Lab Write-Up Midterm: Waypoints and Pathes  
Investigator: Cody Glover

Problem: 
This report covers the results of two labs presented the first is the design and implementation of a 
waypoint driver. This driver would direct the cart in on how to reach a series of gps points in accending order. 
Each point is give an radius creating a circle around the point allowing for some error and later path planning for 
the cart. Other requirements can be added to the waypoint such as speed and angle but for this labs purpose 
those are removed. The other lab covered builds upon the first by restricting the movement and path of the 
cart. For this lab corridors or "roads" are created by creating tangent lines from the edge of the circle created by 
one waypoint and the circle created by the next waypoint. The goal is then to have the cart reach each 
waypoint while staying with in these defined corridors. These two labs expand upon the square driver used in 
the first lab as well as the resources given for it. Most importantly this includes the cart simulator for virtual 
testing and trial runs on the actual ERTS for real world data collection.

Design Method: 
The design method implemented is a logic based decision making process. Which takes the current 
state of the cart and the requested action calculates the best course of action in meeting the request while 
staying within the corridor restrictions. While the labs allow for preplanning of the course as a whole such a 
process would restrict future implementations. So the design method restricts itself to only the current state of 
the cart and the next two waypoints or rather the next corner created by these three points. The decision 
process uses a series of data table which change cart settings in response to the requested actions and 
parameters given.

Waypoint Driver: 
The original waypoint driver was created using lab one's square driver as a base as well as the optimal 
p-value determined during it. The turning function and cart start up functions were kept, while the control 
function was rewritten to go to a waypoint rather than a strict course heading. The control function now takes 
the current heading of the cart and uses python's geopy function library to calculate the direction of the 
targeted point by passing in the latitude and longitude of the cart and waypoint. It also calculates the distance 
in which the cart is from the targeted point. When the calculated distance is less than the radius of the point, 
i.e. the cart is within the waypoint's circle it considers the waypoint reached and begins calculation for the next 
waypoint. This base solution for the lab only considers the requirements of reaching a waypoint and does not 
optimize the carts path to fit the criteria for other lab mentioned in this report.

This version of the waypoint driver correctly reaches all the waypoints given to it by the example data 
mapped above. It also does not exhibit the problem of circling a waypoint if it does not hit the point directly. 
The like seen in the square driver results for the p-value of .00416 the turns are even and the cart does not 
ocillate coming out of the turns. However despite the success of the algorithm in regards too the second lab it 
can be easily seen that this setup will not work for the corridor restraints of the next part. Several noticable 
problems can be seen in the base, the first being that the cart turns too late to be able to stay inside the 
corridors making even shallow turns. As seen in the results posted the major portion of every turn is made 
after the cart has past the waypoint. It is also noticeable that the turns are not sharpe enough during the fifth 
through eight waypoints for the cart to make the sharp turns required to stay within the corridors. Finally since 
the many of the turns end outside of the corridor the cart remains outside of the corridor on its way to the next 
waypoint; this is specifically seen in the driving from the eigth waypoint to the first waypoint in which the cart 
stays completely outside the path during straightaway.

Waypoint Driver Tuning: 
The roughness in the path for the base solution presents several challenges for its translation from its 
current form to the restrictions of the corridors. Since this next phase is going to consist of adding a form of 
pathfinding to the driver there are two decisions that need to be made, how much of the cart path should be
from the pathfinding algorithm and how much of it should be from the optimization of the waypoint driver. Either choice reflects on the requirements of the other options; more detail in the pathfinding algorithm will require less optimization of the driver and greater optimization will need less use of pathfinding. Since the design method decided upon reflects decisions based on only the next two waypoints it is more natural to optimize the driver and utilize less pathfinding since there is only so many points you can make in the interval. Also as always some aspects of both options are needed and while optimization can take care of almost the requirements need for this lab, even extreme versions of pathfinding can fail if the driver is unable to make the required actions. As noted in the paragraph above there were two major problems affecting the suitability of the waypoint driver, first was the start time of the turn and second the turning radius itself. The other problem with the cart not correcting back into the corridor after leaving it can be solved using the pathfinding.

