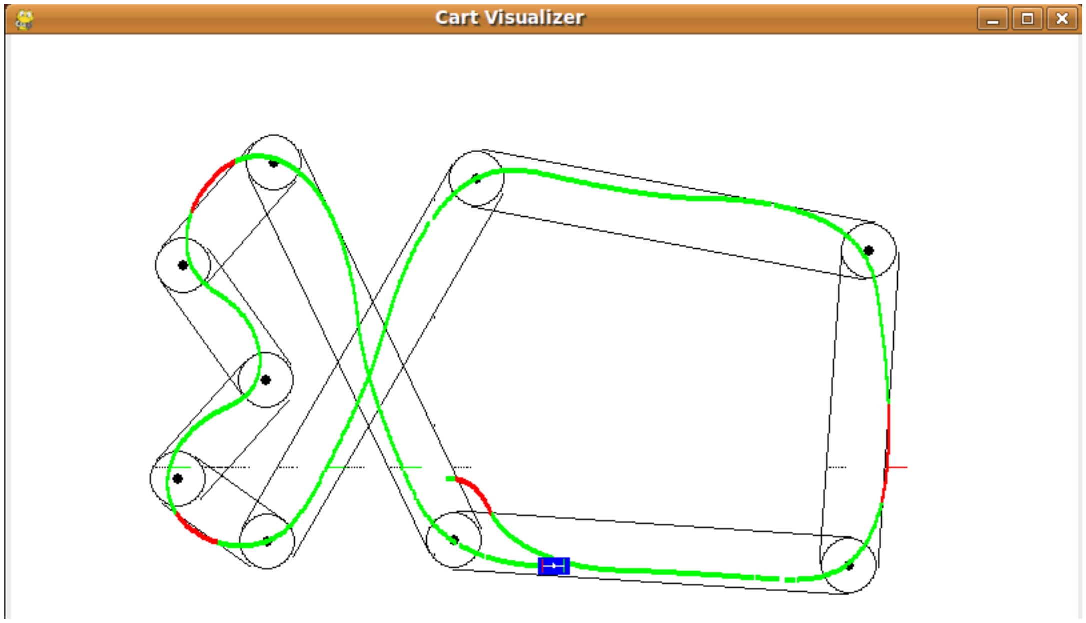

Figure 5: Telemetry data showing latitude and longitude of the cart. The cart navigated the course and passed within the LBO of each of the nine predefined waypoints and the additional waypoints introduced for path planning. Each waypoint, and its corresponding LBO, is colored to make them easier to identify.

formance was adequate a better speed control system would further improve our performance by reducing the time we take to navigate turns and the overall course. More importantly, speed control is tied to our ability to stay within the corridors, so warrants further attention.

1.4 Obstacle Avoidance

1.4.1 Procedure

We continued to use our additional waypoint method and implemented a two-state object avoidance algorithm based on geometry. The two states are:

1. If one obstacle was detected we added an avoidance waypoint to navigate around the object.
2. If two obstacles were detected:
   (a) If both obstacles were in parallel with the track, we added an avoidance waypoint relative to the second object to navigate around the object.

(b) If both obstacles were orthogonal to the track, we ignored the obstacles and continued.

When we add an avoidance waypoint (Cases 1 and 2b) orthogonal to the obstacle and the center line and within the corridor. We can calculate such a point at a predefined distance, h, away from the obstacle based on the distance between the obstacle and the cart, and the angle, θ, of the corridor.

$$\begin{aligned} xAdjusted &= math.cos(beta) * R * 1.0/86358.0 + xObs \\ yAdjusted &= math.sin(beta) * R * 1.0/111122.0 + yObs \end{aligned} \tag{2}$$

where $xObs$ and $yObs$ are the (x,y) coordinates of the obstacle (see Figure 6).

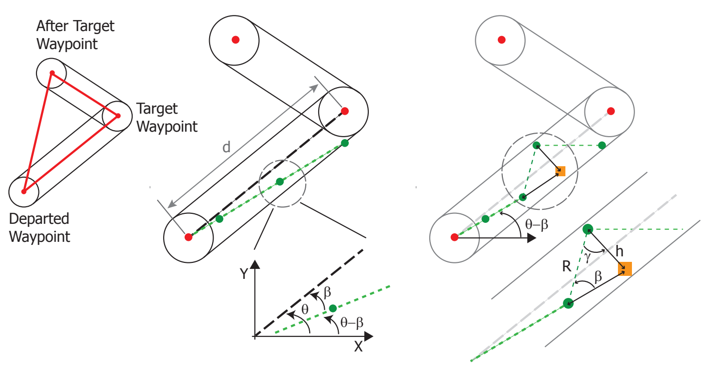

Figure 6: Schematic showing our algebraic calculations of obstacle avoidance.

The laser range-finder returns an exact GPS location of the obstacle and we can easily determine the distance between the cart and obstacle.

1.4.2 Simulated Results

Our initial simulation results (Figure 7) appeared promising. Simulations showed that we were able to stay largely within the course corridors while avoiding all obstacles.

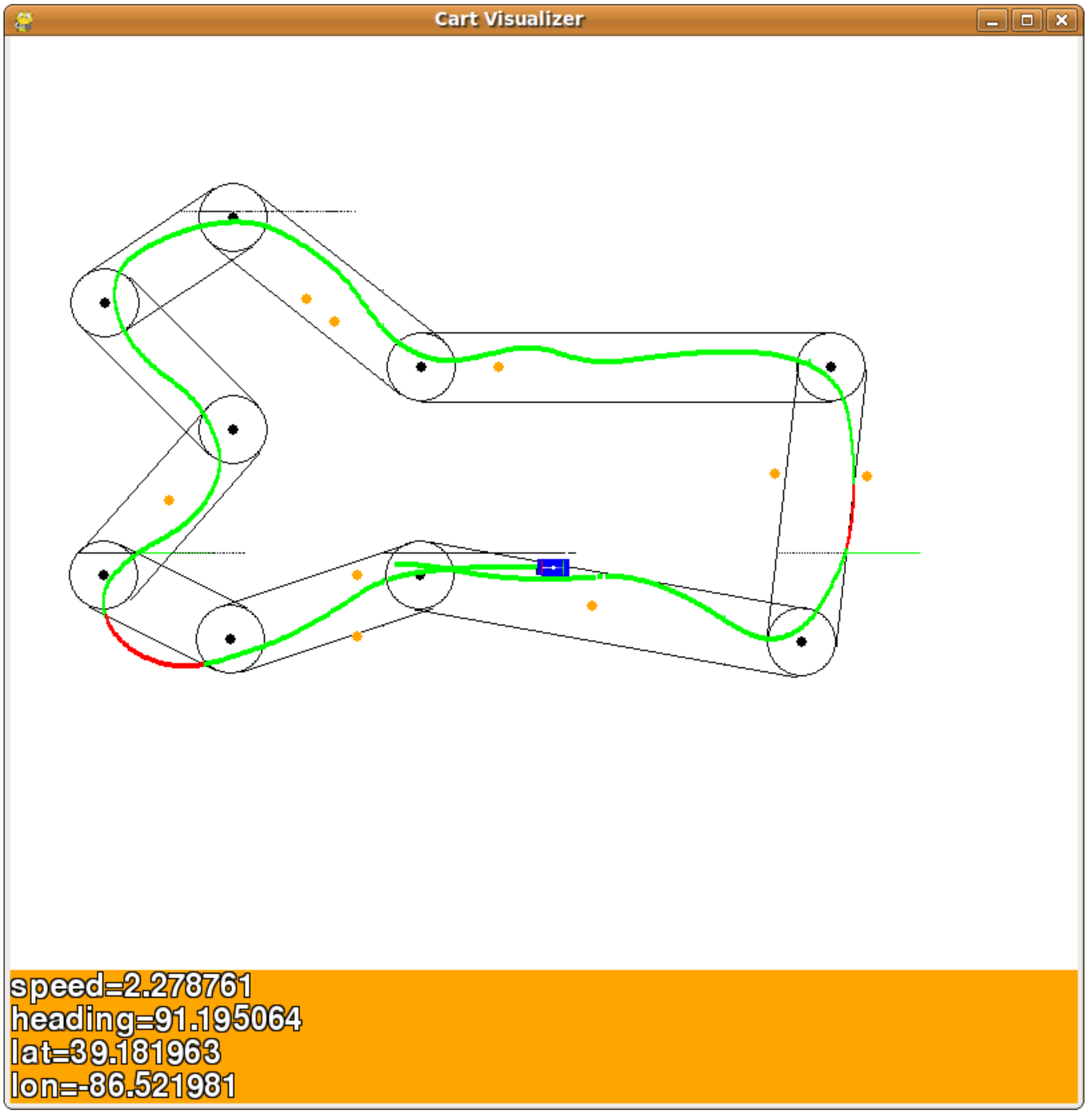

Figure 7: Simulation results of Object Avoidance algorithm. In this example simulation 9 cones are dispersed throughout the course.

With a working simulated avoidance algorithm we were very eager to experimentally verify and refine if necessary. *NOTE: we should have plotted the waypoints added to avoid obstacles to help us confirm our solution.*

1.4.3 Field Results

Our field test confirmed our simulation results. The cart largely followed the same trajectories as the simulated cart. In fact, the experimental results look a bit better than simulated results; the cart only moves out of the corridors at the end of the course (waypoints 7 and 8). See Figure 8.

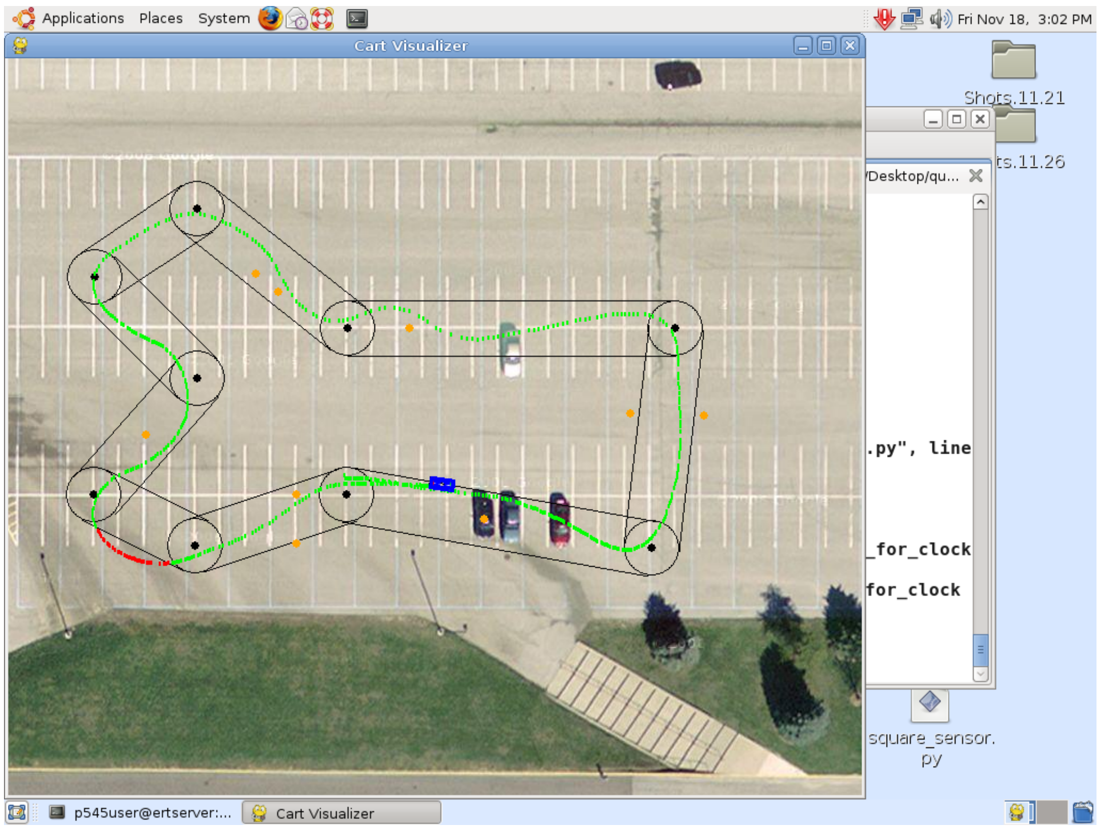

Figure 8: Experimental results of Obstacle Avoidance algorithm

1.4.4 Field Observations

Overall, we successfully navigated the course and avoided obstacles. However, we when returned to analyze our results we noticed several inconsistencies, which did not invalidate the results, but most likely would lead to better performance.

We first want to plot the location of added waypoints used to avoid obstacles. See Figure 9.

Green triangles represent obstacles, red dots represent PWPs used for path planning, and blue hexagons are the locations of waypoints added by our algorithm to avoid obstacles. Notice that the blue hexagons are not orthogonal

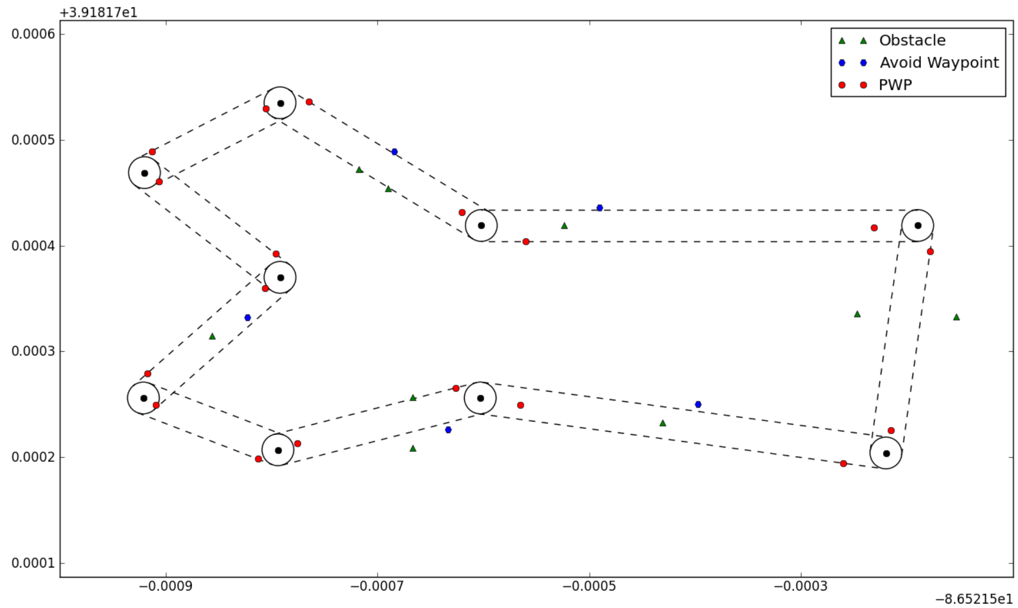

Figure 9: Plot of avoidance scheme based on experimental data

to the obstacle and the corridor. After inspecting the code we found that we were mistakenly using the angle γ instead of β (see Figure 6). While the cart successfully avoid obstacles we were quite lucky that our incorrect geometry worked. Any future work should correct this calculation immediately.

We did not calculate the obstacle distance to the corridor. Luckily, most obstacles were relatively centered along the corridors path and therefore each choice was equally valid. For future work, we should make such a calculation and determine whether to move right or left around an object. Additionally, if one detects two obstacles orthogonal to the corridor, we should calculate the midpoint between the two obstacle and add a waypoint.