GUIDANCE BASED MOTION PLANNING FOR AUTONOMOUS NONHOLONOMIC VEHICLE

By

Muhammad Usman Rafique

Submitted to the Department of Mechatronics Engineering

in Partial Fulfillment of the requirements for the Degree of

Master of Science

in

Mechatronics Engineering

Thesis Advisor

Dr. Kunwar Faraz

College of Electrical and Mechanical Engineering

National University of Sciences and Technology, Pakistan

2010


بسم الله الرحمن الرحيم

To my Parents

For their limitless and infinite Love and Support

To my brothers and my sister

For their never ending encouragement

To my friends

For always being there for me

ACKNOWLEDGEMENTS

I would like thank Allah Almighty for blessing me with all the blessings.

My thesis supervisor Dr. Kunwar Faraz deserves special appreciation for his support, help and guidance throughout this work. He kept the perfect balance between making me do things myself so that I can learn the most and guiding me with his experience to avoid putting effort in the wrong direction. I am thankful to him for all his guidance, suggestions, the books lent to me and the teas and coffees I had in his home while working there.

I would also like to thank department of Mechatronics Engineering, especially the head of the Department Dr. Javaid Iqbal for all the support and facilitation through the course of this work. Maj. Nasir has also been very supportive and was always there to help with any problem at hand.

I would also like to thank all the precious technical advices, reviews, suggestions and encouragement of Sheraz Bhai, Asif Ali, Sufi Ubaid, Ihsan Ullah, Abdul Wahab Butt, Maj Nadeem and Ammar Zafar.

Also, I must not forget the support of MS students in mechanical department who offered me to use their lab when I had no other option. I would like to thank, Maj Farooq, Sadia Riaz and Maj Faheem for that.

Special thanks to those who motivated and encouraged me to pursue higher studies.

ABSTRACT

Guidance Based Motion Planning for Autonomous Nonholonomic Vehicles

In this thesis a novel, fast and computationally inexpensive method of generating the autonomous trajectory for nonholonomic vehicles is proposed. The proposed algorithm is developed for autonomous parking of nonholonomic (car-like) vehicles. The proposed method called Modified Trajectory Shaping Guidance is based on missile guidance laws. The advantage of using guidance law is that it is an analytical method that requires very less computation since the trajectory is shaped by using only one parameter (lateral acceleration).

The application areas of the proposed method include autonomous parallel parking, diagonal parking, docking of mobile robots and autonomous guided vehicles (AGV). The method is robust and works for different combinations of parking space and vehicle initial position. Generated path satisfies the nonholonomic constraints on the motion of vehicle so that the vehicle can actually follow the path generated by the algorithm.

PUBLICATIONS

Following publications have been generated from the work done in this thesis:

1. M. Usman Rafique and K. Faraz, “Guidance Based Autonomous Parking Assistant”, IEEE International Conference on Signal Processing, Robotics and Automation (ICSRA), China, 2010

2. M. Usman Rafique and K. Faraz, “Modified Trajectory Shaping Guidance for Autonomous Parallel Parking”, IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Singapore, 2010

Contents

List of Figures

1 INTRODUCTION

1.1 Motivation

There are number of applications of an efficient algorithm for path planning of parking or docking of autonomous vehicles. Such algorithms become complex when there are constraints on motion of the vehicle i.e. nonholonomic constraints as are in the case of car-like vehicles. Autonomous guided vehicles in industries, mobile robots and vehicles need to park or dock frequently. The utilization of these vehicles can clearly be enhanced if this task is done autonomously. Even the latest generation of automobiles is expected to be equipped with a parking assist system. In such a system, once the driver switches into the parking assist mode (at a starting location) the car-parking controller generates the target trajectory and controls that are required to move the vehicle into the parking space. The difficulty of this problem increases significantly when a complex parking maneuver is required. Therefore, there has been significant research in this area and many solutions have been proposed, but the problem with current solutions is that they are computationally very expensive and cannot be used on the vehicle for real-time online computation without very expensive computations.

In this thesis, we propose a new method to address this problem which is computationally very inexpensive. The proposed method is based on Trajectory Shaping Guidance (TSG), a missile guidance law. The advantage of using TSG is that it is an analytical method that requires very less computation since the trajectory is shaped by using only one parameter (lateral acceleration). In the past TSG has been extensively used by missiles to shape their trajectory to maximize lethality. The basic missile control principle is that the missile always moves in the desired heading and the direction is changed by lateral acceleration. This principle is well suited for nonholonomic vehicles because they also move in a fixed direction and require steering control to change their direction. This similarity has been leveraged to plan the motion of the vehicle in such a way so that they achieve desired angle of

approach to the target i.e. the vehicle reaches the parking space in the desired direction, as is desired in parking maneuver.

1.2 Literature Review

Guidance is the term for path planning when the target is moving. Guidance laws have been used in weapons (guided weapons and missiles) as early as 1916, as discussed by Shneydor in [1]. Also, guidance laws have been well defined and documented in field of missiles. These guidance laws have already been successfully applied for motion planning of mobile robot, for both navigation (when target is fixed) and guidance. There has been much work in this field in University of Toronto. Particularly, Beno Benhabib and K. Faraz [2 - 5] have done lot of work in this field. Some of the application areas in of robotics and motion planning where guidance laws have been successfully used are discussed below.

An application of guidance laws is the Interception of Mobile Targets [2]. Guidance based Online Robot Motion Planning method has been used for interception of Mobile Targets in Dynamic Environments.

Another application area is the autonomous highway overtaking by vehicles [3]. This work also deals with the guidance based laws which are applied in dynamic environment. The method works online and doesn’t require a preplanning and re-planning.

Guidance laws have also been used for the purpose of Rendezvous with moving objects [4]. Rendezvous is the problem in which it is desired to pursue the target such that upon reaching the target, the velocity of pursuer (e.g. vehicle) should match with the moving target. In this work, a time-optimal rendezvous method is used which deals with dynamic environment i.e. obstacles are also moving.

Rendezvous of a vehicle with a moving vehicle convoy [5] has also been implemented using guidance based laws. This again is a motion planning method used by the vehicles which deals with dynamic environment and has the ability to work in real-time application.

However, we see that no guidance law has yet been applied for the problem of autonomous parking or docking.

However, over the period of years a significant amount of work has been done on the problem of autonomous parallel and diagonal (or row) parking using conventional methods. Dubins in [6] first developed a path planning strategy for nonholonomic vehicles. However, this approach could not be used to compute motion in which vehicle could move in the reverse direction. To overcome this limitation, [7] proposed an optimal path planning method in which the vehicle could move both in the forward and the reverse direction.

The next significant advancement in the field of autonomous parking was the emergence of motion planning techniques. An open-loop path planner and feedback tracking controller for parking was explored in [8]. Kim and Chung in [9] used slice projection technique for path planning. Navigation field is used by [10] to develop a vehicle-parking assistant. Advanced numerical methods and artificial potential fields were introduced by [11]. Use of path planning and continuous curvature trajectory planning is presented in [12]. Concatenation of line and polar spline segments were used by [13] for path planning of autonomous fork lift truck. Although these methods provided a sophisticated solution to the autonomous parking problem yet they proved to computationally very expensive and as such could not be used effectively in a real-world scenario.

Another approach to solving the parking problem is the use of Control theory. Posture stabilization was used for autonomous parking by [14]. Switching control algorithms for chained systems were applied by [15] and [16]. Sinusoidal reference functions for the purpose of parking are given in [17]. Optimal control was implied by [18] for optimization of car parking. In [19], open and closed-loop control algorithms. However, the drawback with Control Based Techniques was that they require complex modeling and tedious computations.

In the more recent past, artificial intelligence and machine learning has also been applied to the problem. Genetic algorithms, petri-nets and fuzzy controllers were used for parking in [20]. In [21] an [22], fuzzy logic is used. Genetic fuzzy

system has been used for optimization of fuzzy logic controller in [23]. The advantage of using AI based applications was that there was no requirement to develop a complex vehicle model. However, the complex design and learning requirements of the controller increases the computational expense.

1.3 Research Objective

To develop an efficient, fast and computationally inexpensive method that generates the path for autonomous nonholonomic vehicle. The proposed algorithm is to be developed for autonomous parking of nonholonomic vehicles. The method should be robust and should work for different combinations of parking space and vehicle initial position. Generated path must satisfy the nonholonomic constraints on the motion of vehicle so that the vehicle can actually follow the path generated by the algorithm.

1.4 Thesis Organization

The rest of the report has been organized as follows. Chapter 2 gives the summary of proposed algorithm and explains its basic components. Nonholonomic vehicle model is also included in chapter 2. Guidance law modified, known as Trajectory Shaping Guidance, and corrections and additions in it are elaborated in chapter 3. Implementation of proposed method for Diagonal and Parallel Parking is discussed in chapter 4, which also includes the simulation results of different scenarios. Chapter 5 concludes the work done in this thesis and throws light on future work that can be done in this area.

2 SYSTEM OVERVIEW

In this work, we have developed a new guidance law, which is, in fact, built on an existing guidance law named Trajectory Shaping Guidance [24] and [25]. This method was originally used by missiles to hit the target at a desired angle. Here it used for motion planning such that the vehicle reaches a point from a certain desired direction. The new modified method developed in this work is called the Modified TSG.

Schematic diagram of the proposed Modified TSG is given in Fig. 1. For motion planning of complete parking maneuver, we assume the environment, vehicle initial position and parking space location are known a-priori. The ability of TSG to generate a trajectory given the pursuer initial position, target position and final angle of intercepting the target is used here. As shown in Fig. 1, angles are computed autonomously by the algorithm using knowledge of position of the vehicle and the parking space. It was observed that sometimes these lead to undesirable trajectories. To solve this problem angles are corrected and then TSG is used to generate points of trajectory.

Every iteration of TSG algorithm gives the next point on the desire trajectory. New trajectory point computed is tested for nonholonomic constraints, using knowledge of vehicle model and nonholonomic constraints to which vehicle is subjected, which are discussed in coming sections in detail. If the new computed point on trajectory is beyond the reach of vehicle due to nonholonomic constraints, it is changed to a point feasible for vehicle to reach despite of motion constraints on it.

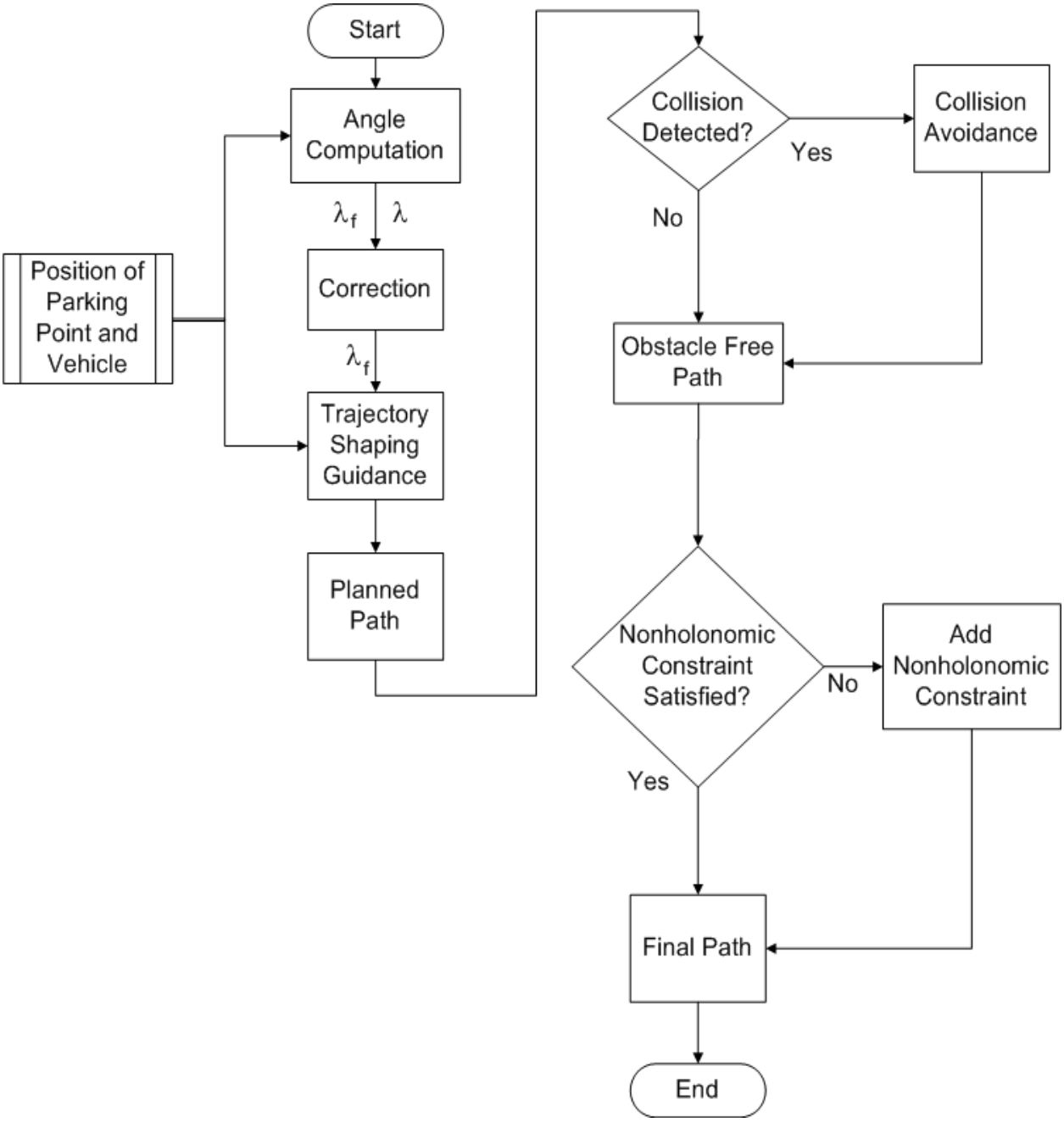


Figure 1: Schematic Diagram of Modified TSG

2.1 Vehicle Model

Here we consider a 4 wheeled car-like vehicle which is modeled as shown in Fig. 2. L is the wheelbase i.e. distance between front and rear wheels. The position

of vehicle, P, is taken at centre point of rear wheels. θ is the heading of vehicle and ϕ is angle of wheels from vehicle direction. The motion of vehicle is governed by following equations.

$$\dot{P}_X = V_P \cos(\phi)\cos(\theta) \tag{2.1}$$

$$\dot{P}_Y = V_P \cos(\phi)\sin(\theta) \tag{2.2}$$

Direction:

$$\dot{\theta} = \frac{V_P}{L}\sin(\phi) \tag{2.3}$$

Steering angle:

$$|\phi| \leq \pi/2 \tag{2.4}$$

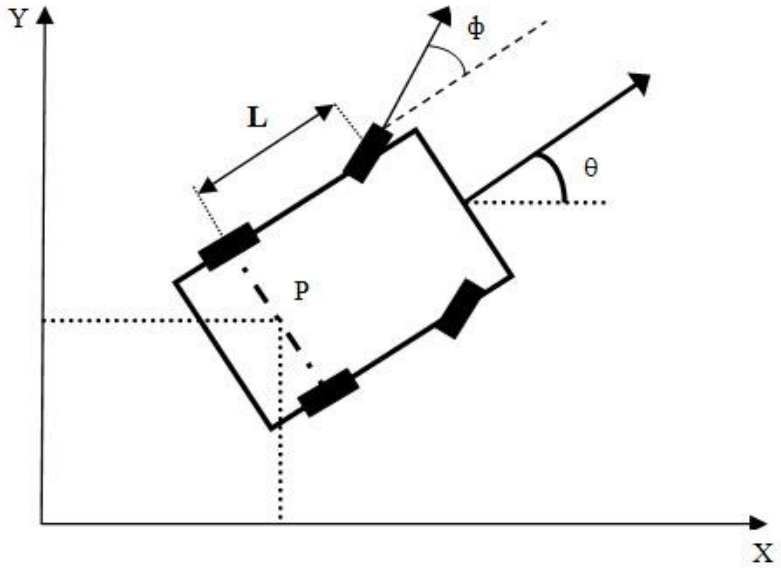


Figure 2: Vehicle Model

3 MODIFIED TRAJECTORY SHAPING GUIDANCE

Trajectory shaping guidance was developed and used by missiles to hit the target at the desired angle to enhance lethality. However, we observed that when autonomously used to generate path for planning, there were some problems due to computation of angles. Also, sometimes, the nonholonomic constraints were violated by the trajectory generated. To solve these problems, TSG was modified, details of which are given below. The modified TSG takes input of Vehicle Initial position and orientation, target point and desired angle of approach and returns a path which has the states of vehicle i.e. its position as well as its orientation.

Section 3.1 briefly describes the original TSG, 3.2 shows the solution of problem caused by computation of angles. In section 3.2, nonholonomic constraints are discussed.

3.1 Trajectory Shaping Guidance

Trajectory shaping guidance is a guidance method developed for guided missiles to hit the target at the desired angle. This law issues lateral acceleration command perpendicular to the instantaneous missile-target line of sight (LOS). The system is shown in Fig. 3 where P is the position of Pursuer and T is the position of Target. λ is angle of LOS, λ_f is desired angle of approach. R is range vector from point P to T. The lateral acceleration command issued by trajectory shaping guidance method is given by:

$$n_C = 4V_C\dot{\lambda} + \frac{2V_C\left[\lambda - \lambda_f\right]}{t_{go}} + n_T \tag{3.1}$$

Here t_{go} is time to go until interception and V_c is closing velocity which is computed by

$$V_C = \frac{-\left[R_X V_{TP_Y} - R_Y V_{TPX}\right]}{\|R\|} \tag{3.2}$$

R_X and R_Y are X and Y components of Range vector R respectively. V_{TPX} and V_{TPy} are components of V_{TP}, given by

$$V_{TP} = V_T - V_P \tag{3.3}$$

Time to go is computed by

$$t_{go} = \frac{\|R\|}{V_C} \tag{3.4}$$

Since in our case target (parking space) is stationary, we use $V_T = 0$ and $n_T = 0$

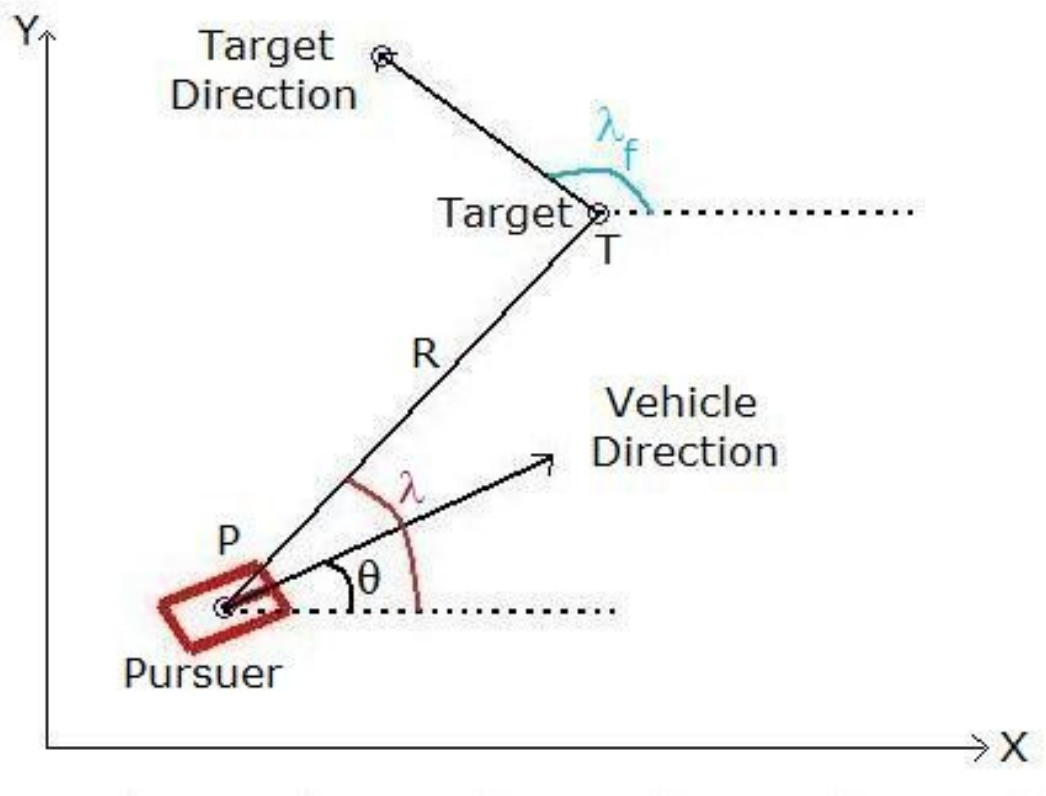


Figure 3: Trajectory Shaping Guidance

Heading Error is defined as:

$$HE = \theta - \lambda \tag{3.5}$$

3.1 Angle Correction

The major drawback in implementing TSG for the autonomous parking or docking problem arises when λ_f and λ are automatically computed by the algorithm. This at times leads to the generation of non-optimal trajectory for the intercept. For example, if T is the position of target, P is initial position of vehicle and Z is a point in desired direction of approach, shown in Fig. 4. Range vector R is given by

$$R = T - P \tag{3.5}$$

Direction vector is found by

$$D = Z - T \tag{3.6}$$

In this case the angles λ_f and λ are calculated by:

$$\lambda = \tan^{-1}\left(R_Y / R_X\right) \tag{3.7}$$

$$\lambda_f = \tan^{-1}\left(D_Y / D_X\right) \tag{3.8}$$

To compute the angles the atan2 command is used. However, since the interval of atan2 is [-π , π] therefore it returns both negative and positive angles. Commanded pursuer acceleration, that governs trajectory shape, given in Eq. (3.1), uses the expression "$\lambda - \lambda_f$". However, different positive and negative equivalent combinations of λ and λ_f can yield different values of the expression "$\lambda - \lambda_f$" for same angles. This in return generates different trajectories using the TSG. This sometimes leads to undesired trajectory; an example of such a case is presented diagrammatically in Fig. 4 and Fig. 5. The result of original TSG with angles computed by above equations is shown in Fig. 4. This figure clearly shows that using the original law leads to the generation of an undesired trajectory. All such undesirable cases were studied in detail. Based on this analysis, a new modified TSG law is proposed.

The modified TSG law converts λ_f to either its equivalent negative or positive angle using the following equations. This in return guarantees the generation of the desired trajectory. This can be mathematically expressed as:

$$\text{IF } (\lambda_f < \lambda - \pi) \text{ THEN } \lambda_f = \lambda_f + 2\pi \tag{3.9}$$

$$\text{IF } (\lambda_f > \lambda + \pi) \text{ THEN } \lambda_f = \lambda_f - 2\pi \qquad (3.10)$$

After using this correction, the trajectory generated for same scenario is shown in Fig. 5.

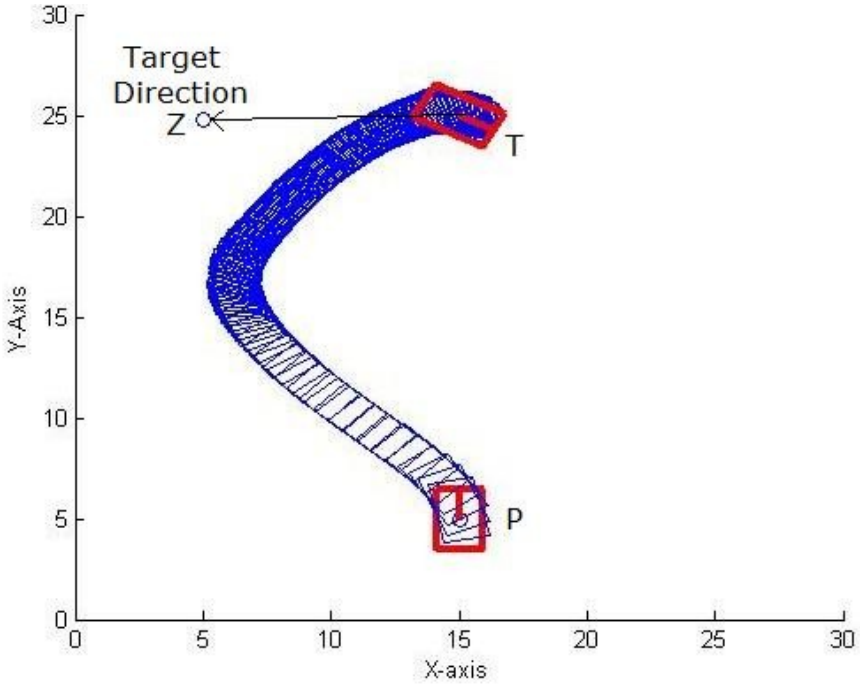


Figure 4: Original TSG Result

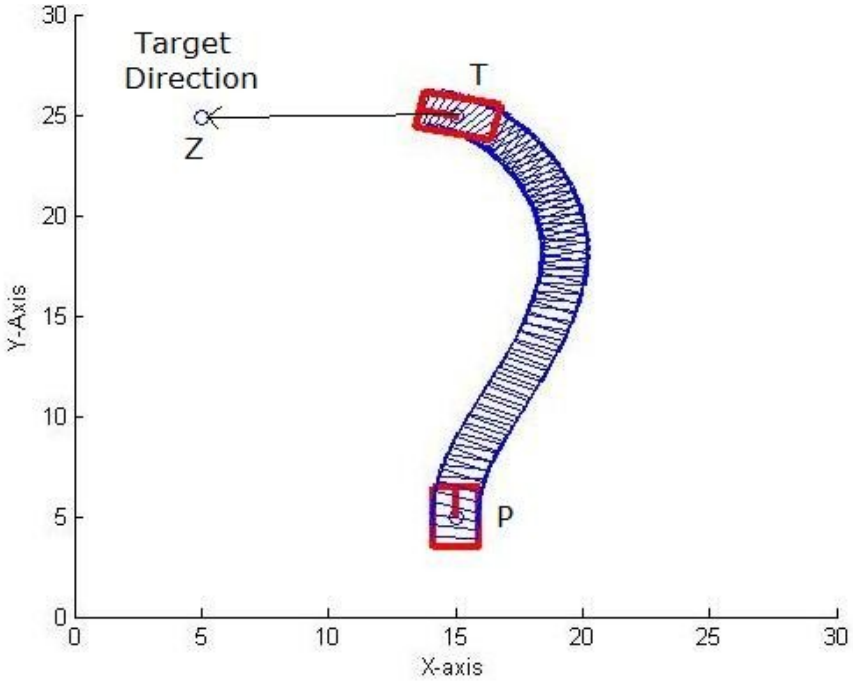


Figure 5: Modified TSG Result

3.2 Obstacle Avoidance

After correcting angle, which ensures a desired trajectory, the algorithm then makes sure to avoid any present obstacle. In this work, for application on autonomous parking, we have considered the obstacle to be static i.e. not moving. A brief discussion on selecting the method for obstacle avoidance is given below.

We know there are many conventional methods for obstacle avoidance used in motion planning techniques for similar scenario i.e. fixed target point and static obstacle. A few of these methods are:

1. **Tangent Bug Algorithm**

   Unlike basic bug algorithms, which use tactile sensing, this method uses range sensing. This method requires only local knowledge of the environment, and of course can be useful even of whole environment is known. This method governs when and how much to follow the boundary of obstacle and when to move towards the goal.

2. **Artificial Potential Fields**

In this method, we need global knowledge of the environment. The goal or target generates an artificial attractive field and the obstacles generate repulsive field. Based on these two types of field, a resultant field is computed which helps navigation. The robot or vehicle moves in the direction of minimum resultant field i.e. the net attractive force acting on it.

3. Navigation Field

The artificial potential fields method discussed above, has resultant field based on an attractive and some repulsive fields (equal to number of obstacles). This resultant field should ideally have only 1 minimum i.e. the goal. But in fact, the potential field method suffers from problem of local minima, which are points that are not goal but have locally minimum resultant field and the robot or vehicle gets stuck in it because it cannot move to a direction of minimum field anymore.

To avoid this problem, a method namely Navigation Field is used. This method also generates an artificial field, governed by its analytical relations by using knowledge of goal and obstacles. This method ensures to have only 1 global minimum.

4. Cell Decomposition

In cell decomposition method (which has further specialized versions Exact Cell Decomposition and Approximate Cell Decomposition) is another method that requires global knowledge of the environment. In this method the environment is broken down to smaller cells and the process is repeated till we have a path by joining cells which do not have any obstacle in them from robot or vehicle to the goal point.

5. Roadmaps

Methods like Cell Decomposition use knowledge of that particular scenario to compute a feasible path from start to goal point. For different scenarios, the whole process is repeated every time which is not efficient if a vehicle has to plan its path frequently in a given environment.

To address this problem, Roadmaps method is used. In this method, we build a roadmap of the given environment, which is similar to the concept of roads and highways used by humans. Once a roadmap is

built for an environment, then for every path planning scenario, we connect the vehicle start position with roadmap. The path of vehicle is planned on the roadmap till the point which is close and can be connected to the goal position.

6. **Probabilistic Roadmaps**

   In probabilistic roadmaps, we don't use a fixed computational method to generate a roadmap. Rather we generate random possible samples or configurations or positions of the vehicle. The points which are found to be inside an obstacle are removed. We keep adding random points or samples till a path from start to goal is found by joining these points.

The methods discussed above all are for the scenario similar like ours, but there is a basic difference in our case i.e. we don't just want to reach the goal, rather we want to reach the goal from a certain direction as well. This is not addressed by planning methods discussed above. And for nonholonomic vehicles, this becomes a rather different task to generate such a method by avoiding obstacles.

Also there is another problem of keeping the desired trajectory shape. It is optimal for nonholonomic vehicle to have desired trajectory shape for ease of planning and optimality of length of trajectory. These above mentioned methods work independently and cannot be used with our method which itself governs the shape of trajectory.

So instead of going with some navigation planning method, we have opted to choose an obstacle avoidance method which deals with velocities. The obstacle avoidance used here is built on a method named "**Velocity Obstacle Method**" [26]. It takes an input the current position of the vehicle, its velocity and information of obstacles and returns a feasible velocity which ensures that there is no collision with obstacles. This method has easily been made compatible with the Modified TSG by selecting an avoidance velocity which is closer to original velocity computed by the Modified TSG. This ensures that the obstacle is avoided with minimum deviation from the original trajectory and hence the trajectory can be as desired after the obstacle has been avoided.

Implementation of the obstacle avoidance method used here is given below.

3.2.1 Implementation

There are following steps of the complete obstacle avoidance method.

1. **Conversion to Configuration-Space**

   To check collision of the vehicle is difficult because we will have to check intersection of whole vehicle area with area covered by obstacle. So for such problems, most commonly used solution is to convert the problem into Configuration Space or C-Space.

   In this conversion, we:

   a. Convert the vehicle into point vehicle

   b. Increase radius of obstacles as much the vehicle's dimensions are reduced. Here, the length of vehicle is larger than its width (which is true for most of car-like nonholonomic vehicles). We increase the obstacle radius equal to the length of vehicle, because that is the larger dimension and it will give some safety margin as well. If Ro is the original radius of the obstacle, new radius of obstacle in C-Space will be 'Ro + Ln' where Ln is the length of the vehicle.

   Conversion to C-Space is shown in Fig. 6 below. The inner circle is the original obstacle in the workspace. The new c-space obstacle is the one shown in the outer circle. Also, we see that the vehicle is reduced to a point, shown in center of vehicle.

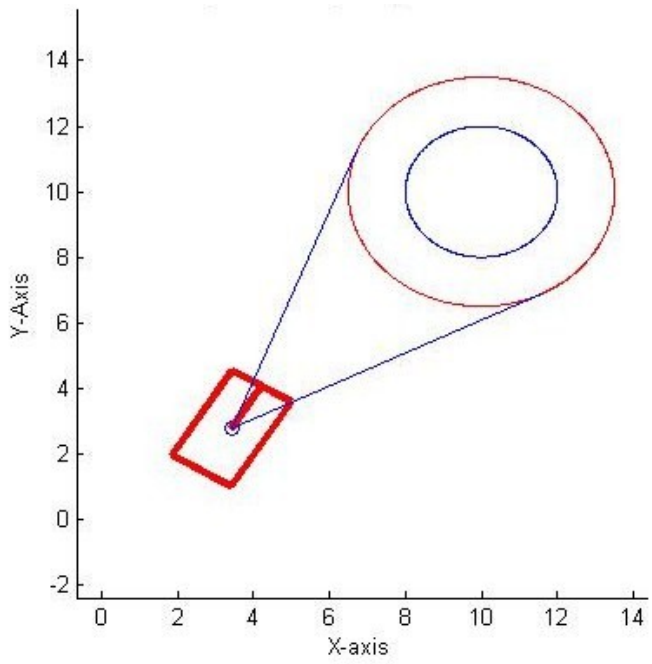


Figure 6: Obstacle Avoidance - Velocity Obstacle

2. Collision Cone

Next, a collision cone is made. It is the cone defined by the tangents drawn on new c-space obstacle from the point vehicle. A collision cone for the same case is shown in Fig. 6

3. Collision Detection

Original Velocity Obstacle method deals with relative velocity of the vehicle and the obstacle (since it assumes moving obstacles). But in our case, the obstacle is static, so we have a rather simple method for collision detection. If the velocity of the vehicle generated the Modified TSG lies within the collision cone, there is going to be a collision. If the generated velocity is outside the collision cone, there will be no collision.

4. Collision Avoidance

If a collision in predicted, the obstacle avoidance method changes the original velocity generated by the modified TSG such that the trajectory is deviated as less as possible, to ensure reaching the target point at desired direction for successful parking or docking application. The method used here changes only the direction of velocity and not its magnitude. In the original velocity obstacle method, a set of

feasible velocities is computed and then an avoidance velocity is selected based on some heuristic. But in our case, we have a separate check for nonholonomic constraints, explained in next section, we don't need to find out a set of feasible velocities. We just chose the nearest possible velocity which guarantees obstacle avoidance. So, one of the 2 tangents we have drawn to make a collision conies chosen as the final velocity to avoid obstacle. This is diagrammatically shown in Fig. 7.

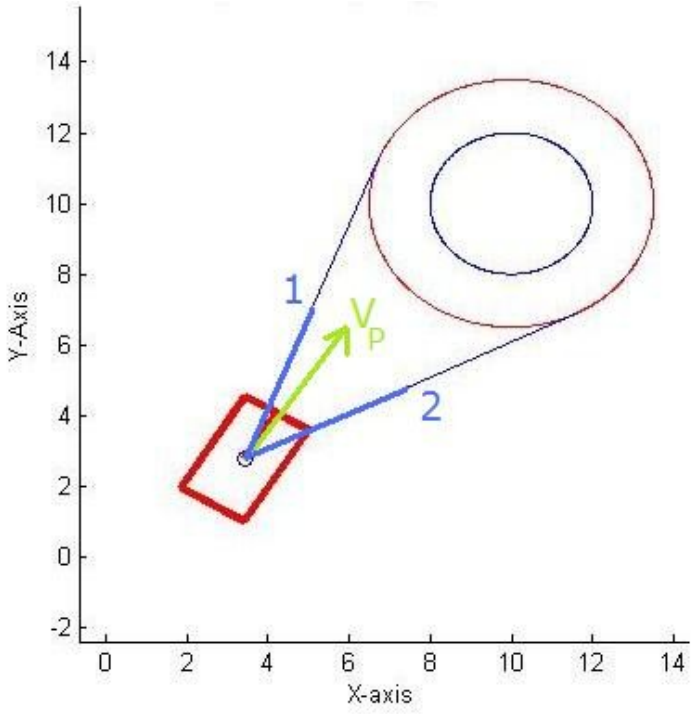


Figure 7: Obstacle Avoidance - Velocity Selection

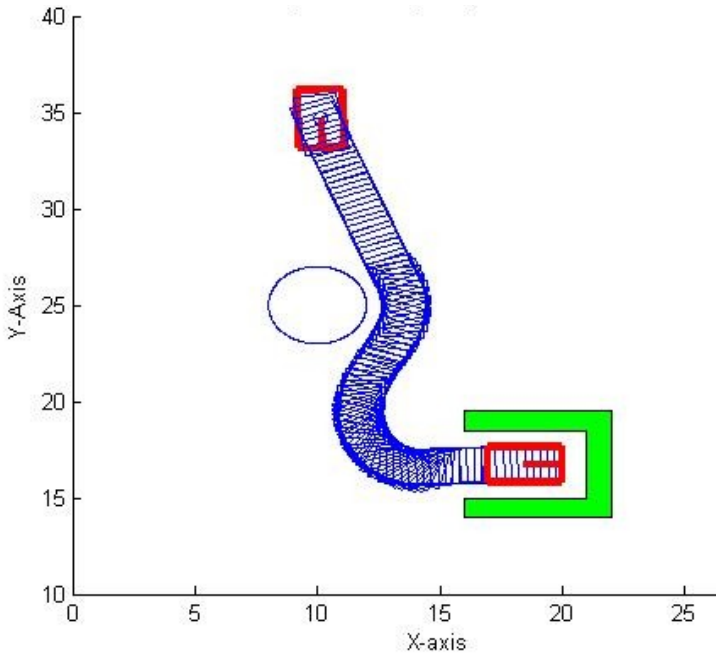
Tangents labeled 1 and 2 are drawn to make the collision cone. We see that the original vehicle velocity V_P is inside the collision cone, so a collision is going to happen. For the sake of avoidance velocity, we see the angle of V_P with tangent 1 and tangent 2 and the one having smaller angle with V_P is closer and hence is chosen. As shown in Fig. 7, tangent 1 will be selected as the new velocity direction. This will make sure that there is no collision having caused the minimum deviation.

An important point to note here is that in step 1, when we converted the Vehicle to the point vehicle, we used its bigger dimension (i.e. length) to add to the radius of the obstacle, so there was already a

safety margin added there. So here we set velocity in direction of the tangent without giving it safety margin as it has already been provided.

3.2.2 Obstacle Avoidance Results

Based on the method explained above, the results of a few cases of obstacles are presented in Fig. 8 and Fig. 9.



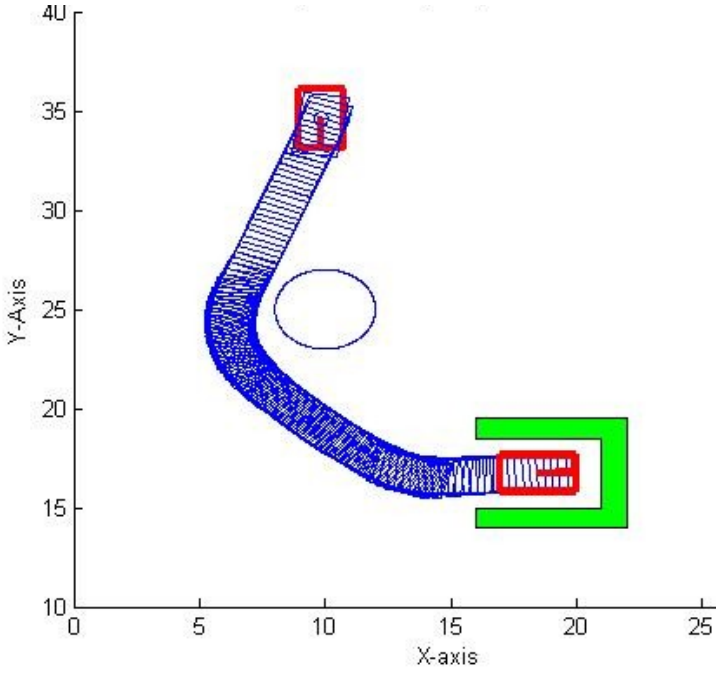
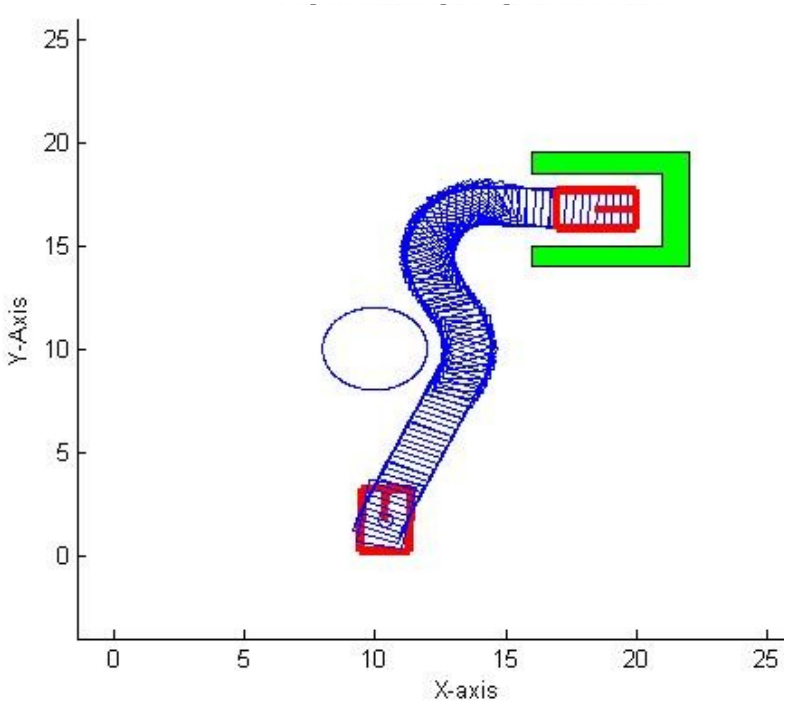


Figure 8: Obstacle Avoidance – Results



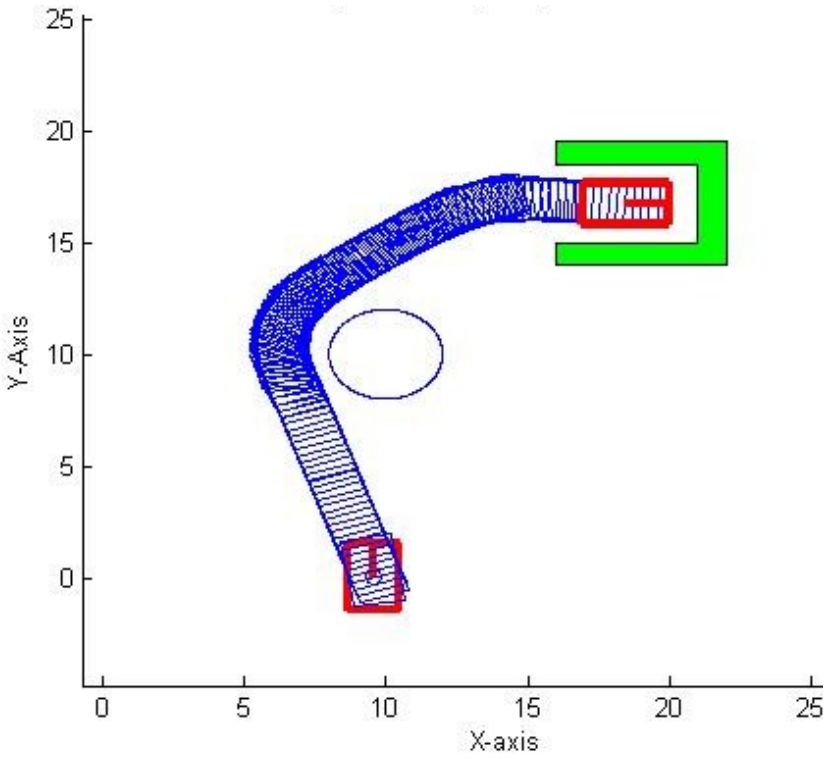


Figure 9: Obstacle Avoidance - Results

3.3 Nonholonomic Constraints

In order to implement the modified TSG law on vehicles, the nonholonomic constraints are checked continuously. If the desired trajectory point cannot be reached by the vehicle due to nonholonomic constraints, a new point is generated which is accessible from vehicle's current state. From Eq. (2.3):

$$\dot{\theta} = \frac{V_P}{L}\sin(\phi) \tag{3.11}$$

ϕ is steering angle which is $|\phi| \le \pi/2$. For maximum turn, $\phi_{\max} = \pi/2$. Applying this, Eq. (3.11) reduces to

$$\dot{\theta} = \frac{V_P}{L} \tag{3.12}$$

where L is the wheelbase and V_P is vehicle velocity (computed in every iteration of the algorithm). The distance moved by vehicle after one iteration is its velocity. Therefore,

$$\dot{\theta}_{\max} = \frac{d}{L} \tag{3.13}$$

where d is distance between P_t and P_{t+1}, shown in Fig. 10. This expression gives the maximum rate of change of vehicle angle in one iteration.

Fig. 10 shows position of vehicle P_t at time *t*, the next point on the trajectory i.e., P_{t+1} at t+1 is calculated by TSG. The direction θ_{t+1} is the angle of vector between P_t and P_{t+1} i.e. current heading of the vehicle is essentially the direction in which it has just moved in its last step.

$$\dot{\theta} = \theta_{t+1} - \theta_t \tag{3.14}$$

If $\dot{\theta} \le \dot{\theta}_{\max}$, the generated point and direction are feasible based on nonholonomic constraints. However if $\dot{\theta} > \dot{\theta}_{\max}$, the point cannot be reached from current state. So, in such cases we don't consider the new computed point and compute a new point which satisfies the nonholonomic constraints. To compute corrected point, we reduce $\dot{\theta}$ and ensure that its value does not exceed the maximum possible value (i.e. θ_{max}). The corrected heading is now computed using the relation

$$\theta_{t+1} = \dot{\theta} + \theta_t \tag{3.15}$$

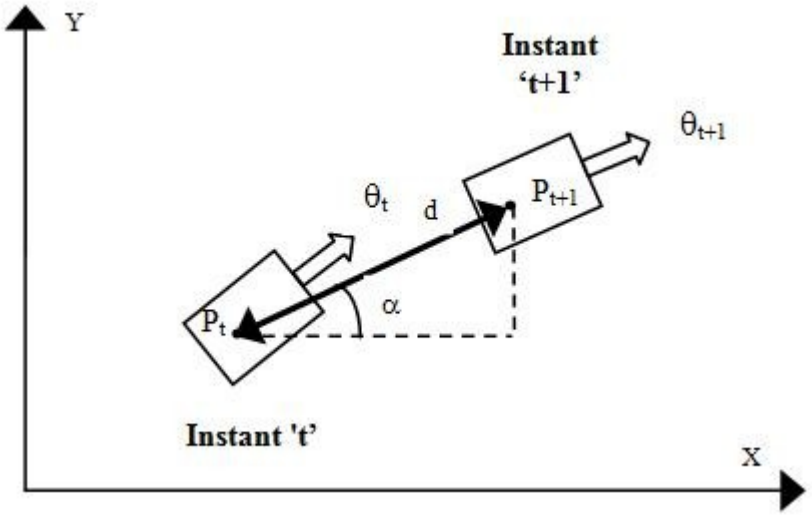
From Fig. 10

$$\alpha = \theta_{t+1} \tag{3.16}$$

New trajectory point will be

$$P_{t+1} = P_t + d[\cos(\alpha) \quad \sin(\alpha)] \tag{3.17}$$

Figure 10:



Nonholonomic Constraints

4 IMPLEMENTATION / APPLICATION

The Modified TSG explained in Chapter 3 has been applied for typical docking or parking problem. There are two types of parking problems, Diagonal or Row Parking and Parallel Parking. The implementation of the modified TSG for both Diagonal and Parallel Parking is explained in subsequent sections.

4.1 Diagonal or Row Parking

Here, the parking space is represented by a rectangle specified by four points H, I, J and K as shown in Fig. 11. Opening of parking space is specified by points H and K. P is the initial position of the vehicle.

We consider that initial position of the vehicle P, its initial orientation and the parking space specified by 4 points is known. Based on this knowledge, points, A, B and C are computed by the relations given below:

$$A = \frac{1}{4}\left(H + I + J + K\right) \tag{4.1}$$

$$B = \frac{1}{2}\left(H + K\right) \tag{4.2}$$

$$C = B + \varepsilon \cdot \hat{IH} \tag{4.3}$$

where $\hat{IH}$ is the unit vector from I to H and ε is the safety distance.

It is observed that TSG generates desired trajectory if the heading error, given by Eq. (3.5), is limited to certain value. If heading error is beyond that value, undesired path is generated. To solve this problem, we use the Modified TSG twice to generate the whole path, breaking into 2 such that heading error is also reduced. The vehicle first goes to an intermediate point, Q, and then goes to parking space. Point Q is computed such that it reduces heading error into half without increasing length of path. Based on heading error, following two cases are developed.

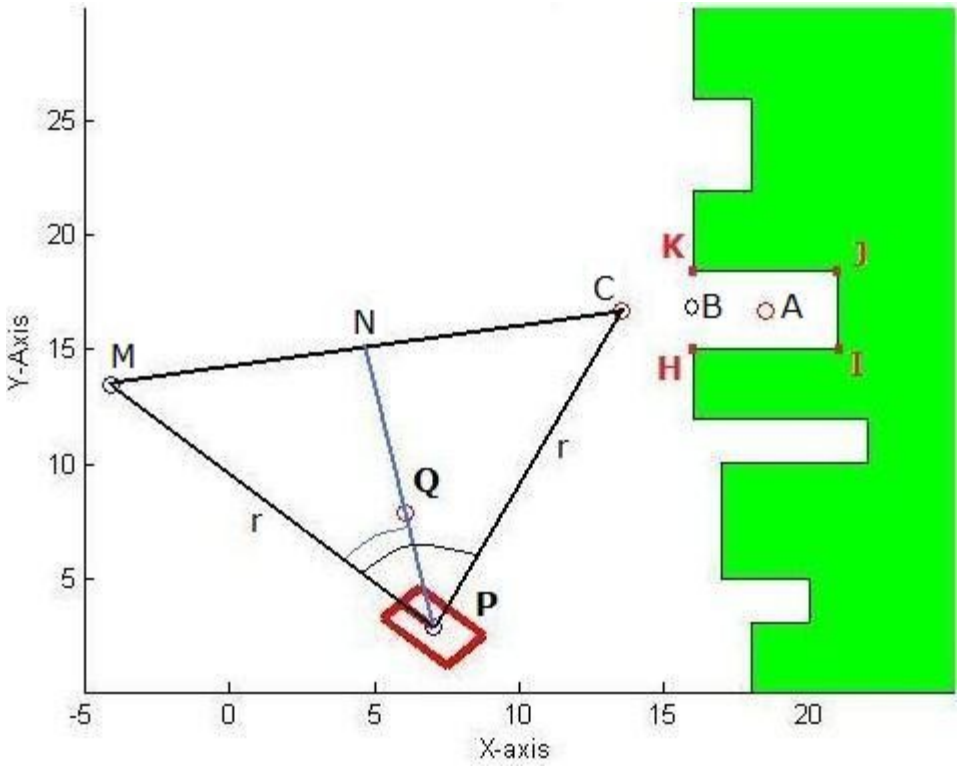


Figure 11: Using MTSG for Diagonal Parking

As mentioned earlier, the modified TSG takes as input the vehicle initial position and orientation, target point and desired angle of approach. So, we compute the target points and their respective angles of approach which complete the whole diagonal parking manoeuvre.

Case 1: Heading Error ≤ 70⁰

Step1 – The vehicle goes to C with λ_f given by

$$\lambda_f = \tan^{-1}(HI_Y / HI_X) \tag{4.4}$$

where HI is the vector from H to I.

Step2 – The vehicle goes to A with λ_f given by

$$\lambda_f = \tan^{-1}(HI_Y / HI_X) \tag{4.5}$$

Case 2: Heading Error > 70⁰

Step1 – The vehicle goes to Q with λ_f given by

$$\lambda_f = \tan^{-1}(PC_Y / PC_X) \tag{4.6}$$

Point Q is computed by following relations:

$$M = P + r[\cos(\theta) \quad \sin(\theta)] \tag{4.7}$$

where r = ||PC||.

$$N = \frac{1}{2}(M + C) \tag{4.8}$$

$$Q = P + \left(\frac{r}{3}\right) \cdot P\hat{N} \tag{4.9}$$

Step2 – The vehicle goes to C with λ_f given by

$$\lambda_f = \tan^{-1}(HI_Y / HI_X) \tag{4.10}$$

Step3 – The vehicle goes to A with λ_f given by

$$\lambda_f = \tan^{-1}(HI_Y / HI_X) \tag{4.11}$$

4.1.1 Results

Extensive numbers of simulations were conducted to test the performance and robustness of the proposed modified TSG law. All simulations were performed using MATLAB. In these simulations, all possible combinations of parking spaces and starting positions of the vehicle were used. A small subset of results is shown in the Fig. 12 and Fig. 13. For the purpose of simulations typical automotive dimensions were considered i.e. Length = 3m, Width = 1.8m, Wheel base = 1.8m.

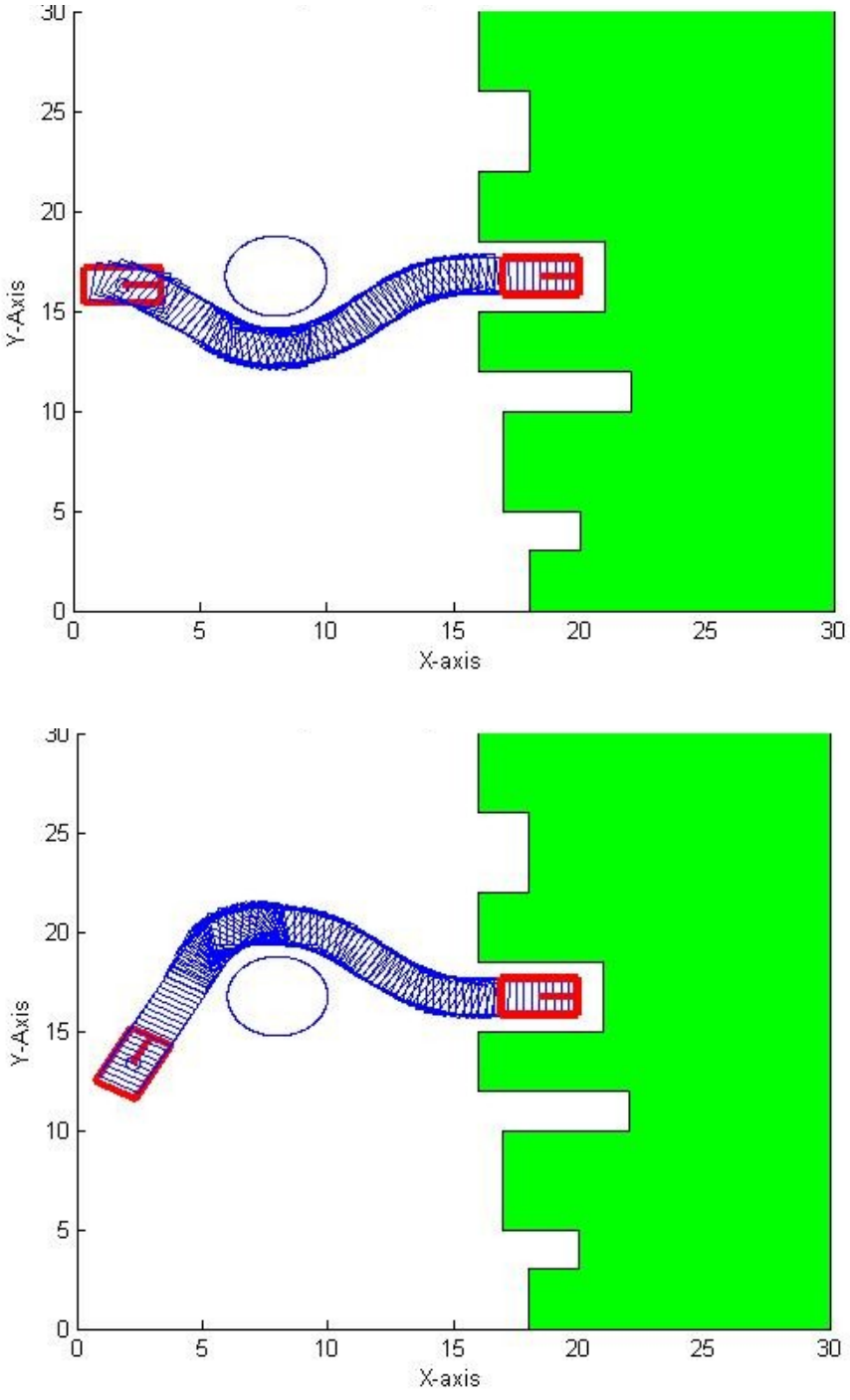


Figure 12: Results of Modified TSG for different Scenarios

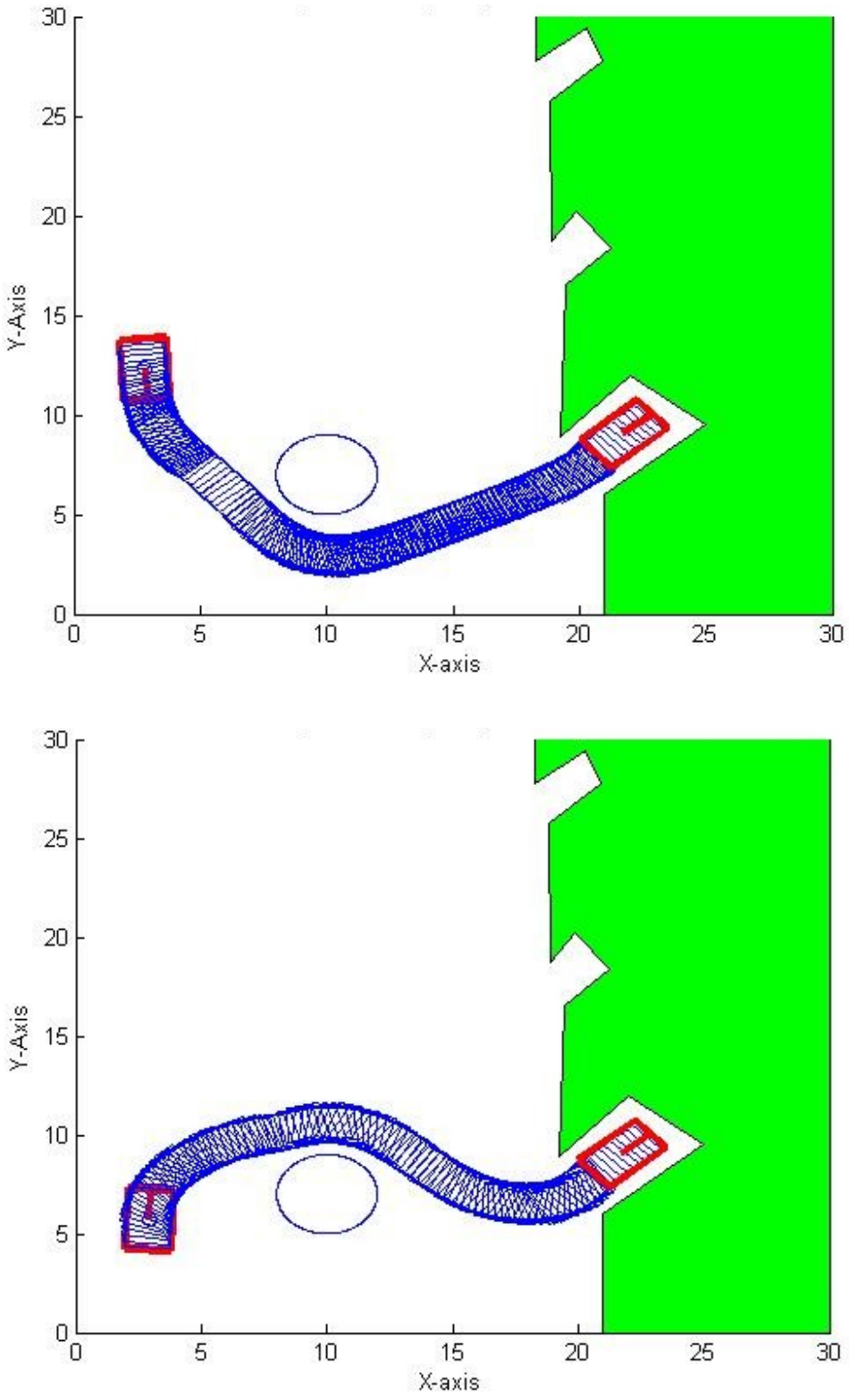


Figure 13: Results of Modified TSG for different Scenarios

4.1.2 Implementation on ASG Faculty Car Parking

The proposed method was applied on model of AG Faculty Car Parking of College of Electrical and Mechanical Engineering, NUST. The car park was modeled in MATLAB according to its actual dimensions. The vehicle starts at the entrance of the parking area and simple function of moving straight is used by the vehicle till it reaches close to the assigned parking slot, and then it uses the modified TSG for parking maneuver. The developed model and target points computed from knowledge of that model are shown in Fig. 14.

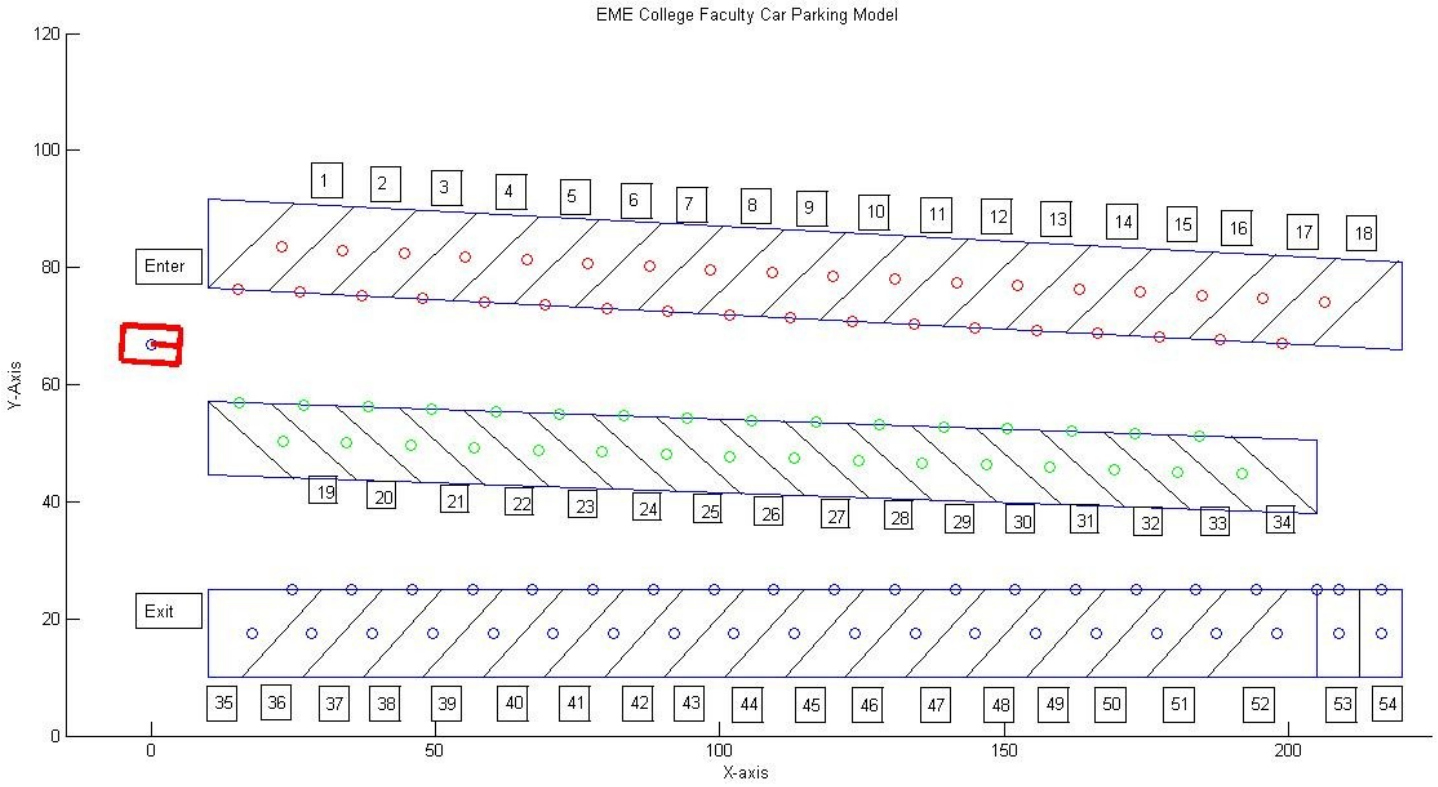


Figure 14: ASG Faculty Car Parking Model, EME College

Available parking slots are numbered for all 54 parking slots. For every parking slot, 2 points are computed, as shown in Fig. 14, to complete the parking maneuver. First point is the center of opening of that slot, and second point is center of that parking slot.

Since the environment is known a-priori, the desired angle of approach (λ_f) for each slot is computed in the beginning of the simulation. We see, from Fig. 14, that slot 1 – 18 have same direction hence same λ_f. Similarly slots 19 – 34, slots 35 – 52 and slots 53 and 54 have same λ_f.

First the vehicle goes to the outer point using the modified TSG and to the central point of that slot again using modified TSG. Reason for using an outer point as well and not relying only on the central point of that slot is to avoid collision. There might be a car parked in the next slot, so first using modified TSG for outer point ensures that no part of vehicle enters any neighboring slot and hence avoids any collision. For the sake of simulation, the user enters the desired slot number and the complete trajectory, from entrance of parking to the desired slot, is generated and displayed.

Also, for slot numbers 35 to 54, a turn is required. Even this turn has been achieved by using the modified TSG, giving it the final point and orientation as input.

Some selected results for different slots are shown in Fig. 15 and Fig. 16.

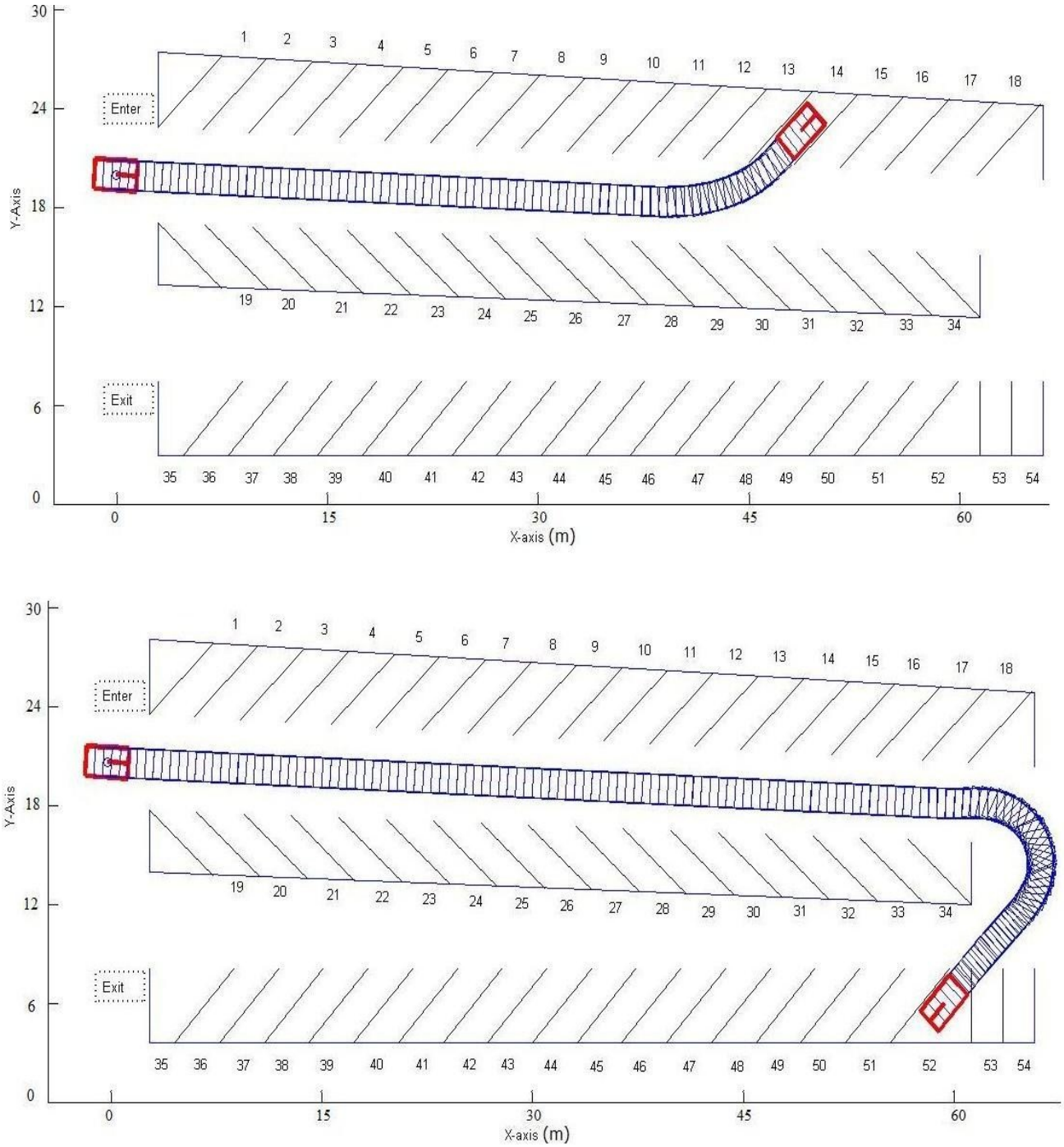


Figure 15: Results ASG Faculty Car Parking EME College

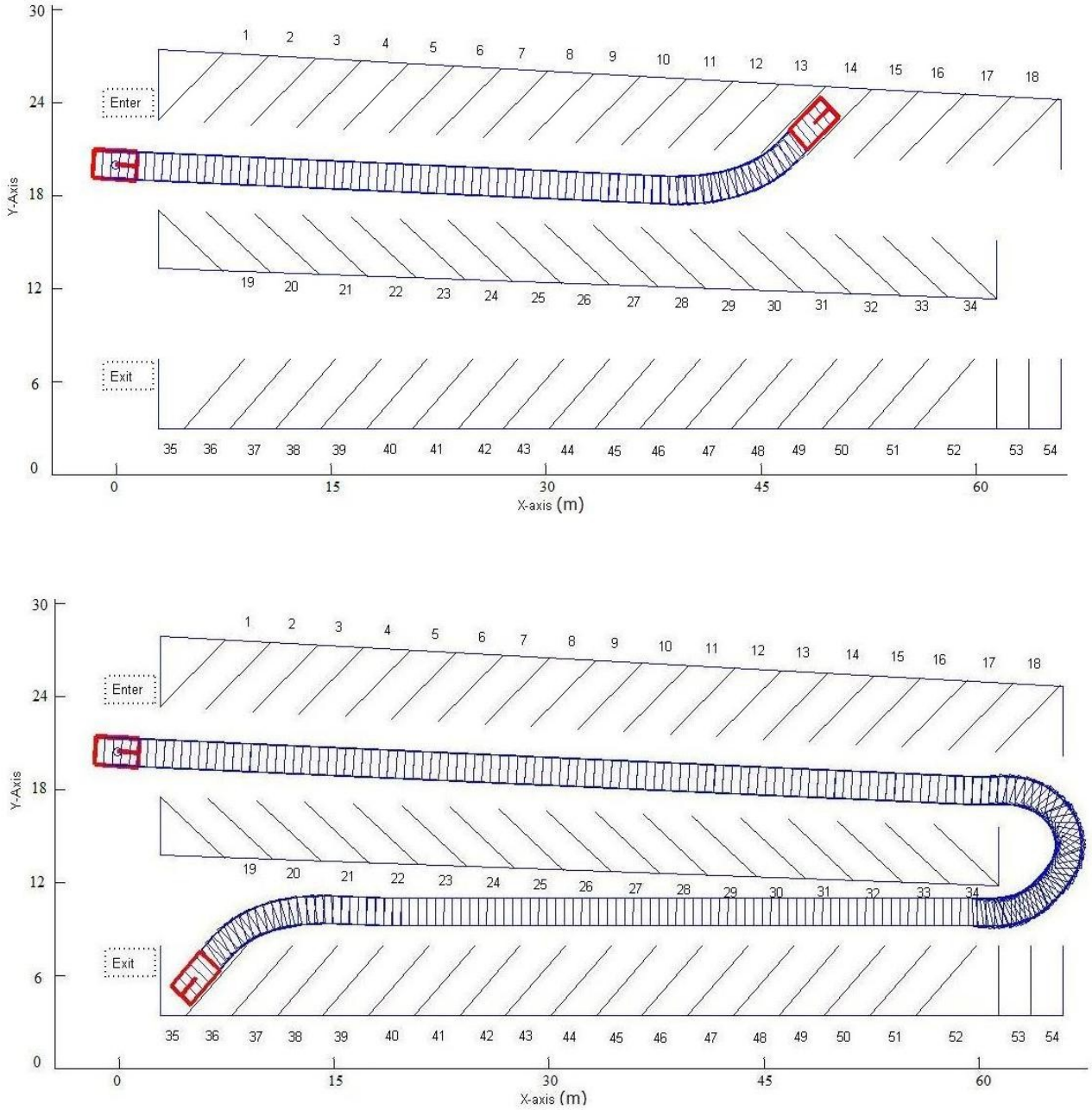


Figure 16: Results ASG Faculty Car Parking EME College

4.2 Parallel Parking

Parallel parking is the special parking type in which the vehicle has to be parked in a direction parallel to its heading at the beginning of parking process.

Since the parking space is limited, and final orientation needs to be parallel to the parking space, this task becomes complex and difficult. That is why, the parallel parking methods use a forward – backward motion to complete this task, i.e. the vehicle moves in both directions in different steps to complete the task.

In the method our application, we have proposed a 3 step motion, each governed by the Modified TSG, to complete the parking maneuver. In first 2 steps, the vehicle enters the parking space at specific orientation, and in third step, the vehicle reverses to reach the center of parking space and to ensure that its orientation is parallel to the parking space boundary.

Since the Modified TSG takes input of Vehicle initial position and orientation, a Target point and desired angle of approach, we need to compute these for parallel parking. It means, that for all 3 steps mentioned above we need to find, the target point, and we need to compute the desired angle of approach i.e. λ_f.

The parking space is again represented by a rectangle, which is defined by four points H, I, J and K. Opening of parking space is specified by points H and K. Based on this knowledge, points A, B, C, G, E and F are computed as shown in Fig. 17 . These points are computed by the relations given below.

$$G = \frac{1}{2}(H + I) \tag{4.12}$$

$$B = G + \left(\frac{L_V}{1.5} + \varepsilon\right) \cdot I\hat{J} \tag{4.13}$$

where $I\hat{J}$ is the unit vector from point I to J, L_V is the length of vehicle and ε is the safety distance

$$A = G + \left[\|IJ\| - \left(\frac{L_V}{1.5} + \varepsilon\right)\right] \cdot I\hat{J} \tag{4.14}$$

$$C = \frac{1}{4}(H + I + J + K) \tag{4.15}$$

$$E = K - W_V \cdot K\hat{J} - \varepsilon \cdot I\hat{J} \tag{4.16}$$

where $K\hat{J}$ is the unit vector from K to J and W_V is the width of the vehicle.

$$F = H - W_V \cdot K\hat{J} + \varepsilon \cdot I\hat{J} \tag{4.17}$$

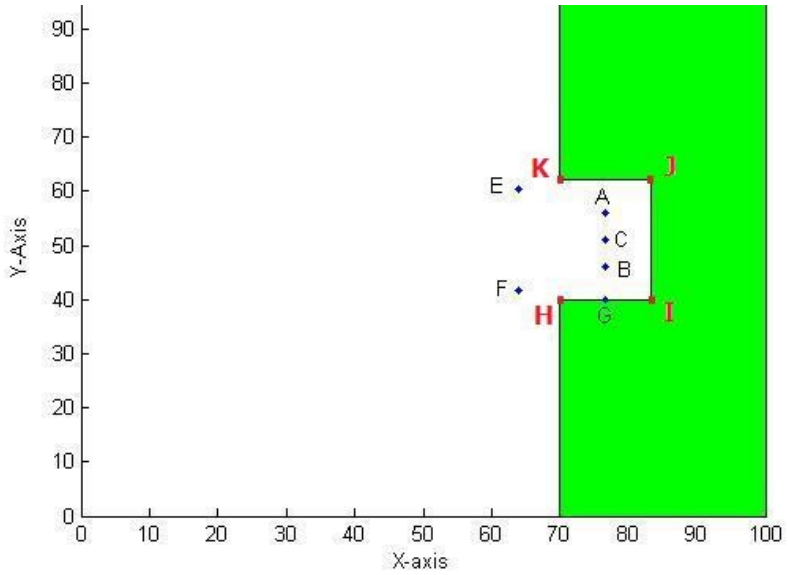


Figure 17: Using MTSG for Parallel Parking

If the parking space is known, and we know that vehicle has to perform parallel parking, it means essentially that the initial heading of the vehicle will be parallel to the parking space. This leaves us with 2 possibilities of parking maneuver. One when the vehicle starts from above the parking space, as shown in Fig. 19. Second case can be when the vehicle starts from below the parking space which is shown in Fig 20. The three steps required for parallel parking will be different for both the cases. So, different targets points and desired angles of approach will be needed for both cases. This problem has been addressed to ensure the robustness of the algorithm.

Conditions, target points and desired angles of approaches for both cases is discussed below.

Case 1: Upward Parking

Condition of this case is that distance between P (vehicle initial position) and F is less than distance between P and E. Parking process is:

Step1 – The vehicle goes to F with λ_f given by

$$\lambda_f = \tan^{-1}(FA_Y / FA_X) \quad (4.18)$$

Step2 – The vehicle goes to A with λ_f given by

$$\lambda_f = \tan^{-1}(IJ_Y / IJ_X) \quad (4.19)$$

Step3 – The vehicle reverses to C with λ_f given by

$$\lambda_f = \tan^{-1}(JI_Y / JI_X) \quad (4.20)$$

where FA is vector from F to A, IJ is vector from I to J, JI is vector from J to I.

Case 2: Downward Parking

Condition of this case is that distance between P and E is less than distance between P and F. Parking process is:

Step1 – The vehicle goes to E with λ_f given by

$$\lambda_f = \tan^{-1}(EB_Y / EB_X) \quad (4.21)$$

Step2 – The vehicle goes to B with λ_f given by

$$\lambda_f = \tan^{-1}(JI_Y / JI_X) \quad (4.22)$$

Step3 – The vehicle reverses to C with λ_f given by

$$\lambda_f = \tan^{-1}(IJ_Y / IJ_X) \quad (4.23)$$

4.2.1 Results

A large number of simulations were conducted in MATLAB to test the effectiveness of this algorithm. Different combination of vehicle initial position and parking space were tested to check the robustness.

A sequential example of parallel parking trajectory generated by the modified TSG is shown in Fig. 18 below.

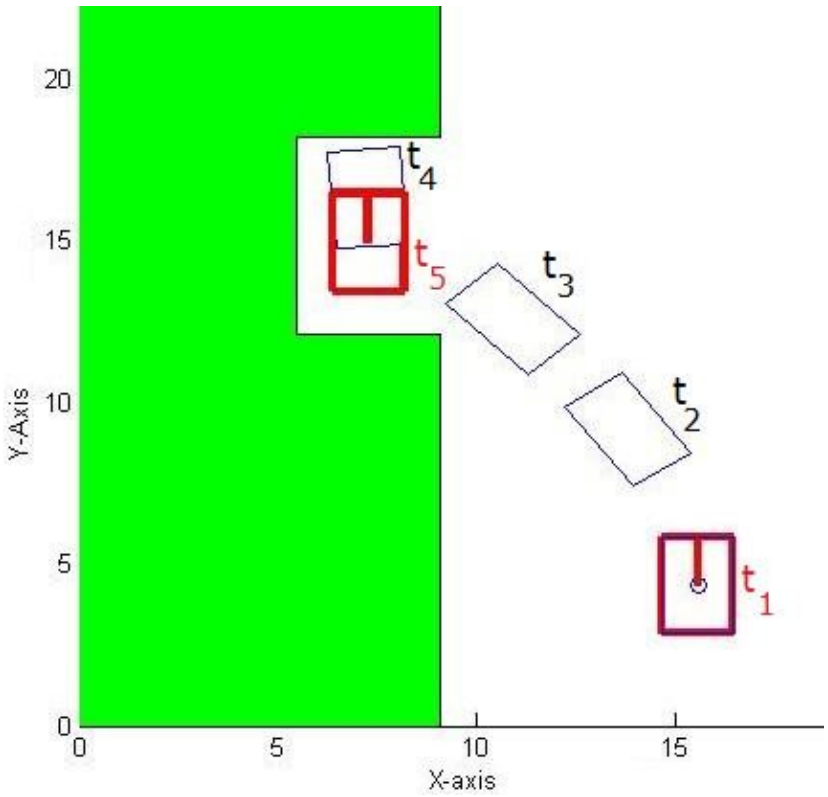


Figure 18: A Sequential Example of Parallel Parking

Instant t_1 shows the initial starting position of the vehicle. We see that initially the vehicle is parallel to the direction of parking slot. At t_2 and t_3 the vehicle is moving towards the parking space, which is the first step of parallel parking. At t_4 it moves inside the parking space, as governed by step 2. And then finally at t_5, the vehicle reverses to the center of the parking space.

A small subset of results of parallel parking is shown in Fig. 19 and Fig. 20 showing trajectories for parallel parking generated by the algorithm for different starting positions.

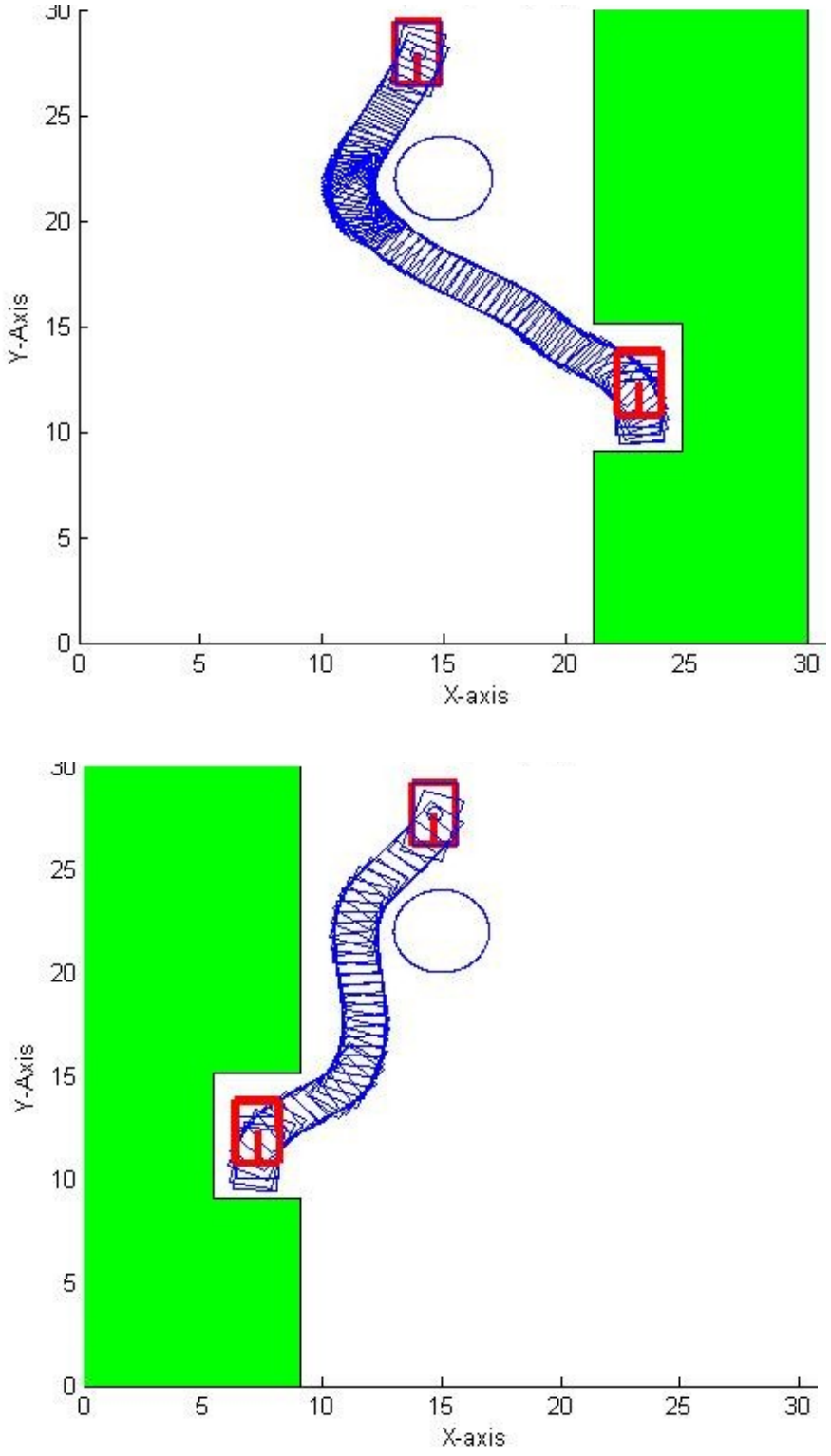


Figure 19: Modified TSG Results of Parallel Parking

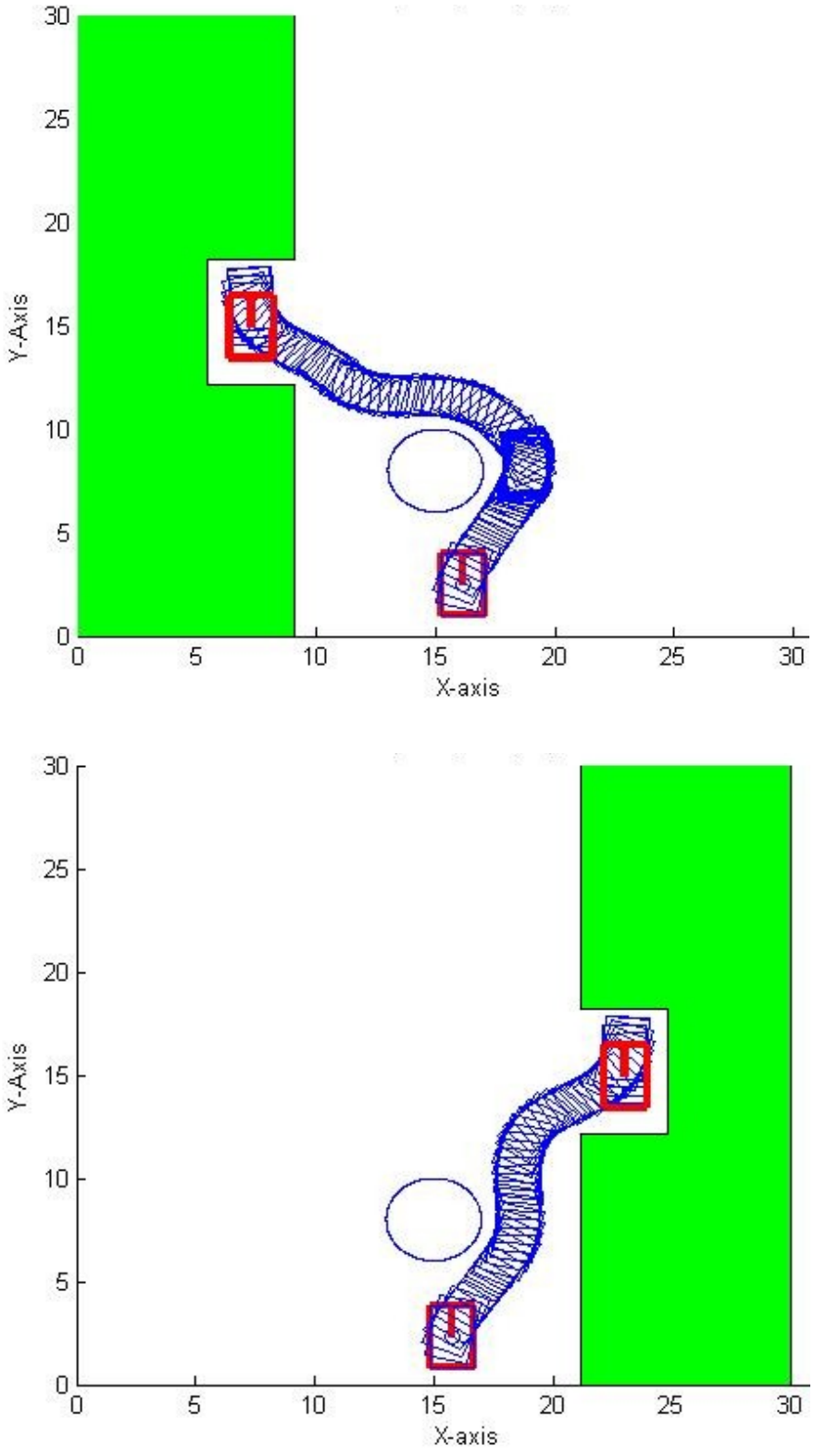


Figure 20: Modified TSG Results of Parallel Parking

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Comparison of Results

The main advantage of the proposed autonomous parking method over the existing ones is the simplicity of architecture and very less computational expense. This advantage is because that method used an already developed guidance law which has been optimized for best solution to hit the target here. So here, we are not only reducing the complexity and computational expense, rather we also have the optimal solution.

However, the robustness of this algorithm also ensures it performs well in all scenarios. A comparison of results of this algorithm with a couple of existing algorithm is presented below.

First we compare results of the proposed algorithm with the results presented by T. Hsu *et al* in [27]. The parallel parking solution it provides is shown in fig. 21:

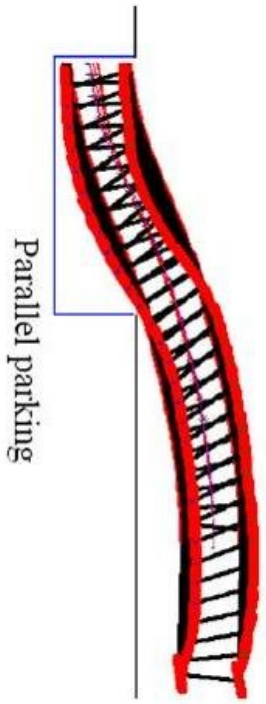


Figure 21: Parallel Parking Result of [27]

We can see from Fig. 21 that the method doesn't use the forward – backward motion and hence requires the larger parking space. While the method proposed in this thesis uses a forward – backward motion and hence ensures a more efficient parking maneuver in a confined parking space. The parallel parking result of our proposed method is shown in Fig. 22.

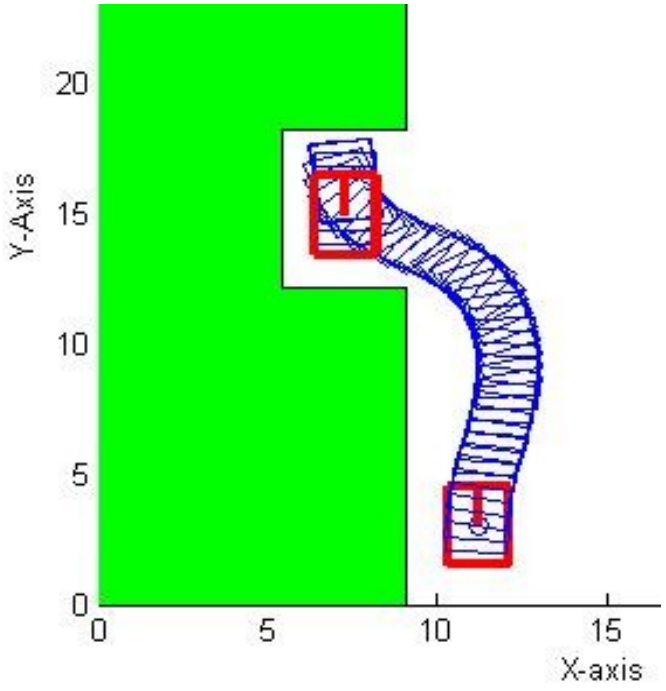


Figure 22: Result Parallel Parking using Modified TSG

Now, we compare the computational expense of the method presented in [27] and our proposed method. The method in [27] uses a two-step approach; path planning and path tracking. In first step, it plans the path i.e. it computes path comprising of circles, from starting position to end position. In path planning, it continuously makes circle passing through both current position and goal position. These geometric computations, made at every iteration, make the algorithm quite expensive computationally. In contrast to this, the method proposed in this thesis doesn't require continuous computations of arcs. It only uses an analytical solution which computes the trajectory based on only one parameter i.e. the lateral acceleration and it gives the desired trajectory, making it computationally inexpensive.

Now we compare the results of the proposed method with the results presented by Imae *et al* in [18]. Optimization has been considered in [18] and they

present the result of parallel parking maneuver. The parallel parking simulation result is presented in [18] is shown in Fig. 23.

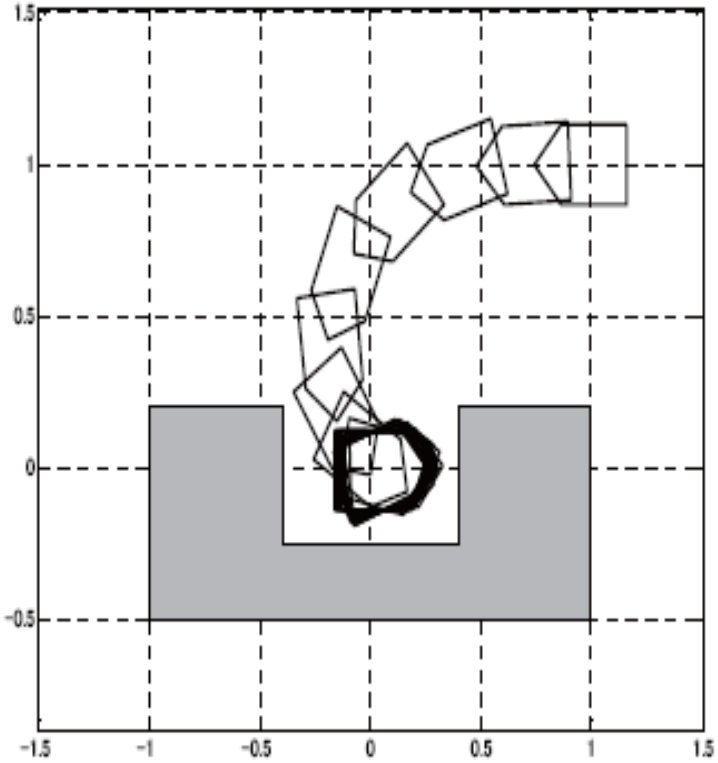


Figure 23: Parallel Parking Result Presented in [18]

We can see from Fig. 23 that the result of optimization results into a maneuver such that the distance is increased and the vehicle parks in the direction opposite to its original direction. However, if we use forward – backward motion, we can reduce the distance and park the vehicle in the direction parallel to its initial heading. This is elaborated in Fig. 24 below which shows result of the proposed method. We can see that the vehicle doesn't have to turn the whole way back to complete the parallel parking maneuver; rather it uses the motion in backward direction to gain optimum result. Also, an important point to note down here is that the proposed method uses the Guidance Law named as Trajectory Shaping Guidance, which was developed considering the optimal result. So the result in Fig. 24 is an optimum result ensuring no extra distance and turn.

We can see, from Fig. 24, that the vehicle doesn't take the long and suboptimal path, as it does in case of [18]. Also, the use of forward – backward

motion ensures parallel parking with vehicle parked in same heading as it was before the parking maneuver.

Now we compare the computational requirements of our proposed algorithm with those of the method presented in [18]. Optimality is achieved in [18] by formulating the problem as optimal control problem and then solving it. The authors state that “parallel parking problem with such dynamics is extremely difficult to solve numerically/analytically”, so they opted for the optimal control theory. They formulate a 6-element state vector and it iteratively solved to achieve the optimality. This leads to much more computations, as opposed to those required by our algorithm being analytical. TSG, used by our proposed algorithm, is an analytical law, that governs the trajectory shape y considering only one parameter and it has been derived considering the optimality of the trajectory. So, the Modified TSG needs less computation, yet it is based on optimal trajectory computational method.

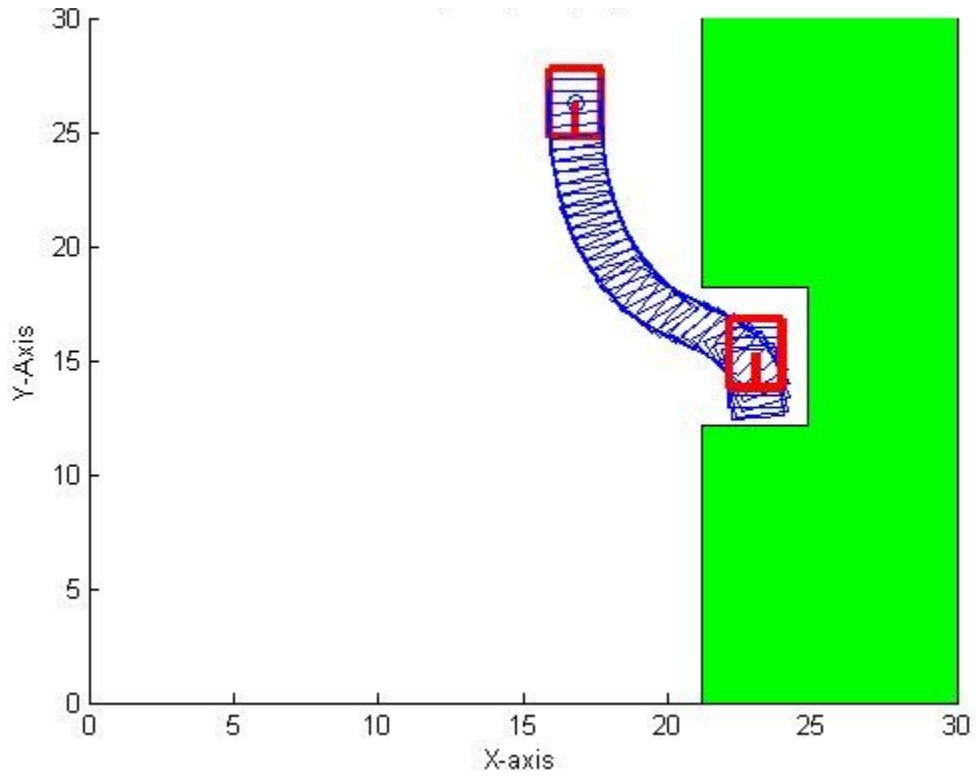


Figure 24: Parallel Parking Result using the Modified TSG

5.2 Conclusions

In this thesis a novel method, the Modified Trajectory Shaping Guidance, for trajectory planning of nonholonomic vehicles is presented. This method generates the trajectory for cases which require a desired angle or direction of reaching the target. This is suitable for car-like nonholonomic vehicles which have to be parked or docked or for any other similar application in which we have a desired final orientation.

The proposed method, the Modified TSG, is a trajectory planning method for nonholonomic vehicles based on guidance law Trajectory Shaping Guidance which is an analytic method and hence requires very less computations. Similarity of basic principle of missiles and nonholonomic vehicles has been used here to develop a fast method for nonholonomic vehicles. This is unlike the usual trajectory planning methods for nonholonomic vehicles which have complex system and require too much computation.

The Modified TSG has been developed for the task of autonomous parking of both types; Diagonal Parking and Parallel Parking. We see that without very complex design, system model, or complex computations, the method is able to generate the complete trajectory which satisfies the nonholonomic constraints. Also, the obstacle avoidance, considered the factor for much higher complexity of nonholonomic motion planning, has been successfully used without any drastic increase in complexity of system or computational requirements.

Detailed simulations results show the effectiveness of this algorithm. Different scenarios are presented in results of the algorithm to show the robustness. Diagonal Parking algorithm has been successfully applied to the ASG Faculty Car Parking Area, of College of Electrical and Mechanical Engineering, NUST and results are presented. This also shows that the proposed method can be easily applied for any environment.

5.3 Recommendations and Future Work

The proposed algorithm, the Modified TSG, has been developed which is a trajectory planner for nonholonomic vehicles also capable of obstacle avoidance. The method is fast, robust and simple to apply on any scenario. It has been implemented for the tasks of autonomous Diagonal and Parallel Parking.

But the applications of this algorithm are not limited to these areas. This algorithm can be very easily applied to any other application requiring path planning for nonholonomic vehicles. The ease and simplicity with which the algorithm was applied on autonomous parking method demonstrate the ease of adaptability and applicability of this algorithm.

Few of the possible applications of the Modified TSG are briefly discussed her:

1. The method can be applied for docking of autonomous mobile robots, which is needed for several applications. In docking, there is requirement to reach the desired place keeping a certain angle. For simple docking, the autonomous parking algorithm can be used directly. But a more complex docking system will need slight modifications and adaptations according to the requirements of the system.

2. This method can be used for motion planning and management of Autonomous Guided Vehicles (AGVs) in industries. The AGVs that are subject to nonholonomic vehicles and have to reach several areas and then need to be docked can highly benefit from this method. A complete motion management method that controls more than 1 AGVs can be built using this algorithm.

3. It can also be used for a multi-robot or a multi-vehicle setup having nonholonomic vehicles. Such vehicles or robots that start from different random starting position have to come into a formation to perform tasks in a group. This method can be used by vehicles to reach to certain points keeping a desired

angle so that they can come into a formation. So a multi-vehicle setup can use this method.

4. Since this is a general guidance law, it can be applied for Unmanned Ground Vehicles very easily because that is also working in 2D and might need to reach different points from a desired direction. Although most UGVs are not subject to nonholonomic constraints of car-like vehicles used here. But still they can benefit from this method as it provides a trajectory planning method which changes direction while moving. Nonholonomic constraints used in Modified TSG will need to be revised for this application.

5. This work can be applied for Unmanned Aerial Vehicles (UAVs) and Unmanned Underwater Vehicles (UUVs) as well. The difference here is that these vehicles work in 3d as opposed to vehicles and mobile robots. But if the target is known, the 2D problem can be formulated by considering an instantaneous plane which has both vehicle and target. As the vehicle and/or the target moves, this plane keeps changing, but every given instant, we can have such plane. In plane, the problem becomes 2D and the Modified TSG can be applied. Missiles use this guidance law in a similar fashion. However, the constraints on motion will need to be modified according to the actual constraints on that vehicle.

References

[1] N.A. Shneydor, Missile Guidance and Pursuit, 1st ed., Horwood Publishing, Chichester, England, 1998.

[2] F. Kunwar, F. Wong, R. Mrad, and B. Benhabib, “Guidance Based Online Robot Motion Planning for the Interception of Mobile Targets in Dynamic Environments”, Journal of Intelligent and Robotic Systems, 2006

[3] U. Ghumman, F. Kunwar, and B. Benhabib, “Guidance-Based On-Line Motion Planning for Autonomous Highway Overtaking,” International Journal on Smart Sensing and Intelligent Systems, 2008

[4] F. Kunwar, F. Wong, R. Ben Mrad, and B. Benhabib, “Time-Optimal Rendezvous with Moving Objects in Dynamic Cluttered Environments Using a Guidance Based Technique.” IEEE International Conference on Intelligent Robots and Systems, pp. 283-288, Edmonton, Canada, August 2005

[5] F. Kunwar and B. Benhabib, “Motion Planning for Autonomous Rendezvous with Vehicle Convoys” IEEE International Conference on Intelligent Transportation Systems, Toronto, Canada, 2006

[6] L.E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents.” American Journal of Mathematics, Vol. 79, pp. 497-516, 1957.

[7] J. A. Reeds and L. A. Shepp, “Optimal paths for a car that goes both forwards and backwards.” Pacific Journal of Mathematics, Vol. 45, 1990.

[8] K. Lee, D. Kim, W. Chung, H.W. Chang and P. Yoon, “Car parking control using a trajectory tracking controller”. SICE-ICASE International Joint Conference, 2006.

[9] D. Kim, and W. Chung, “Motion planning for car-parking using the slice projection technique”. International Conference on Intelligent Robots and Systems, 2008.

[10] M. B. Oetiker, G. P. Baker and L. Guzzella, “A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant.” IEEE Transactions on Vehicular Technology, Vol. 58, No.3, 2009.

[11] K. Kondak and G. Hommel, “Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms.” International Conference on Robotics and Automation, 2001.

[12] B. M¨uller, J. Deutscher and S. Grodde, “Trajectory generation and feedforward control for parking a car.” International Conference on Control Applications, 2006.

[13] M. Hentschel, D. Lecking and B. Wagner, “Deterministic path planning and navigation for an autonomous fork lift truck.” Symposium on Intelligent Autonomous Vehicles, 2007.

[14] J. Zang, D. J. Xuan, J. W. Kim and Y. B. Kim, “Development of autonomous balanced parking control system which used algorithm.” International Conference on Control, Automation and Systems, 2007.

[15] T. Inouel, M. Q. Dao and K. Z. Liu, “Development of an auto-parking system with physical limitations.” SICE Annual Conference, 2004.

[16] K. Z. Liu, M. Q. Dao and T. Inoue, “Theory and experiments on automatic parking systems”. Internatioal Conference on Control, Automation, Robotics and Vision, 2004.

[17] I. E. Paromtchik and C. Laugier, “Motion generation and control for parking an autonomous vehicle.” International Conference on Robotics and Automation, 1996.

[18] J. Imae, K. Yoshimura, G. Zhai and T. Kobayashi, “Real-time optimization for parallel-parking control of four-wheeled vehicles.” SICE Annual Conference, 2008.

[19] Z. Joung, X. DongJi, K. J. Wan, K and Y. Bae, “A study of autonomous parking for a 4-wheel driven mobile robot.” 26th Chinese Control Conference, 2007.

[20] Chen-Kui Lee, C. L. Lin and B. M. Shiu, “Autonomous vehicle parking using artificial intelligent approach.” The 4th International Conference on Autonomous Robots and Agents, 2009.

[21] T. H. S. Li, Y. C. Yeh, J. D. Wu, M. Y. Hsiao and C. Y. Chen, “Multi-functional intelligent autonomous parking controllers for car-like mobile robots.” Proceedings of TIA, 2008.

[22] C. H. Chao, C. H. Ho, S. H. Lin and T.H.S. Li, “Omni-directional vision-based parallel-parking control design for car-like mobile robot.” International Conference on Mechatronics, 2005.

[23] Y. W. Ryu, S. Y. Oh and S. Y. Kim, “Robust automatic parking without odometry using enhanced fuzzy logic Controller.” International Conference on Fuzzy Systems, 2006.

[24] P. Zarchan, Tactical and Strategic Missile Guidance, 3rd ed., AIAA Progress in astraunatics and aeronautics, Reston, VA, 2002.

[25] J.A. Lukacs and O. A. Yakimenko, “Trajectory-shape varying missile guidance for interception of ballistic missiles during the boost phase.” AIAA Guidance, Navigation and Control Conference and Exhibit, 2007.

[26] P. Fiorini and Z. Shiller, “Motion Planning in Dynamic Environments using Velocity Obstacles”, International Journal of Robotics Research, Vol 17, No. 7, pp 760-772, 1998.

[27] T. Hsu, J. Liu, P. Yu, W. Lee and J. Hsu, “Development of an automatic parking system for vehicle”, IEEE Vehicle Power and Propulsion Conference, China, 2008.