Start Value: Currently the cart starts turning toward the next waypoint too late to be able to stay inside the corridor. The current setup starts the turn right as the cart enters the waypoint which is calculated by checking that the distance from the cart to the waypoint is less than the radius of the waypoint. To start the turn sooner a start distance is added mean the cart will declare the waypoint reached and the begin heading to the next waypoint before it actually enters the circle created by the waypoint. It then relies upon the forward momentum of the cart to truly reached the waypoint meaning that if too high a start value is used the cart will double fault as it will not reach the waypoint and thus leave the corridor as it goes to the next point.

Original trials of the using a start value used a static value and the results displayed below show a simulation and real world run with a static start value of two meters. This value was determined to be the best after simulation testing of the values one, two and three meters. The testing showed that starting only one meter from the waypoint was too late with the cart's turn taking it outside the corridors and three meters was too soon as the cart started cutting corners, missing the waypoint. The real world trials on the ERTS show that the start value of two is not quite optimal as while the cart only error by a small margin in some places at others it goes far outside the corridors. By looking at these points it can be determined that the cart's speed is a major factor in these errors as the points with greater error occur when the cart is heading down the slope that the test field is on thus increasing speed. Because of this it was determined to use a dynamic start value in regards to speed rather than distance in the final trial. It just so happens that normal speed of the cart under 60% power is close to two meters a second so a dynamic start value equal to the cart's speed was used meaning that the cart starts the turn one second before entering the waypoint.

P-Value: The current p-value used is the same as the one used in the square driver and while they are smooth they are not sharp enough to meet the required turns in the course. To solve this real world driving style is taken into account. Most people when making a sharper turn will oversteer in the beginning before making smaller and smaller corrections back to straight. This strategy means that rather than a static p-value a dynamic value is needed. Since the cart is not trying to form uniformed curves like in square driver greater values of p are allowed. The simulation and real world results shown uses a table with four different values for p based upon the angle of the turn which is shown in the table next to the results. Originally these results were affect by a max turning radius programed into the square driver which limited the turns to a set amount no matter how the p-value was, this was removed to allow for sharper turns needed. As the results show that the cart oversteers at the start of the turn when the larger angle dictates the use of a higher p-value but it does not oscillate as the lower p-values are used during correction. As with the implementation of the turn start value the cart's speed does have an effect on the amount of oversteering and oscillation of the cart such that both of these effects are greater at higher speeds.
Speed Control: The final optimization of the waypoint driver is the addition of basic speed control since as noted in the implementation of the start value and p-value speed plays a crucial role in the behavior of the cart. Faster movement means that more distance is covered during turns and it has less time to turn once it reaches the start point. Also because the refresh rate of the control program is static the cart will move farther between changes made by the program. The currently implemented controller utilizes a look up table which takes the difference between the targeted speed and current speed to decide the setting of the cart's throttle. It does not use the cart's brakes so great decreases in speed are not available but since the set speed and downward slope of the test field are low throttle control works just was well. By moving at a more consistent speed the real world trials reflect the simulation trials more allowing for the results to be judged more accurately in the simulator. Speed control also keeps the behavior of the cart more consistent since since it is not overshooting waypoints or oversteering as during turns. This stability in turning is greatly needed as it allows for the results of the turns to be based more on the angle of the turn rather than the speed as was seen during the implementation of the start value and p-value.

The origin implementation of the speed control increased throttle if the cart was slower than the targeted speed and cut throttle if the cart was above. The amount of increase or decrease determined by the difference between the two speeds. This showed better results than trials without speed control but still not perfect since the cart was constantly changing throttle power causing the rider to jerk back and forth as the cart accelerated and decelerated with each refresh. Also depending on if the cart was decelerated or accelerated during a portion of the turn the results were different. This led to a change in the controller to allow for giving around the targeted speed keeping a steady throttle position so long as the difference was no greater than |.3| and eliminating the many of the inconsistencies seen in the original implementation.

Pathfinding
The design method of the control program was limited to looking at only the next two waypoints reducing the usefulness of extremely detailed pathfinding methods. So instead of using a complex algorithm to produce points, only two points were calculated for the next turn. The first was the corner point of the turn which is the point on the next waypoint's circle directly opposite of the following waypoint. The other was the midpoint between the two waypoints being considered. The reason for looking at only these two points was because they provided the best turning scenario and a way to corrected the cart back to the center of the corridor during straightaways.

Corner Point: In this strategy the corner point replaces the waypoint as the target for the cart. This is done by using the formula latp2= latp1+(t)cos(a) and lonp2=lonp1+(t)sin(a) where p1 and p2 are the waypoints lat/long and corner points lat/long respectfully, t is the distance of the translation and a is the angle between the two waypoints in radians. The reason the corner point replaces the waypoint for the cart's path is that by going to the outside edge of the corridor the cart is given a lot more room in which to complete the turn with leaving the corridor. This is similar to real world driving where people will often turn toward the outside edge of the road before making a sharp turn. This decreases the sharpness of the while increase the area they have, it also allow them to oversteer the turn without going off the road.

While the optimal corner point would be on the edge of the waypoint circle having it there causes the cart to ride the edge of the corridor and some times go just outside it as seen in the results posted. To prevent this the t value is set so the corner point is not translated all the way to the edge (in the case of the example course it is set to 1.3 meters rather than 1.5 which is the static radius given to the waypoints). This decreases the area in which the cart has to turn by .2 meters but still allows for greater turn spaces than targeting the midpoint itself and provides useable results in our pathfinding algorithm. With just the corner point implemented if the cart goes outside the corridor during a turn it has no way to correct back into the corridor unless the next corner point is on the opposite side of the corridor. If the point is on the same side of the corridor as the cart is when it is outside the corridor the cart may miss the next waypoint as well as its turn takes it around the waypoint and thus never inside the circle created by it. For these reason the inclusion of the midpoint is needed to correct the cart back to the middle of the corridor before targeting the next corner.
**Mid Point:** The mid point is calculated using the pythagorean theorem to calculate the mid point of the between two way points. Since \(a^2 + b^2 = c^2\) where \(a\) is the difference in latitude, \(b\) the difference in longitude and \(c\) the length of the corridor. By taking half the value of \(a\) and \(b\) and applying them to the next waypoint the point directly between the two waypoints in question is given. By having the cart target this point after a turn it forces the cart back inside the corridor there for allowing the cart to reach the next waypoint even if its turn ended outside the corridor. There are several down sides to this however, the first being that some corridors are to short to support a mid point namely those created by points 5 through 8 in example given. Since these turns are so close together the mid point target is left out for these corridors. Another disadvantage is that the shorten distance between the two targeted points allows for less oversteering as seen in the results posted. This has its advantages though as too much oversteering will cause the cart to ride the edge of the corridor. The requirements of the lab does not put a restriction on where the cart is within the corridor so while the movement of the cart within the corridor to reach the mid point are not need the forced correction keeping the cart inside the corridor out weighs this unneeded movement.

**Conclusion:**

The final result of implementing the optimizations to the waypoint driver and adding the simple pathfinding meets the requirements of the the lab by having the cart tranverse the course of waypoints while staying within the corridors created by the waypoints. The simulator results show that the control program produces almost perfect results but as always the real world results are the important ones. As can be seen in the recording of a real world trial above where the control program guided the cart through the eight waypoint course for three laps, the cart successfully reached every waypoint. Also barring the exit of the turn at the fifth waypoint cart stayed with in the corridors created by the waypoints. This single point where the cart exit the corridor is right on top of the line defining the corridor meaning that the error was only a few centimeters. The overall speed of the course was consistant without any major increases or decreases; it also didn't have the acceleration/deceleration issues the the original speed control had. This consistancy with the cart speed kept the path of the cart consistant throughout the three laps so each lap was close to identical to the others. Since the control program operates in real time using only the next two waypoints this consistancy would allow it to be export to other waypoint course without a great deal of overhead. The use of the mid point aided in this and kept the cart near the center of the corridor during even with the oversteering coming out of turns, which it helped minimize as well. This also had the effect of not allowing for oscillation of the cart upon exiting the turns.
Despite the control programs good performance during these last series of test there are a few bugs to be worked out though they are minimal in regards to the overall results. The slight out of bounds at the exit of the fifth waypoint turn and the oversteering when exiting the the shallower turns could easily be fixed with calibrations of the speed control. For this changing from a table based controller to a PID controller would probably be give the best results. Also eliminating the allowed error between the carts heading and the targeted direction would help keep the cart on an even more consistant path and make use of the full turning area of the corner point. While the current control would work for almost any waypoint list give it would probably be best for portability to have the program read the points rather than have them hard coded in. Despite these improvements listed the current results adequate meet the requirements of the both labs covered in this report.

References: