Faculty of Engineering and Physical Sciences

Department of Mechanical Engineering Sciences

University of Surrey

Automatic Reverse Parking of an Articulated Vehicle using Fuzzy Logic

A dissertation submitted to the Department of Mechanical Engineering Sciences of the University of Surrey in partial fulfilment for the degree of MEng Mechanical Engineering.

Ramon Jose Ward Santiago

Under the Supervision of

Dr. Hartavi Karci

Personal Statement of Originality:

I confirm that the submitted work is my own work. No element has been previously submitted for assessment, or where it has, it has been correctly referenced. I have also clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook. I agree that the University may submit my work to means of checking this, such as the plagiarism detection service Turnitin® UK. I confirm that I understand that assessed work that has been shown to have been plagiarised will be penalised.

Ramon Santiago

Abstract

This thesis looks at the design of an automated parking system capable of autonomous reverse docking of Fords articulated vehicle. This vehicle is tasked with the loading and unloading of various goods, resulting in a vehicle that varies in mass. The autonomous parking of an articulated vehicle requires both a path planner and a control system to allow for stability in the reversing manoeuvres. Both the path planners and the controllers were designed using fuzzy logic. The fuzzy path planning systems have been designed for both forwards and backwards manoeuvres. The forwards manoeuvres are used to drive the vehicle to the desired reversing position, to allow for jack-knife prevention, and to realign the vehicle with the parking space if the reversing manoeuvre is unsafe. The design of the reversing system required two controllers, the first controller requests a trailer orientation angle that allows the vehicle to achieve a desirable final position and orientation. The second controller is used to prevent jack-knifing of the vehicle during this orientation adjustment. To model the vehicle a kinematic bicycle model was used alongside with a longitudinal inertia model to allow for variations in acceleration, when loaded and unloaded. The automated parking system shows successful simulated docking of all sensible initial conditions and the majority of imprudent initial conditions the driver might engage the system in.

Acknowledgements

Throughout this project I have had the pleasure of working with Dr Hartavi Karci. I am honoured to have had the pleasure of study under her, especially in such an interesting field of research. The guidance and support given has made this project very enjoyable and rewarding. I would also like to acknowledge Amr Mahmoud Soliman, Carlos Munoz, Alex Makedonas, and Amy Tribe for all their support throughout this project. Finally, I would like to thank Kathy Santiago for proof reading of major documentation.

Contents

Table of Abbreviations

Abbreviation	Full Name
T&T	Truck and Trailer
MPC	Model Predictive Controller
LQC	Linear Quadratic Controller
FLC	Fuzzy Logic Controller
NNC	Neural Network Controller
NMPC	Non-Linear Model Predictive Controller
TS	Takagi-Sugeno
LDE	Linear Differential Equation
HGV	Heavy Goods Vehicle
VUBFC	Variable Universe Based Fuzzy Controller
TMU	Trajectory Management Unit
PD	Proportional Differential
FFPP	Forwards Fuzzy Path Planner
FFC	Forwards Fuzzy Controller
RFPP	Reverse Fuzzy Path Planner
RFC1	Reverse Fuzzy Controller 1
RFC2	Reverse Fuzzy Controller 2
LHS	Left Hand Side
NL	Negative Large
NM	Negative Medium
NS	Negative Small
N	Negative

Z	Zero
P	Positive
PS	Positive Small
PSM	Positive Small Medium
PMS	Positive Medium Small
PM	Positive Medium
PML	Positive Medium Large
PL	Positive Large
PVL	Positive Very Large
S	Small
SL	Small Large
KPI	Key Performance Indicator

Table of Parameters and Variables

Parameter	Description	Value
A	The centre of the truck's front axle	N/A
B	The hitch point	N/A
C	The centre of the truck's rear axle	N/A
D	The centre of the trailer rear axle	N/A
x_c	X position at C from origin	(m)
y_c	Y position at C from origin	(m)
$\dot{x}_c$	X direction velocity at point C	(m/s)
$\dot{y}_c$	Y direction velocity at point C	(m/s)
x_D	Y position at D from origin	(m)
y_D	Y position at D from origin	(m)
$\dot{x}_D$	X direction velocity at point D	(m/s)
$\dot{y}_D$	Y direction velocity at point D	(m/s)
L_1	The length of the truck from A to C	3.6m
L_2	The distance of the hitch point from the truck's rear axle	0.51m
L_3	The length of the trailer from B to D	5.01m
θ_1	Orientation of truck to horizontal axis.	(^{o})
$\dot{\theta}_1$	Angular velocity of truck	(^{o}/s)
θ_2	Orientation of trailer to horizontal axis.	(^{o})
$\dot{\theta}_2$	Angular velocity of trailer	(^{o}/s)
γ	The hitch angle $(\theta_2 - \theta_1)$	(^{o})
$\dot{\gamma}$	Rate of change of hitch angle $(\dot{\theta}_2 - \dot{\theta}_1)$	(^{o}/s)
V	Longitudinal velocity of the vehicle from C	(m/s)
V_2	Velocity perpendicular to trailer	(m/s)
δ	Steering angle	(^{o})
F	Force	(N)
m	Mass	(Kg)

a	Acceleration	(ms^{-2})
$bv(t)$	Resistance due to current velocity	(N)
$G(s)$	Transfer function in frequency domain	(Hz)
$V(s)$	Velocity in frequency domain	(Hz)
$V_r(s)$	Reference Velocity in Frequency Domain	(Hz)
$F(s)$	Force in Frequency Domain	(Hz)
K	Gain	N/A
τ	Time Constant	(s)
$\mu_A(x)$	Membership Function for Universe (x)	N/A
$\mu_B(x)$	A Different Membership Function for Universe (x)	N/A
$\mu_c(z)$	Output Membership Function for a Fuzzy Universe (z)	N/A
U	Fuzzy Universe	N/A
x	An element in the universe U	N/A
A	A Fuzzy Set	N/A
B	A Different Fuzzy Set	N/A
z	Crisp Output	N/A
$U(s)$	Reference Velocity in Frequency Domain	(Hz)
$Y(s)$	Velocity Response in Frequency Domain	(Hz)
$y(t)$	Velocity in time domain	(s)
k	Gain constant in time domain	N/A
$x(t)$	Distance in time domain	(ms^{-1})
c	Constant of integration	(m)
t	Time	(s)
ϵ	Overall Error	N/A
ϵ_p	Final Position Error	(m)
ϵ_θ	Final Orientation Error	(°)
x_{De}	X Position Error at Point D	(m)
y_{De}	Y Position Error at Point D	(m)
θ_{1e}	Truck Orientation Error	(°)
θ_{2e}	Trailer Orientation Error	(°)

1. Introduction

The aim of this project is to design and simulate automated parking of Ford's Truck and Trailer (T&T) vehicle inside a parking space. The vehicle is a heavy goods vehicle (HGV) consisting of a truck and trailer joined together at a hitch point. The HGV is tasked with delivering goods to a docking yard, these goods need to be unloaded and loaded. For this reason, the vehicle must be able to reverse park into the selected parking space, seen in Figure 1 the actual space can be found in Appendix 1.1. The HGV operator will drive the vehicle into the docking bay entrance, stop the vehicle and then engage the automated parking system. The operator should be able to do this from a wide range of initial positions inside the bay entrance. The vehicle should always be able to complete the require manoeuvres with the same or better accuracy than the operator in the same or less time.

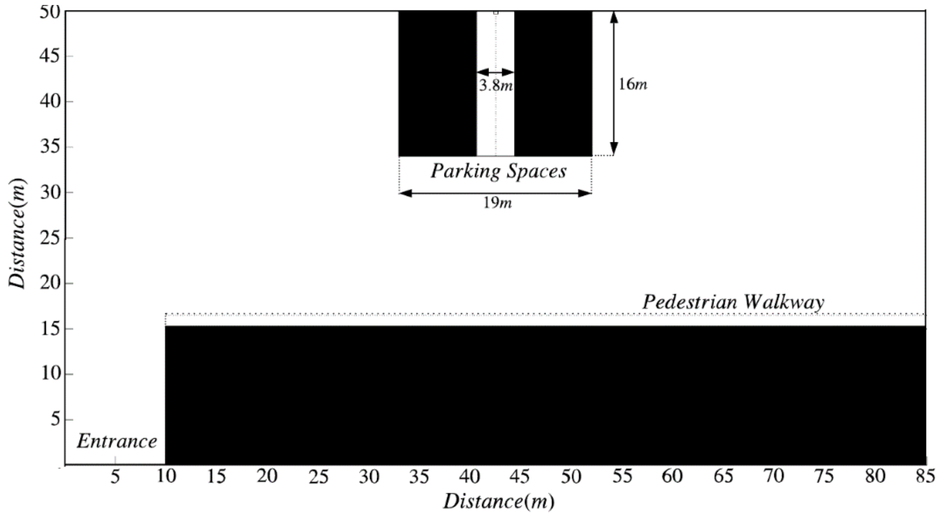


Figure 1, the parking space geometry provided by Ford.

The desired parking space in Figure 1 is the central parking space (in white). Inside this parking space is a small imaginary box at the back of the space. The very rear of the trailer must stop inside this box for it to be considered better than a human driver. The box is $40cm$ wide and $20cm$ long. The vehicle must also have a trailer orientation error within $\pm 2°$, and a truck orientation error within $\pm 5°$ at this final position. To get to this final position the vehicle should not undergo more than 10 manoeuvres and should not take longer than 2 minutes to complete the docking manoeuvre. A path planning system and control system will be designed that can develop and follow a trajectory that ensures the vehicle doesn't collide with any boundaries, goes over the pedestrian walkway or the other parking spaces. The control system should follow this trajectory to achieve the desired final position and orientation. The automated parking system should be able to solve in real time using a time step that shows convergent behaviour. The key performance indicators of this project are summarised in Table 1.

Table 1, the KPIs of this project,

1). Error in Final Trailer Angle	2). Error in final truck angle
3). Error in final trailer position	4). Error in final hitch angle
5). Velocity and Acceleration of vehicle	6). Time required to park
7). Number of manoeuvres	8). Computational time

2. Literature Review

2.1. Motivation:

In recent years, the development of truck and trailer (T&T) trajectory controllers has been in high demand for transportation of manufactured parts from the factory to storage or dispatch, in particular the docking, loading, and unloading of products. This is because the reverse docking, and other reversing manoeuvres, require highly skilled operators. An average drivers skill would lead to large errors in position, trailer orientation and truck orientation, a great deal of costly training and practice is required to develop the required skills (Eatherley and Petriu 1995 p. 810).

The backwards trajectory control of articulated vehicles is inherently a nonlinear control problem. The difficulty in the design of a suitable control system is not only caused by the nonlinear dynamics, but also by its nonholonomic constrains (T. Mason 2007 p 7-8) as well as the physical limitations of the vehicle. One of the more difficult limitations to overcome is the jack-knife phenomenon, once the T&T reaches a critical angle between the truck and trailer the vehicle cannot be recovered with any steering input, it must drive forwards in order to overcome this loss of control.

The aim of this project is to design a controller that can automatically park Fords truck and trailer in the parking space provided. The vehicle must first move forwards to reach the desired parking space, once there it should reverse park into the bay with minimal errors in truck and trailer orientation as well as x and y positions of the truck and trailer.

2.2. Review of Control Strategies:

Currently in the literature, there has been many control strategies implemented to allow for the trajectory control of a T&T vehicle. Some of the more notable strategies have been; a Model Predictive Controller (MPC), a Linear Quadratic Controller (LQC), a Fuzzy Logic Controller (FLC) and a Neural Network Controller (NNC).

(Backman, Oksanen and Visala, 2012) looks at the lateral control of an agricultural tractor and trailer using a Non-Linear Model Predictive Controller (NMPC), the control strategy designed made use of 9 state variables with an Extended Kalman Filter to allow for accurate estimation of these states. The results show that the vehicle was able to keep within a lateral error of 10cm at a 12km/h, which is similar to the error of a human operator. This paper only addresses the forward movement of an articulated vehicle in uncertain ground conditions. The novelty of this paper comes from the use of a fully automated system capable of trajectory planning and lateral control, the paper also looks at using a vehicle slip angle model to help deal with the uncertain ground conditions. However, the vehicle control system was not designed to complete reversing manoeuvres nor was it capable of switching between them.

In (Pradalier and Usher, 2008) the authors use a Model Predictive Controller, forwards and backwards manoeuvres were used to help avoid potential jack-knife situations that propagated due to the trajectory oscillating between the two sides of the desired path, if this occurred at the wrong frequency the vehicle would jack-knife. In their experiments the vehicle was required to move forwards, the use of a hitch angle servo helped to keep the vehicle stable. The Ford T&T vehicle does not include an actuator at the hitch point, this either requires the vehicle to self-stabilise using the steering input or drive forwards when a potential jack-knifing situation occurs.

Another commonly selected control method found in the literature is the use of an LQC. In (Divelbiss and Wen, 1997) the use of feedforward PID into an LQC was used to track a nonholonomic reference path for a car and trailer. The use of this system showed promising results with small lateral error when following the reference path for both forwards and reversing manoeuvres. However, the controller showed large errors in the steering angle input compared to the desired steering angle. The path generated for the car and trailer shows the vehicle driving forwards past the parking space before engaging in a reversing manoeuvre. The system designed shows capable results, nevertheless, the need for a path planning system that feeds into a PID for trajectory generation feeding into a LQC seems like a complicated approach.

An alternative to this LQC method is proposed in (Hoel and Falcone, 2013). This paper looks at the design of a controller system capable of automatically turning a T&T vehicle with a Nordic combination by 90 degrees in a tight environment that requires a forwards and backwards manoeuvre. The control strategy does this by allowing for two controller inputs, the steering angle and the velocity. The controller uses a ghost approximation of the vehicle, which is simpler to the real vehicle, having the same steady state behaviour but slightly dissimilar transient behaviour. The approach uses a path planning algorithm to generate the trajectory, this is then fed into a ghost approximation of the vehicle, which is linearized and fed into an LQC. The method shows good simulation results but no testing on the real vehicle was done.

Currently in the literature, there has been little mention of a fuzzy control systems undergoing both forwards and reversing manoeuvres in a single simulation to achieve the desired docking position, while avoiding jack-knifing. A fuzzy control systems behaviour is more representative of an expert actions, and therefore linguistic rules can be applied to achieve an improved result. In (Eatherley and Petriu 1995) a fuzzy controller was used to control both forward and reversing manoeuvres for the docking of a T&T vehicle, however, the forward and backwards velocities were limited to 3 constants. This use of varied velocities is realistic, however, having the velocity membership functions as singletons would result in an inherently non-fuzzy output with unrealistic vehicle accelerations. In (Leng and Minor, 2017) a trajectory controller for a car and trailer is proposed that uses both forward and backwards manoeuvres. The paper has various merits; the vehicle has realistic velocities, accelerations and rate of steering input, however, the modelling method does not include any variations in mass, this could result in large errors if the vehicle was fully loaded. Along with this, the paper only addresses forwards or backwards manoeuvres.

Within the literature there are many examples of papers that look to solve the problem of docking a T&T vehicle with the use of a single fuzzy controller or fuzzy-neural controllers (Eatherley and Petriu 1995), (Kong and Kosko, 1992), (Yang, Yuan and Yu, 2006), (Siamanta and Manesis, 2009), (Tanaka, Taniguchi and Wang, 1997), (Tanaka and Sano, 1994), and (ichihash and Tokunag, 1993). These papers can be further divided into two main categories; papers that use a Mamdani controller structure and those that use a Takagi-Sugeno (TS) structure. The papers that use the Mamdani structure with a single fuzzy controller inevitably require a large number of rules, with the minimum found being 105 (Yang, Yuan and Yu, 2006) this increases the computational time substantially. In (Yang, Yuan and Yu, 2006) the author combats this problem by introducing the use of variable universe based fuzzy controller (VUBFC), this reduces the total number of rules resulting in less computational loading. However, the number of rules is still relatively high, and requires a sample time of 0.1s to solve in real time.

The following papers use a multi-fuzzy controller design strategy to solve the docking problem of a T&T vehicle; (Zimic and Mraz, 2006), (Kodituwakku, 2011), and (Riid, Lei and Rüstern, 2009). (Zimic and Mraz, 2006), looks at reducing the large number of fuzzy rules a single controller needs to produce one control output. This is achieved by using a decomposition method that reduces the number of inputs and consequently the number of rules for each controller. The control structure proposed is logical and works well for the reversing manoeuvre as the steering output is based on the hitch angle error, preventing jack-knifing. However, the same control strategy would not make sense for the forward manoeuvre as the hitch angle is already self-stabilised. This paper gives a keen insight into the possible controller decomposition methods that can be used to solve complex T&T reversing manoeuvres. In (Kodituwakku, 2011), the authors compared the usage of fuzzy logic and neural network controllers for T&T reverse docking. This paper used a much simpler decomposition method than (Zimic and Mraz, 2006), with two fuzzy controllers using a TS structure. For these controllers a limited use of initial conditions was used to demonstrate the robustness of the controller strategy. The authors also didn't consider the potential danger of not controlling the truck orientation angle, this could result in the truck causing damage to adjacent vehicles in the docking area. Finally, in (Riid, Lei and Rüstern, 2009) another fuzzy controller strategy is used for the reversing of a truck and multiple trailers, this controller strategy uses a Mamdani structure. This paper adds novelty to the literature as the first fuzzy controller in the hierarchy is a trajectory management unit (TMU). The TMU acts as a simple path planner system using fuzzy rules to assign desired trailer angles to specific x, y positions of the vehicle in the parking universe. This essentially guides the vehicle to a specific location, this location can be changed and a desired trailer angle can be assigned to the final position. This method is not particularly robust, if the geometry or size of the parking universe is altered then the fuzzy input rules of the TMU need to be redesigned. The second stage of the control system uses two controllers to reduce the error in the hitch angles based on the desired trailer orientation or the previous bodies hitch angle. The last stage of the control strategy uses a PD controller to select a steering angle input to achieve the desired trailer angles and prevent jack-knifing. The use of a PD controller here is unrealistic as a steady state error of zero is advantageous.

2.3. Review of Plant Models:

In all the papers reviewed the authors have modelled their vehicle using a kinematic bicycle model. Models found in the literature are either multi-rigid kinematics (Cariou and Lenain, 2010) or multi-chained kinematics (coordinate transformation approach), these are then further divided into continuous or discreet methods.

The following paper (Zimic and Mraz, 2006) makes use of a discrete multi-chained kinematics model to mathematically model the T&T vehicle it uses. This vehicle consists of a single axle truck with a two-axle trailer, the vehicle has the hitch point behind the truck's rear axle. The author has decided to use a discrete method as most of the simulation work was done in Simulink, which can solve either discrete or continuous.

During research a total of 7 papers were found using a multi-chained continuous kinematics model, the most notable paper was (Altafini, Speranzon and Wahlberg, 2001). This paper modelled a T&T system with a double joint mechanism in-between the truck and trailer, the first hitching point is located behind the truck's axle and the second is on the trailer's front axle. The use of this kinematic method produced a relatively complex and non-linear model. However, the author proved using Lyapunov stability that the reversing of the vehicle is asymptotically stable. The authors then use a small scale model to demonstrate that their model and controller can accurately track the reference path.

14 papers were found using the multi-rigid kinematics method, 4 discrete and 10 continuous. In (Pradalier and Usher, 2008) the authors used this technique to model a T&T vehicle, the hitch point was located behind the rear axle of the truck and in front of the front axle of the trailer. The truck made use of two axles while the trailer used one. This paper ran both simulations and experimental tests, in both tests the strategy managed to follow the trajectory with acceptable errors, with the experimental tests showing similar results to the simulated results.

2.4. Conclusion:

The main drawback to using a Neural Network Controller is that this technique requires self-learning. The learning algorithm is based on the plant model provided, this algorithm then runs repeated simulations to learn and optimise the controller's actions to ensure accurate reference following. The problem with this method is the inaccuracy in the plant model to the real T&T behaviour. The controller would need to be run in the real vehicle until sufficient accuracy was achieved, this causes major safety issues, especially for a heavy goods vehicle. A NNC was used in (Kodituwakku, 2011) and compared to a FLC, this paper did not show a drastic improvement in the reduction in errors when using the NNC. For this reason, a NNC will not be used in this thesis.

The large majority of physical dynamic systems can't be represented using linear differential equations (LDE) due to their non-linear nature. However, linear control methods make use of linearization techniques that allow for control of these systems. These linear control methods often require small operational ranges, if they are large the system is prone to becoming unstable. Furthermore, for decidedly nonlinear systems formulating an accurate mathematic model can be very difficult, and often only input and output data is available to run the system. In these situations the use of Fuzzy Logic Control is superior as accuracy in the plant is not vital to its success (Mehran, 2008). For this reason, FLC will be used in this thesis.

Another reason not to use either LQC or MPC was due to the added complexity in using these controller strategies compared to FLC. The added computational loading required for the prediction horizons could result in controller lag and potential loss of control, this is particularly emphasized due to the non-linear dynamics and non-holonomic constraints in the vehicle's model and path. When reviewing the literature it was found that the papers for LQC, MPC and FLC all showed relatively similar results, so logically the simplest one has been chosen.

The use of TS fuzzy systems are computationally efficient, allowing for the use of single fuzzy controllers to compute complex real life systems. However, Mamdani systems are more intuitive and are well suited to human input systems such as the docking of a T&T vehicle, for this reason a TS systems will not be used **(**Uk.mathworks.com, 2012). The alternative is to use multiple fuzzy controllers, to reduce the total number of rules and complexity while also reducing the computational time. As microcontrollers are inexpensive this method will be used to solve the computational loading problem without unneeded complexity.

Another issue that hasn't been mentioned in the literature is the variation in the mass of the vehicle. The mass of the vehicle will vary depending on weight of cargo hauled. In reality, if the vehicle's weight is increased then the velocity response will change accordingly. This point is especially true for the trajectories that have multiple forwards and backwards manoeuvres as the acceleration and deceleration of the vehicle will vary with the load. In (Pradalier and Usher, 2008) the author makes use of a first order transfer function to accurately predict the real behaviour of the vehicle's velocity to a step input, this was validated experimentally. This method will be used to model the variation in velocity to a step input.

As the vehicle only needs to park in a particular docking area, a simplistic path planner will be developed using fuzzy rules similar to (Riid, Lei and Rüstern, 2009). The use of this planning algorithm reduces complexity in the generation of a reference trajectory, this planning algorithm allows the vehicle to park in the docking bay from a multitude of realistic initial positions and orientations.

The vehicle will be modelled using a multi-rigid kinematic model due to its simplicity, its common use in literature and non-tedious derivation method (Chang and Ma, 2009). As a FLC is being used this model should suffice and has been proven in multiple papers (all mentioned above) to produce accurate results in simulation and real world testing.

In summary: The trajectory planner and lateral Mamdani Fuzzy Controller structure will be adapted from (Riid, Lei and Rüstern, 2009), a multi-rigid body kinematic bicycle model will be used for modelling the system. With this model the velocity will be simulated using a first order transfer function, allowing for different responses to loaded and unloaded conditions. Jack-knifing will be prevented using a hitch angle error controller, and a precautionary system will drive the vehicle forwards if the hitch angle exceeds a critical value.

A Summary of all the papers reviewed for this project can be found in the Appendix 2.4.1, I have only mentioned the most relevant papers above.

3. Plant Model

3.1. Justification of Kinematic Model.

Kinematics are used to find expressions for the position, velocity, and acceleration in terms of the variables that represent the states of the system. As the aim of this project is to control the position and orientation of the vehicle, therefore, this approach seems appropriate. The use of a kinematic model to represent a large dynamic vehicle has been readily used in the literature, this is due to its surprising accuracy compared to the real vehicle. The use of a kinematic bicycle to model dynamic systems is a good approximation for low speed manoeuvres$(< 5m/s)$ (Rajamani, 2012, p.21) This is because it has been assumed that at low speed the lateral force generated by the tyres due to centripetal forces is small. From this assumption it is logical to assume that the velocity vector at each wheel is in the direction in which the wheel is driving. This is equivalent to assuming that there is zero slip angle between all the wheels.

Although the mass is not included in a kinematic model, the problem of having a loaded and unloaded vehicle still needs to be addressed. For this reason the longitudinal velocity of the vehicle will include an inertia model to allow for accurate representation of the vehicle's response to a change in mass. This model will only affect the longitudinal behaviour of the vehicle, the lateral characteristics will remain unchanged as we have previously assumed negligible lateral forces. In summary it has been assumed that the tyres have perfect traction with the ground (no slip), the ground is perfectly flat and that the lateral forces are negligible.

The kinematic bicycle model assumes that the two wheel on a common axle are the same, and can be approximated as one central wheel. These wheels are assumed to have no lateral or longitudinal slip, this model also only allows for a steering input from the front wheels. Restricting the model to 2D allows for this kinematic approach to ensure the nonholonomic conditions are not broken. The bicycle model does include Ackermann steering, however, the central wheel represents the average steering angle of the two wheels (Waslander and Kelly, 2017).

3.2. Derivation of Kinematic Model.

The derivation of the vehicle makes use of the multi-rigid-body method, the paper that most accurately represents the derivation is (Ren, 2013). However, the Ford vehicle has it's hitch point located in front of the rear most axle of the truck, in the literature no vehicle used this location to join the T&T. Model seen in Figure.

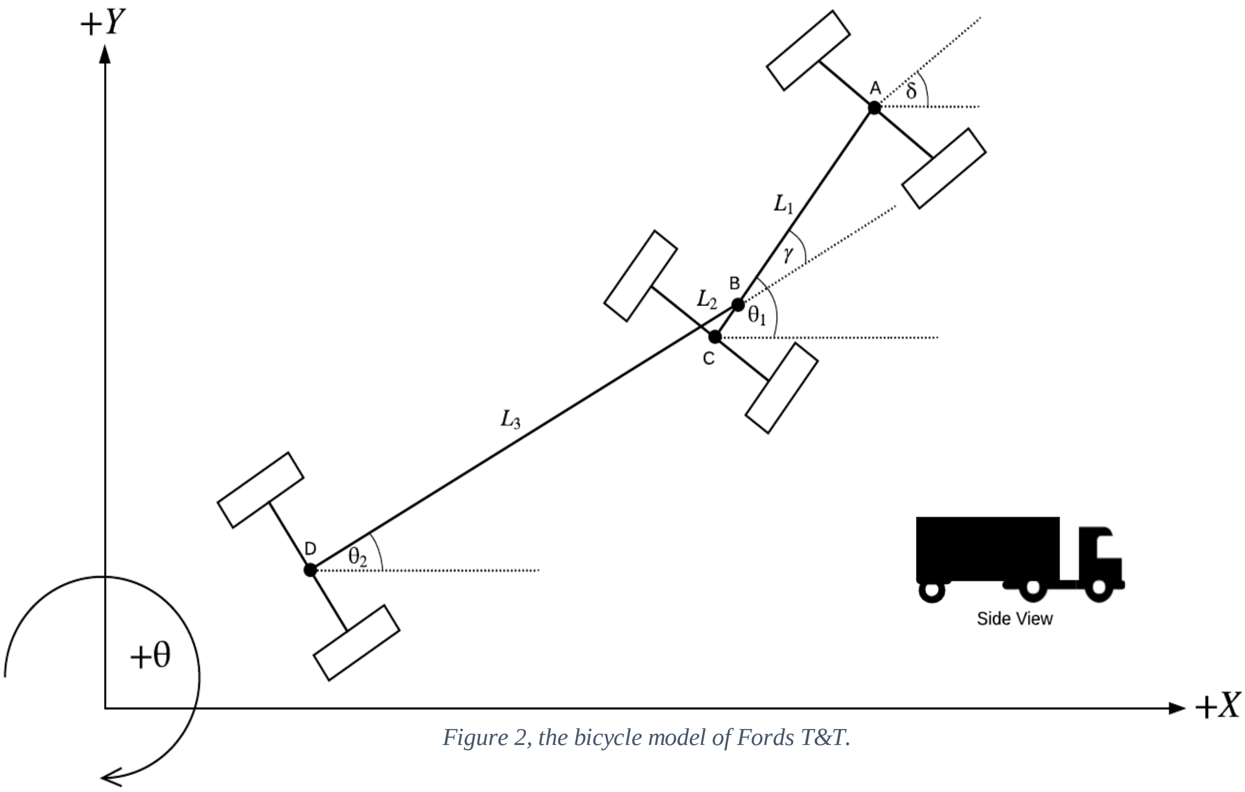


Figure 2, the bicycle model of Fords T&T.

The table of parameters and variable shows all of the parameters that are used in Figure 2 and the derivations below. The system uses a Cartesian coordinate system seen in Figure 2. The definitions of the vehicle's positive and negative directions and orientations are defined in Figure 3.

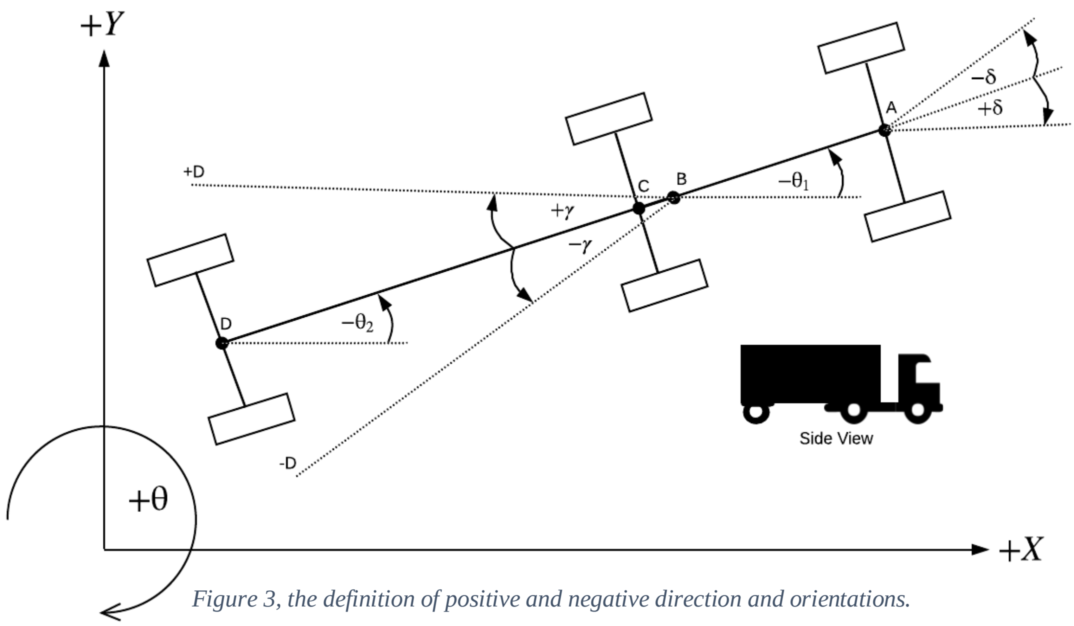


Figure 3, the definition of positive and negative direction and orientations.

Figure 3 shows that the steering angle from the front tyres is defined from the centre line of the truck, clockwise is positive, and anticlockwise is negative. The two dotted lines –D to B and +D to B show an imaginary trailer, these lines are used to help show the reader the direction of positive and negative hitch angle; clockwise about B is positive and anticlockwise is negative. The truck and trailer both show negative orientation due to their anticlockwise rotation from the horizontal axis. If the vehicle was in line with the horizontal, then the orientation would be 0 degrees.

To derive meaningful equations of position and velocity the minimum number of variables required to control the vehicle need to be known, i.e. the states of the system. In layman's terms the system can be described as being two rods joined together by a hinge. In order to control this system only one set of x, y coordinates need to be known, along with the angle of the truck and trailer. The vehicle is linked together, as long as the x, y position somewhere on the vehicles body are known, as well as the orientations, then the vehicle can be completely described. If the x, y position somewhere else on the vehicle is required, for example when reverse docking the rear trailer position is important, then a simple geometric equation can translate the coordinates to that point, see Figure 4.

The states chosen to describe the forwards control of the system were: $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{x_C}, \boldsymbol{y_C}$ and for backwards control $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{x_D}, \boldsymbol{y_D}$. An alternative to these states could make use of any x, y position in any location, however, this position has been chosen because the truck is rear-wheel drive and therefore the velocity vectors will be derived from this point making this position the simplest to model. In terms of control, the rear position of the trailer is advantageous to use as the distance from the vehicle to the dock can be more easily understood. The hitch angle could have also been used instead of the truck or trailer orientation. The hitch angle was not included as a state because the trajectory planner for the forwards and backwards manoeuvres requires both the truck and trailer orientation, using the hitch angle as a state would just increase the number of states while adding no additional information about the vehicle.

Consider Figure 2: This diagram only considers the truck side of the vehicle to derive a relationship between the vehicle's steering angle, the vehicle's longitudinal velocity, the vehicle's yaw rate, and the vehicle's geometry.

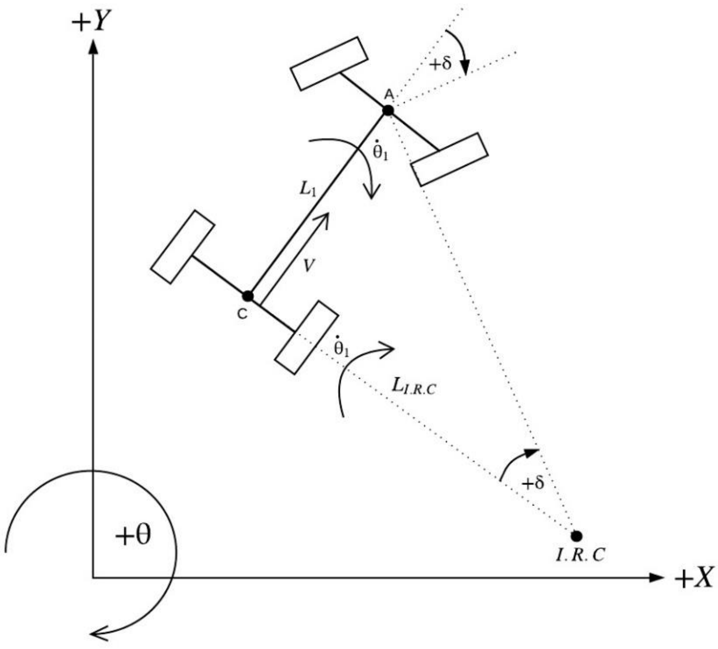


Figure 2, the diagram used to derive the relationship between the steering angle, the trucks angular velocity, longitudinal velocity, and the trucks length.

The truck in Figure 2 has a positive constant steering angle, this produces an instantaneous radius of curvature (IRC) some distance L_{IRC} away from the point C (the centre of rear axle). This constant steering angle produces a yaw rate $\dot{\theta}_1$, which is also dependent on the velocity of the vehicle. This relationship is captured in Equation 1. In Equation 2, the trigonometric relationship between the steering angle and the two lengths is defined, by rearranging the equation and substituting for L_{IRC} an expression for the yaw rate of the vehicle depending on velocity and steering angle can be found.

$$\dot{\theta}_1 = \frac{V}{L} = \frac{V}{L_{IRC}}$$ *Equation 1*

$$\tan\delta = \frac{L_1}{L_{IRC}} \quad \therefore \quad \dot{\theta}_1 = \frac{V}{L_1}\tan\delta$$ *Equation 2*

To find the relationship between the steering angle and the trailers yaw rate the truck and trailer needs to be considered, the velocities and geometry can be seen in Figure 3.

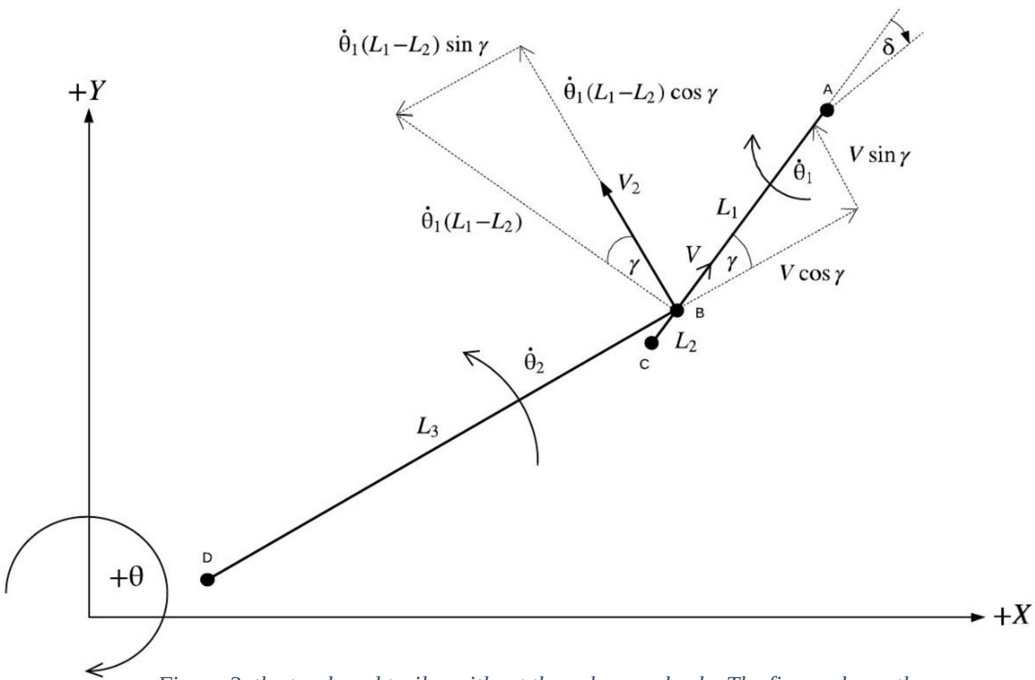


Figure 3, the truck and trailer without the axles or wheels. The figure shows the kinematic response to a velocity at point B and a steering angle input.

A positive constant velocity and steering angle are input into the vehicle, the truck side of the vehicle experiences a positive yaw rate described but Equation 2. The velocity vectors caused by the truck's yaw rate can be described at the hitching point (B). Similarly, the velocity vectors caused by the truck's longitudinal velocity (V), can be broken down into components in respect to the hitch angle, at the hitch point. Breakdown of these two velocity vectors allows the trailer's yaw rate to be solved at the hitch point, this can be seen in Equation 3. Using the relationship of angular velocity to linkage length and perpendicular velocity Equation 4 can be derived.

$$V_2 = V\sin\gamma + \dot{\theta}_1(L_1 - L_2)\cos\gamma$$ *Equation 3*

$$\dot{\theta}_2 = \frac{V_2}{L_3} \quad \therefore \quad \dot{\theta}_2 = \frac{V \sin\gamma + \dot{\theta}_1 (L_1 - L_2)\cos\gamma}{L_3} \qquad \text{Equation 4}$$

To find the position of the vehicle from the origin to either the rear of the trailer (point D) or the rear of the truck (point C), the velocity components of both $\dot{x}, \dot{y}$ need to resolved to the axes of the coordinate system. This will first be done from the rear of the truck and then translated to the trailer rear. These velocity equations then need to be integrated with respect to time to determine the position of the point. It's clear from Figure 3 that the velocity from C to A can be resolved to give Equation 5 and Equation 6.

$$\dot{x}_c = V \cos\theta_1 \qquad \text{Equation 5}$$

$$\dot{y}_c = V \sin\theta_1 \qquad \text{Equation 6}$$

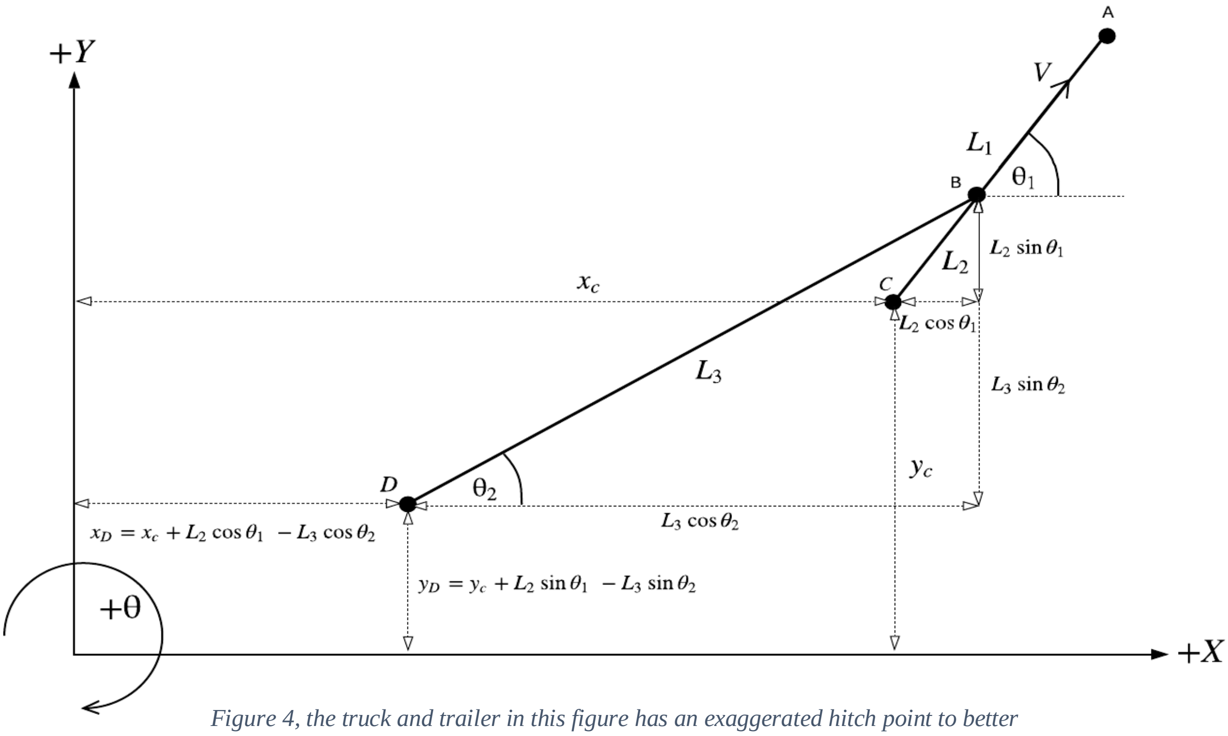


Figure 4, the truck and trailer in this figure has an exaggerated hitch point to better demonstrate the coordinate conversion. C to D.

Figure 4 is used to demonstrate how to convert from the position in Cartesian coordinates from the rear of the truck to the rear of the trailer. This will be used for when the vehicle engages in a reversing manoeuvre and the rear position of the trailer needs to be controlled. Equation 7 and Equation 8 show the conversion equations described in Figure 4.

$$x_D = x_C + L_2 \cos\theta_1 - L_3 \cos\theta_2 \qquad \text{Equation 7}$$

$$y_D = y_C + L_2 \sin\theta_1 - L_3 \sin\theta_2 \qquad \text{Equation 8}$$

In summary, the following equations (Table 2) will be used to obtain the states of the system for forwards and backwards manoeuvres. The state required are $x_C, y_C, \theta_1, \theta_2$ for forwards, and $x_D, y_D, \theta_1, \theta_2$ for backwards.

Table 2, the equations used to occur the necessary control states of the vehicle.

Equations used to get the Orientation of Truck and Trailer	
$\dot{\boldsymbol{\theta}}_1 = \dfrac{\boldsymbol{V}}{\boldsymbol{L_1}} \tan \boldsymbol{\delta}$ $\dot{\boldsymbol{\theta}}_2 = \dfrac{\boldsymbol{V} \sin \boldsymbol{\gamma} + \dot{\boldsymbol{\theta}}_1 (\boldsymbol{L_1} - \boldsymbol{L_2}) \cos \boldsymbol{\gamma}}{\boldsymbol{L_3}}$	
Equations used to get the Position of the Truck	Equations used to get the Position of the Trailer
$\dot{\boldsymbol{x}}_c = \boldsymbol{V} \cos \boldsymbol{\theta}_1$ $\dot{\boldsymbol{y}}_c = \boldsymbol{V} \sin \boldsymbol{\theta}_1$	$\boldsymbol{x_D} = \boldsymbol{x_C} + \boldsymbol{L_2} \cos \boldsymbol{\theta}_1 - \boldsymbol{L_3} \cos \boldsymbol{\theta}_2$ $\boldsymbol{y_D} = \boldsymbol{y_C} + \boldsymbol{L_2} \sin \boldsymbol{\theta}_1 - \boldsymbol{L_3} \sin \boldsymbol{\theta}_2$

3.3. Longitudinal Inertia Model

As discussed previously the vehicle is required to be controllable when loaded and unloaded. As the kinematic model doesn’t include mass the longitudinal accelerations will be constant for varying mass, this behaviour doesn’t represent the real vehicle. If the vehicle modelled didn’t include a longitudinal inertia model the vehicle would instantly reach the velocity requested, this would result in unrealistically fast manoeuvres when switching between forwards and backwards movement. Realistically the vehicle will have larger accelerations for an unloaded compared to loaded, for this reason a basic inertia model has been developed. In Figure 5 the forces action on the vehicle are evaluated.

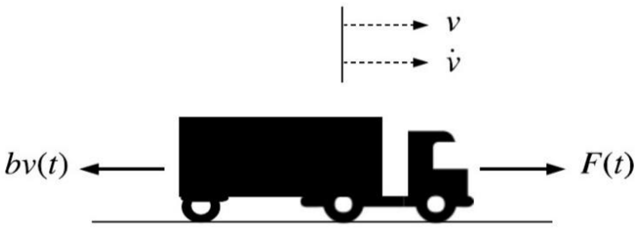


Figure 5, the forces acting on the vehicle as it accelerates.

The input into this system is the driving force generated by the rear wheels of the truck $F(t)$. The frictional forces are modelled on the basis of an increase in velocity will result in an increase in friction, no precise equation for aerodynamics or friction are used here, that level of accuracy is not required. Using Newton’s second law Equation 9 can be found. From this equation a solution can be found using the Laplace Transform for an open loop response

Equation *10*.

$$F = ma$$

$$F(t) - bv(t) = m\frac{dv}{dt}$$

$$m\frac{dv}{dt} + bv(t) = F(t) \qquad \textit{Equation 9}$$

$$L\left\{\frac{d^n f(t)}{dt^n}\right\} = s^n F(s) - s^{n-1} f(0) - s^{n-2}\frac{df(0)}{dt} - \cdots - \frac{d^{n-1}f(0)}{dt^{n-1}}$$

$$using\ above \ \rightarrow \ msV(s) + bV(s) = F(s)$$

$$G(s) = \frac{V(s)}{F(s)} = \frac{1}{ms + b} \qquad \textit{Equation 10}$$

This equation shows that the velocity response of the vehicle can be modelled as a first order transfer function. As the force from the drive chain is unknown at this point the exact Equation 10 can't be used, instead a time constant first order will be used, this is seen in Equation 11, K is the gain and τ is the time constant.

$$\frac{V(s)}{V_r(s)} = \frac{K}{\tau s + K} \qquad \textit{Equation 11}$$

3.4. Stability Analysis of Model.

In order to ensure the vehicle can be controlled the stability has to be considered. If a control action results in the vehicle becoming unstable then that action should be avoided. However, it was mentioned in the literature that the reversing of the vehicle is asymptotically stable, this was proven using Lyapunov Stability. This means that the vehicle is only in the stable region for a finite amount of time, before exiting the stable region the controller is required to generate inputs that then bring the vehicle back to the centre of this region.

By comparing the truck and trailer system to a double pendulum a useful analogy can be made. The vehicle driving forwards is inherently stable, when the truck turns the trailer will follow with a slight delay. This is equivalent to the double pendulum hanging from a fixed point, with gravity stabilising the two linkages. The velocity and steering angle inputs are similar to pulling a double pendulum along from the fix, clearly the system will always stabilise itself. When the vehicle is reversing it is equivalent to having both linkages above the fixed point with gravity constantly trying to destabilise the system. This system then needs to prevent the hitch angle (angle between the two linkages) from exceeding a critical value. See Figure 6.

For this reversing manoeuvre there are critical steering inputs and hitch angles that allow for asymptotic stability. Regarding the double pendulum again, if the angular velocity of the first linkage is too large for too long then the system will become unstable in one direction, if it is too small for too long it will become unstable in the other direction. For the vehicle these angular velocities are equivalent to the combination of vehicle velocity and steering angle. Figure 6 shows this stability analogy.

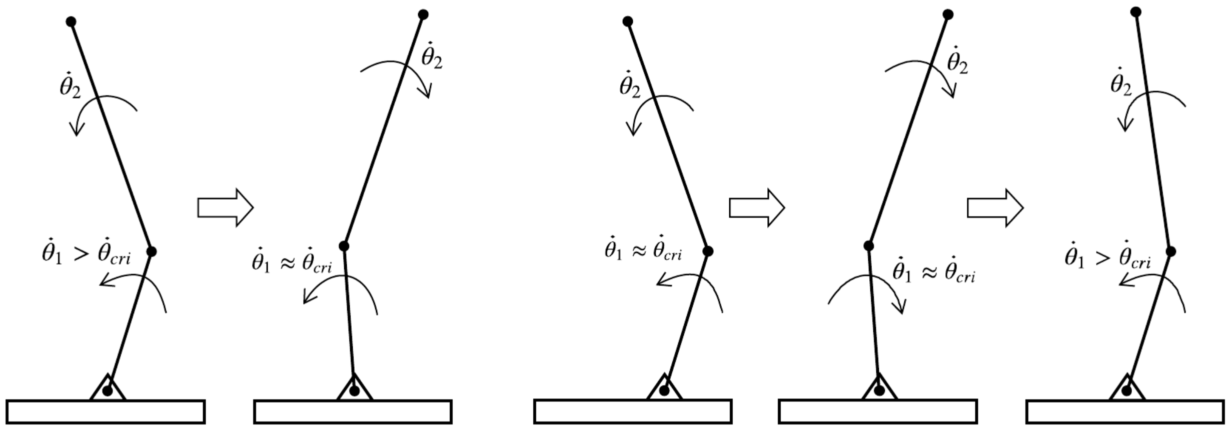


Figure 6, on the L.H.S the input angular velocity is larger than the critical value the system becomes unstable. On the R.H.S the input angular velocity is within the set of critical values the system stabilises.

It can be seen from Figure 6**Error! Reference source not found.** that the input angular velocity is too large for too long and that the system becomes unstable after some amount of time (shown on the left). However, if this angular velocity is controlled to stay within the set of critical angular velocities then the system will stabilise itself. From the vehicle it is clear that critical steering angles for specific hitch angles need to be determined, for simplicity the stable curve will be defined using a constant velocity of 2 ms^{-1}.

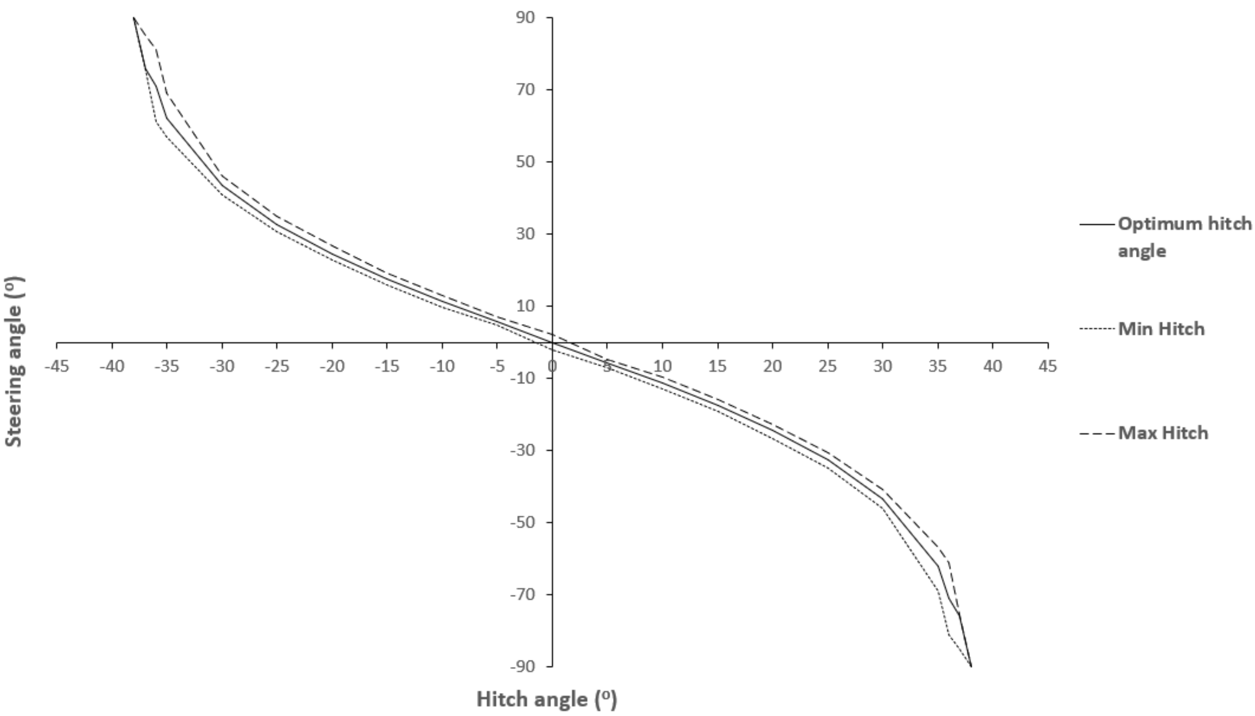


Figure 7, the region in which the reversing vehicle is asymptotically stable.

Figure 7 shows the stability behaviour of Ford vehicle reversing for a constant velocity of $2ms^{-1}$. The solid curve represents the optimum steering angle required to stop the hitch angle from changing (asymptotically stable). To reduce the hitch angle a steering angle needs to either be on the left or right side of the solid curve. E.g. if the hitch angle is -30 °, then the steering angle that will stop this hitch angle from change is 23 °. To reduce this hitch angle a steering angle on the left hand side of this point should be chosen, for example 22 °, this would reduce the hitch angle until the hitch angle reached the asymptotically stable point for this new steering angle. However, having the steering angle too far on the L.H.S will also cause the hitch angle to increase. For this reason a maximum and minimum region has been developed, this is shown between the dotted lines. These maximum and minimum regions expand if the velocity is reduced, allowing for a larger set of useful steering angles that will reduce the strain on the controller.

The way this graph was derived was by implementing the equations mentioned in Table 2 into Simulink. For information regarding this process see section 5.2. Once the equations were modelled in Simulink it was possible to run reversing simulations for the vehicle. In these simulations hitch angles in 5 ° intervals were selected, from these hitch angles constant steering angles were selected to see how long the system would take to grow the hitch angle by an arbitrary amount (0.1°). The steering angle that allowed the hitch angle to stay within this arbitrary amount for the longest period of time was then found iteratively to $\pm 0.05°$ of accuracy. A table of all the values can be seen in Appendix 3.4.1. The minimum and maximum curves were then defined by the steering angle required to keep the system asymptotically stable for more than $0.1s$.

It's important to mention that the asymptotically stable region is very small using the constraints above. For example, for a steering angle of 30 °the maximum hitch angle allowed is $-46°$ this is only asymptotically stable for $0.1s$ but takes $15.2s$ to reach the jack-knife condition ($hitch\ angle\ \geq\ \pm 90$ °). This means that the vehicle could still be stabilised within this $15.2s$ period.

This graph will be used to design the behaviour of the Fuzzy Controller responsible for stabilising the hitch angle while the vehicle reverses. The graph also shows that past a certain steering angle ($\approx\ \pm 37$ °) the vehicle is unrecoverable. It can also be seen that past $\pm 30°$ the range of values to control the vehicle become more extreme, for this reason the vehicle's steering input will be limited to ± 30 °. The maximum allowable hitch angle at this point was found to be ± 46 °, if the vehicle's hitch angle exceeds this the vehicle will lose control. If the vehicle reached this hitch angle while reversing it will drive forwards to reduce the hitch angle.

4. Controller Model

4.1. Fuzzy Control Theory

Fuzzy control theory was first introduced in 1965 by Lotfi A. Zadeh (Zadeh, 1965). The use of fuzzy controllers was first used in industry for the control of cement kilns. The use then progressed to being used in automated control of subway trains in Japan, regulate temperature on space shuttles and automation of vehicles. The most commonly used fuzzy control inference is the Mamdani fuzzy control structure and was among one of the first control systems to use fuzzy set theory.

Before fuzzy logic, computer programs made use of traditional logic, true or false decisions. Fuzzy logic allows for intermediate truth values, rather than definitive outputs (true or false) a fuzzy controller can have an output that is 60% true and 40% false. The control output is then a weighted average of these two percentages. Consider the following example of a bathroom tap: A control system using traditional logic, or crisp logic, would consist of two buttons, one for hot water and one for cold water. Only one button can be pressed at a time, this means that the tap will either output very hot water or very cold water. On the other hand, a fuzzy control system is analogous to the tap having one nob for hot and cold water, this nob can rotate clockwise to increase the percentage of cold water and anticlockwise to increase the percentage of hot water, the middle 90° will have a balance of hot and cold. This allows for a more comfortable water temperature to be achieved. Let the desired water temperature be 21 ℃, the two fuzzy inputs are the hot water (100 ℃) and the cold water (0 ℃). Therefore, the result can be achieved by using 21% of the hot stream and 79% of the cold stream. In the tap example the use of fuzzy logic and linguistic rules have been defined. However, the meaning of too hot or too cold is difficult for a computer to interpret. For this reason, Fuzzification, Inference and Defuzzification methods have been developed.

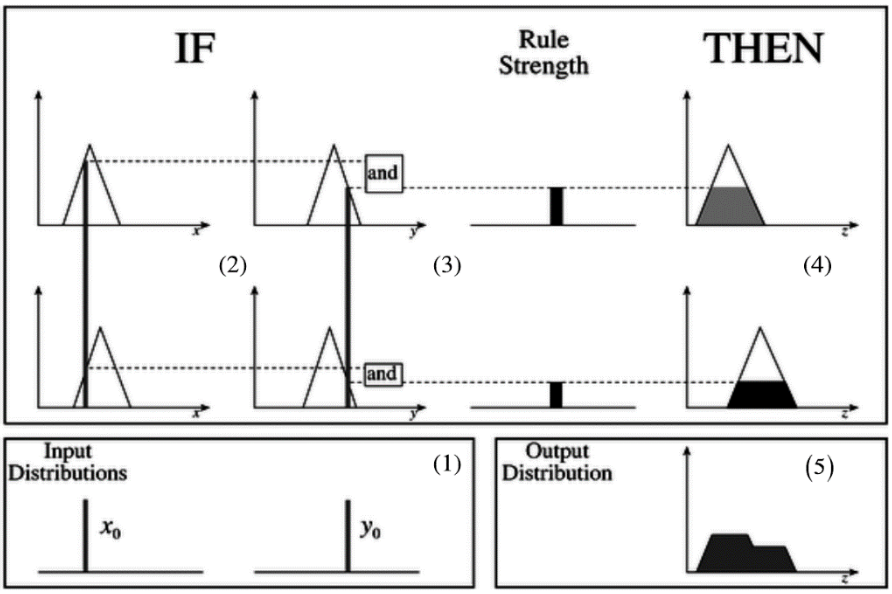


Figure 8, the process of Fuzzification, Inference and Defuzzification. (1) - (2) is Fuzzification, (3) - (4) is Inference, and (5) is Defuzzification (Sangalli, 2009).

For the following few paragraphs please referrer to Figure 8. The way a computer would control a system using fuzzy logic is to first convert the crisp logic into fuzzy sets. Fuzzy sets are constructed from a given universe U, the universe is the set of all possible inputs for that input. The set A, is then determined by a membership function $\mu_A(x)$ that translates these elements (x) of the universe to a degree of membership, which ranges from $[0, 1]$. In Figure 8, (1) Shows the crisp inputs and (2) shows the membership functions creating fuzzy sets for both inputs. The membership function can be described in many ways, however, in this thesis only triangular membership functions have been used. The mathematical representation of the function can be seen in Equation 12 and Figure 9.

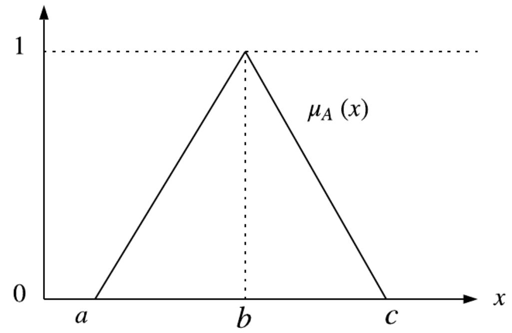


Figure 9, an example of a triangular membership function.

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$ Equation 12

This process of creating a fuzzy set essentially converts crisp logic into fuzzy logic, this is known as the fuzzification step in fuzzy control (Nainar, 1996). A general fuzzy set can be described in Equation 13, where A is the fuzzy set with limits $[0, 1]$, x is the crisp input, and $\mu_A(x)$ is the membership function.

$$A = \{(x, \mu_A(x)) | \, x \in U\}$$ Equation 13

Going forwards the definitions of fuzzy "and", "or" need to be understood. For this process there are many methods, in this thesis Zadeh method has been used and can be described in Equation 14 and Equation 15. These "and", "or" operations simply take the smallest or largest (respectively) value out of the fuzzy sets.

$$A \cap B = T(\mu_A(x), \mu_B(x)) = MIN(\mu_A(x), \mu_B(x))$$ Equation 14

$$A \cup B = T(\mu_A(x), \mu_B(x)) = MAX(\mu_A(x), \mu_B(x))$$ Equation 15

The system has a crisp input, this input is then turned into a fuzzy set using the membership functions associated with that crisp input. If there are multiple inputs then multiple fuzzy sets will be created, the "and", "or" definitions above have two crisp inputs that then create two fuzzy sets A and B. Once these two fuzzy sets have been obtained they go through the "and" operation shown in Equation 14, and the smallest of the two values is carried forwards. The "and" process seen in Figure 8 is used at (3) to find the minimum value from the two fuzzy sets. The value that has been carried forwards is then used to determine the cut-off point to limit the area of the output membership function. The association of the value outputted from the "and" operation to a specific output membership function is determined by the linguistic rules. Linguistic rules are conditional logic in the form of $IF\ A\ THEN\ Y$, where A is a fuzzy input set and Y is a fuzzy output set. These logical conditions correspond accurately to the actions of human operators, often referred to as experts. For the above tap example linguistic rules would follow such structure:

$$IF\ too\ hot\ THEN\ turn\ nob\ clockwise$$
$$IF\ too\ cold\ THEN\ turn\ nob\ anticlockwise$$
$$IF\ the\ temperature\ is\ good\ THEN\ do\ nothing$$

In Figure 8 the linguistic rules used the output "and" value from (3) combined with the rule strength to associate it to the defined rule, or output membership function (4). Once the areas of the output membership functions have been limited from these "and" values they are combined using the "or" operation mentioned in Equation 15, shown in Figure 8 going from (4) to (5). The result from combining these limited output membership functions with an "or" operation is called the output distribution. There are many methods to convert this output distribution back into crisp output values, one of which is the centre of mass method, this is shown in Equation 16. Where z is the output crisp value, μ_c is the output membership function at value z_j (Sangalli, 2009).

$$z = \frac{\sum_{j=1}^{q} z_j \mu_c(z_j)}{\sum_{j=1}^{q} \mu_c(z_j)} \qquad \text{Equation 16}$$

4.2. Forwards Controller

The forward controller is responsible for driving the vehicle from the initial position in the parking space to just before the pedestrian walkway. As outlined in section 3, the forward controller is also responsible for aligning and driving the vehicle forward in the case that it doesn't fit inside the parking space after a reversing manoeuvre. This process will be will be controlled using the following input and output states: inputs x_C, y_C, θ_1, output δ.

In order for the vehicle to drive it requires a reference trajectory, this will then feed into the forward controller outputting a steering angle. The velocity of the vehicle is determined using another system, which will be discussed later in section 5.3. The vehicle's path planning system will be constructed using a fuzzy controller, this process was seen in the literature (Riid, Lei and Rüstern, 2009), and the same logic will be applied here. The way the path planning works is by assigning a truck orientation to a particular position in the parking space geometry. The input to this path planning unit will be the x, y positions truck at point C, and the output will be a requested (reference) truck orientation angle. This orientation angle will then be subtracted from the actual truck orientation in a feedback loop to produce a truck orientation error. This will then be fed into another fuzzy controller that will try to reduce this truck orientation angle to ensure the trajectory generated is followed accurately, see Figure 14. The first iteration of the forwards path planner is shown in Figure 10.Figure 9, an example of a triangular

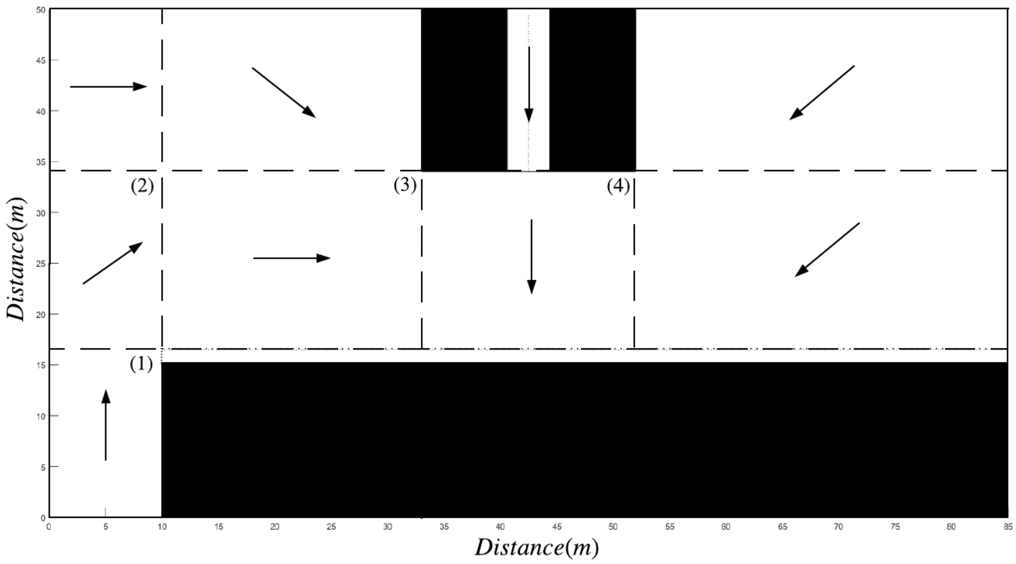


Figure 10, the diagram showing the truck orientation angles assigned to different sections of the parking space.

membership function.

When the vehicle enters the parking space, on the bottom left (1), the truck orientation angle requested is -90 °. The vehicle then leaves the entrance and a second truck orientation is requested (2), this time -45 °is requested. The vehicle then moves into (3) and the requested angle is 0 °, finally the vehicle moves into (4) were it tries to align itself with the centre of the parking space line having a requested angle of 90 °. Other areas are defined with other orientation angles to prevent the vehicle from colliding with the parking spaces boundaries or overshooting the desired final position. This can concisely be summarised by the membership functions, Figure 11, and the linguistic rules Table 3**Error! Reference source not found.**. Table 3 and Figure 11 show the first iteration of the path planner, this system will be improved for better performance in this section.

Table 3, the first iteration of linguistic rules for the forward path planner

$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PS\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NM$$
$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NS$$
$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ Z$$
$$\boldsymbol{IF}\ Xposition\ is\ PS\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ Z$$
$$\boldsymbol{IF}\ Xposition\ is\ PS\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PS$$
$$\boldsymbol{IF}\ Xposition\ is\ PM\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PM$$
$$\boldsymbol{IF}\ Xposition\ is\ PM\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PM$$
$$\boldsymbol{IF}\ Xposition\ is\ PL\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PL$$
$$\boldsymbol{IF}\ Xposition\ is\ PL\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PL$$

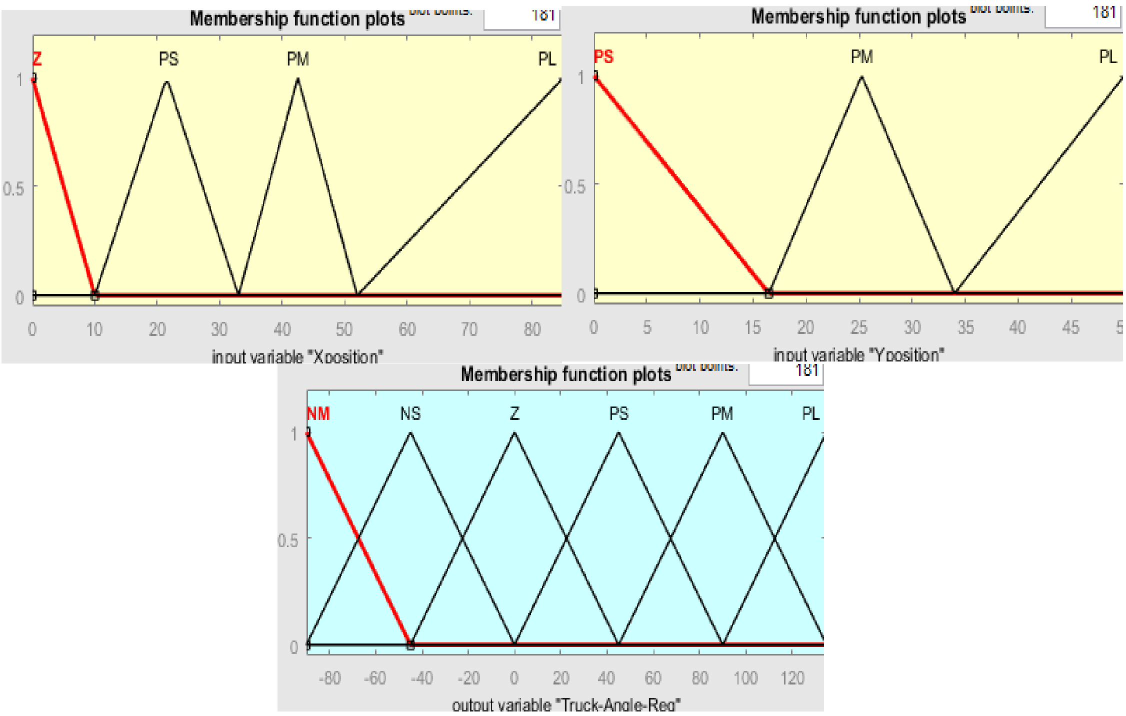


Figure 11, the input and output membership functions of the forward path planning controller. The top is the membership functions for the trucks x position. The top right is the membership functions for the y position. The bottom figure is the output membership functions determining the trucks orientation. Units: Meters and Degrees.

Now that the path planner has been defined another fuzzy controller will be used to ensure the vehicle obtains the requested truck orientation. As driving forwards in the vehicle automatically stabilises the vehicles trailer the hitch angle does not need to be controlled here, only the error between the requested truck orientation from the forwards path planner and the actual truck orientation is needed. This means the fuzzy controller can be a simple single input single output system. The linguistic rules can be seen in Table 4 and the controllers input and output membership functions can be seen in Figure 12.

Table 4, the linguistic rules for the forward fuzzy controller.

IF *Truck Orientation Error is NL* ***THEN*** *Steering Agnle is NL*
IF *Truck Orientation Error is NM* ***THEN*** *Steering Agnle is NM*
IF *Truck Orientation Error is Z* ***THEN*** *Steering Agnle is Z*
IF *Truck Orientation Error is PM* ***THEN*** *Steering Agnle is PM*
IF *Truck Orientation Error is PL* ***THEN*** *Steering Agnle is PL*

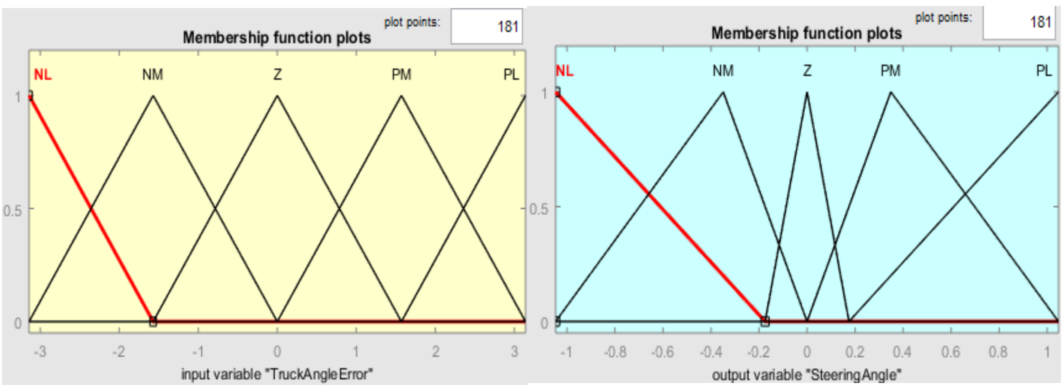


Figure 12, the membership functions of the forwards fuzzy controller. Input on the left, output on the right. Units: Radians

Figure 12 and Table 4 can be interoperated as follows: If there is a large negative truck orientation error then the vehicle tries to correct this with a large negative steering angle, this reduces the size of the orientation error. If there is no, or very small orientation error then the vehicle has zero or a very small steering angle. If the vehicle has a large positive truck orientation angle then the control outputs a large negative steering angle to reduce the error. This can be more easily understood from Figure 13.

The output membership function seen on the right of Figure 12 has a concentration around the output steering angle of 0° this makes the response to large orientation errors large and makes the response to small orientation errors smaller. This prevents the vehicle's steering angle oscillating around the desired value as the smaller response slows down the rate of steering angle.

In Figure 13 the vehicle has a negative truck orientation that is less than the negative requested truck orientation, this produces a negative truck orientation error. The vehicle then needs to turn towards the orientation angle to reduce it, this requires a negative steering angle. The same behaviour can be described for the above membership functions and linguistic rules.

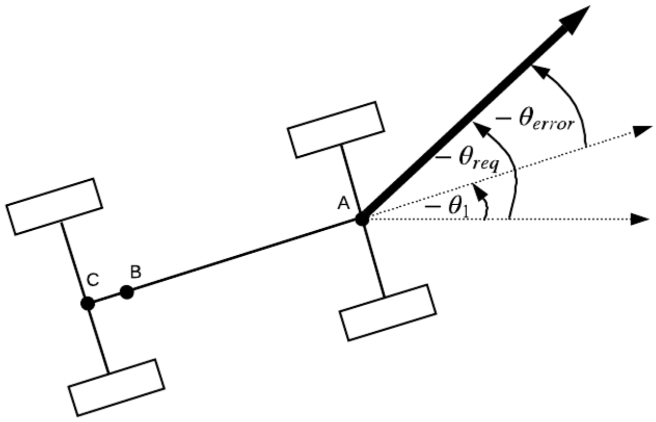


Figure 13, the vehicle is shown without the trailer. The truck shows a negative orientation, the requested orientation is a large negative angle, therefore, the error between them is also negative.

The forward fuzzy controller and path planner can be seen together in Figure 14. In Figure 14 the forward fuzzy path planner (FFPP) has two inputs, the truck's x_C, y_C positions, this then outputs a requested truck orientation angle. The actual truck's orientation angle and the requested angle are subtracted, and the error goes into the forward fuzzy controller (FFC), this then outputs a steering angle to reduce the error in orientation.

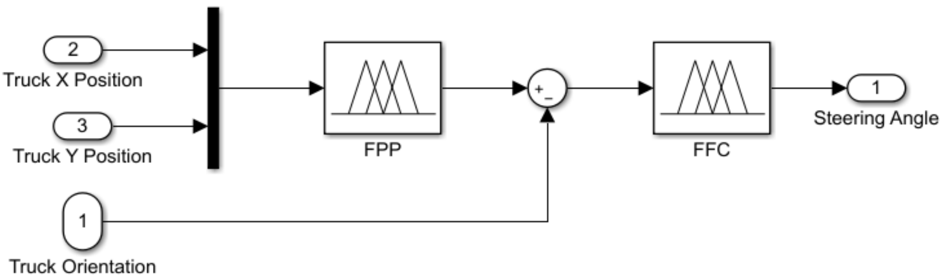


Figure 14, the Simulink schematic of the two forward fuzzy controllers.

The FFC developed to reduce the error of the truck orientation angle didn't need any further tweaking, it produced good results from the first iteration. However, the design of the forward path planning controller required multiple iterations. Figure 15 shows the first simulation of the vehicle, the initial conditions of the vehicle are $x_C = 5, y_C = 0, \theta_1 = -90°, \theta_2 = -90°$.

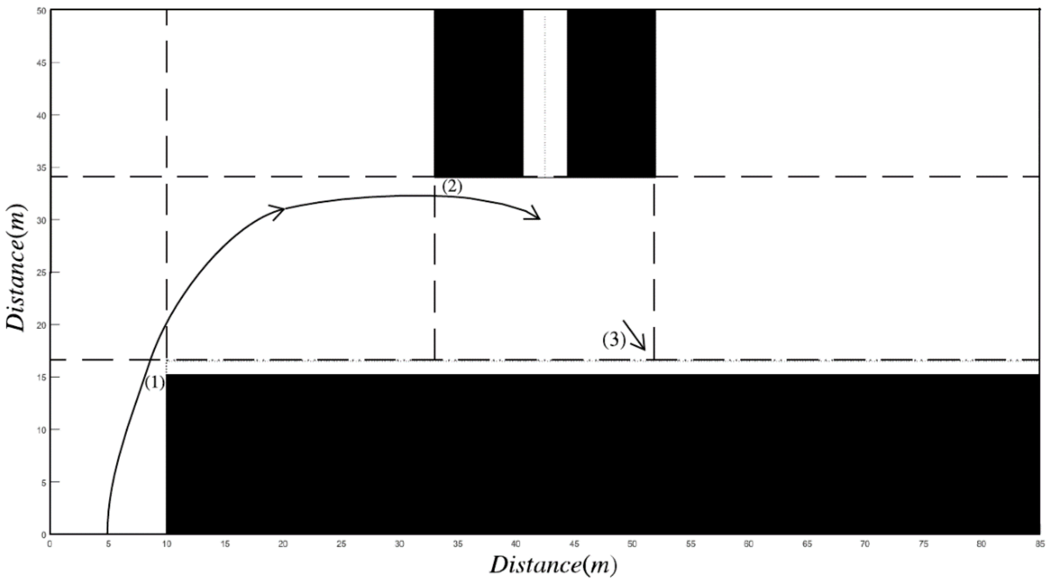


Figure 15, the trajectory of the vehicle tracked when following the first iteration of the FFPP. The curved line shows the path traced out by the x, y position of the truck.

The trajectory in Figure 15 is not satisfactory. One of the reasons is that the only simulations with initial x positions less than 5m won't crash into the side of the parking lot, the vehicle gets close to a boundary at (1) and at (2). The second problem is that the vehicle doesn't align itself with the desired parking space very well. If the simulation kept running the vehicle would reach (3) before stopping. It's important to keep in mind that the trajectory plotted is from the centre of the vehicle, this means that there needs to be clearance between obstacles and the trajectory. The trailer is the widest part having a width of $2.55m$ therefore a clearance of $1.275m$ is required. An overshoot from the desired parking space is required, this will make the reversing manoeuvre easier as the trailer will already be pointing in the direction of the parking space. Currently the vehicle unintentionally overshoots.

For the trajectory to be acceptable the vehicle would not collide with the boundaries of the parking space, e.g. (1), there should be at least $1.275m$ of space between the curve and the boundaries. The trajectory line should also leave 1.275m from the parking space, e.g. (2), as other vehicles might be parked there. This should be true for all initial x_C positions $1.275m \leq x_C \leq 8.725m$, currently it is only true for $1.275m \leq x_C \leq 5m$. When the truck reaches the position around (3), it should have an orientation close to 90°, this will allow for the trailer, which has a slight lag, to have an appropriate orientation for reverse parking. For example, if the truck has an orientation of 90° then the trailer will have an orientation of around 80° due to this lag.

The reason the vehicle preemptively turned while still inside the parking space entrance, at (1), was because the range of truck orientations for that rule were large, around $-90°\ to - 45°$, this resulted in the vehicle not following the desired vertical path. To fix this the sharpness of the output membership functions were changed for the $-90°, 0°\ and\ 90°$ outputs. In theory this makes the requested truck orientation angle have a much smaller range. The new possible range of truck orientations in the parking space entrance is now $-90°\ to - 80°$. The changes to the output membership functions can be seen in Figure 16.

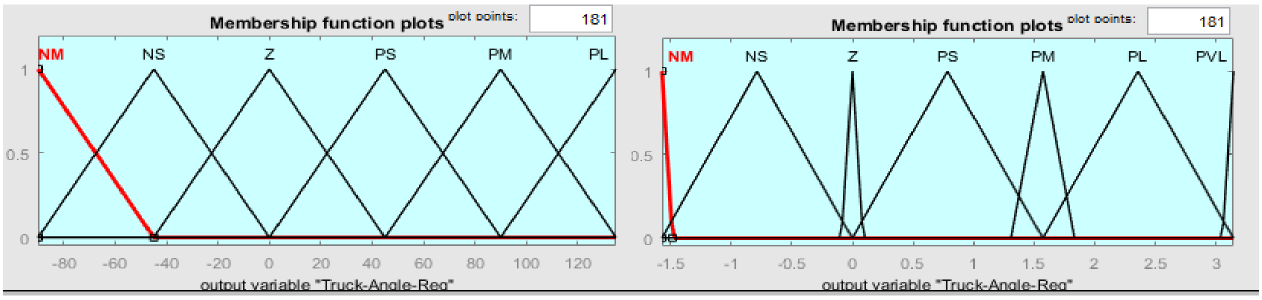


Figure 16, the output membership functions on the left are the first iteration (degrees), and the ones on the right have the discussed changes (this graph is in radians).

In Figure 16 the PM membership functions is not as sharp as the others, this is because this rule is used in area (4) of Figure 10. If the requested steering angle was too sharp in this area the vehicle wouldn't overshoot the parking space. A PLV membership function has been added to request a larger orientation angle 180° in the spaces on the far right of the parking lot. To ensure the overshoot can be controlled the x position input membership functions were edited. To ensure that the truck's orientation angle converges to 90° at a desired point, say $x = 43.5m$, then the membership functions need to overlap to direct the truck to this x position. The change in this x position membership function can be seen in Figure 17. Changing the x and y position membership functions change the areas the parking space is divided into. The trajectory using this new output membership functions can be seen in Figure 18.

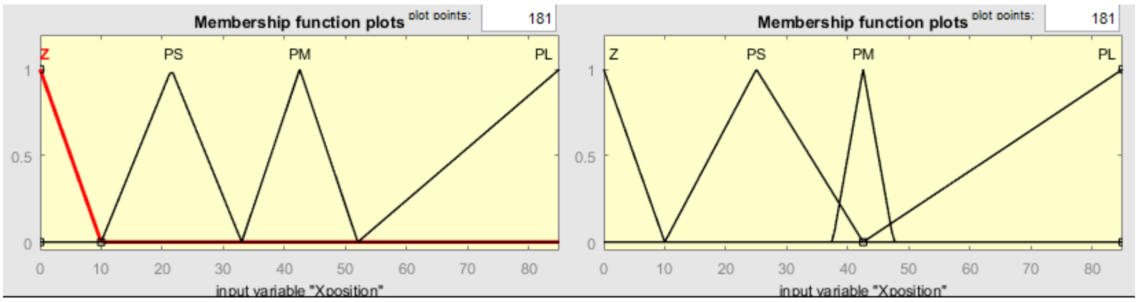


Figure 17, shows the change in x position membership functions. The right is the improved membership functions. Units: Meters

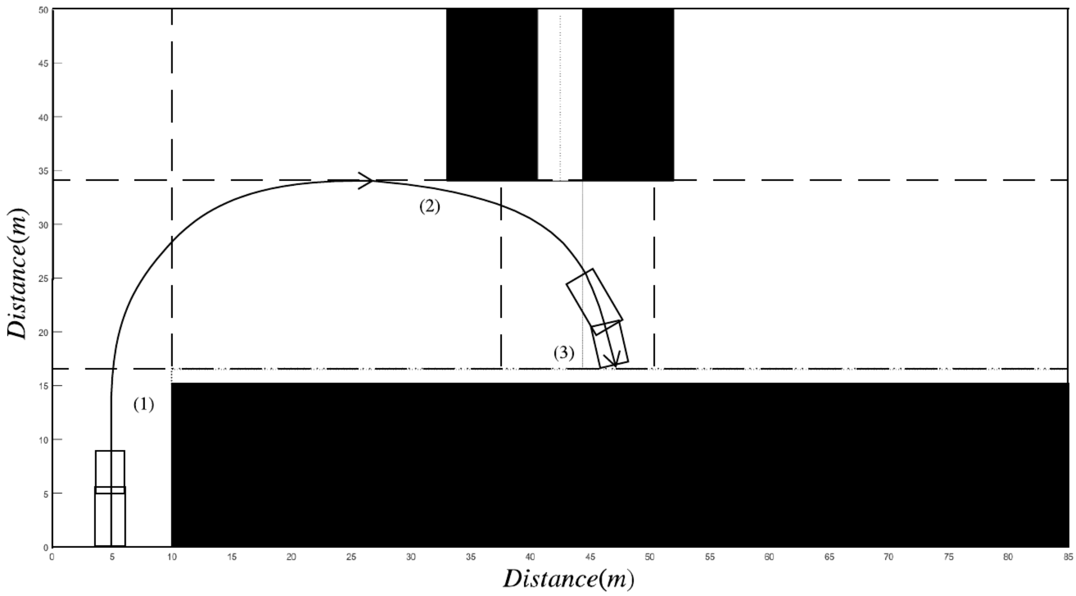


Figure 18, the vehicles trajectory with the changes made to both the x position and steering angle membership functions.

Looking at (1) in Figure 18, it's clear that the issue with limited starting x positions has been solved, with the vehicle requesting around $-90°$ in this space. Figure 18 shows the new trajectory, the vehicle has been included in this figure to illustrate the lag issue with the trailer. When the vehicle gets to the pedestrian walkway (3) the truck still hasn't reached $90°$, this means that the trailer orientation is smaller, around $75°$. This results in the trailer having a smaller orientation that in-turn will make reversing more difficult. To help correct this, the truck should reach $90°$ when it stops before the pedestrian walkway. Another problem with this trajectory is that the vehicle goes over other parking spaces, this is shown at (2).

To stop the vehicle from driving over the parking spaces the y position membership function was overlapped to allow for a converging position in the y direction in the area (2). The converging position is the y position the vehicle will drive to in this area. To ensure the vehicle didn't go over the parking space the y position was chosen to be $32.7m$, this was found by subtracting the parking space's y position from half the trailer's width $34 - \left(\frac{2.55}{2}\right) \approx 32.7m$. This change can be seen in Figure 19. This fix along with small adjustments to the sharpness of the output steering angles result in good forward trajectories. Figure 20 shows the new trajectory.

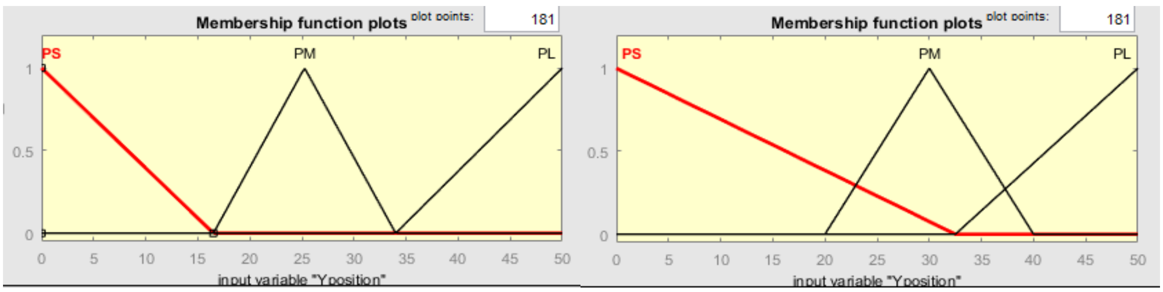


Figure 19, the change in y position membership functions to stop the vehicle from driving over the parking space. Units: Meters

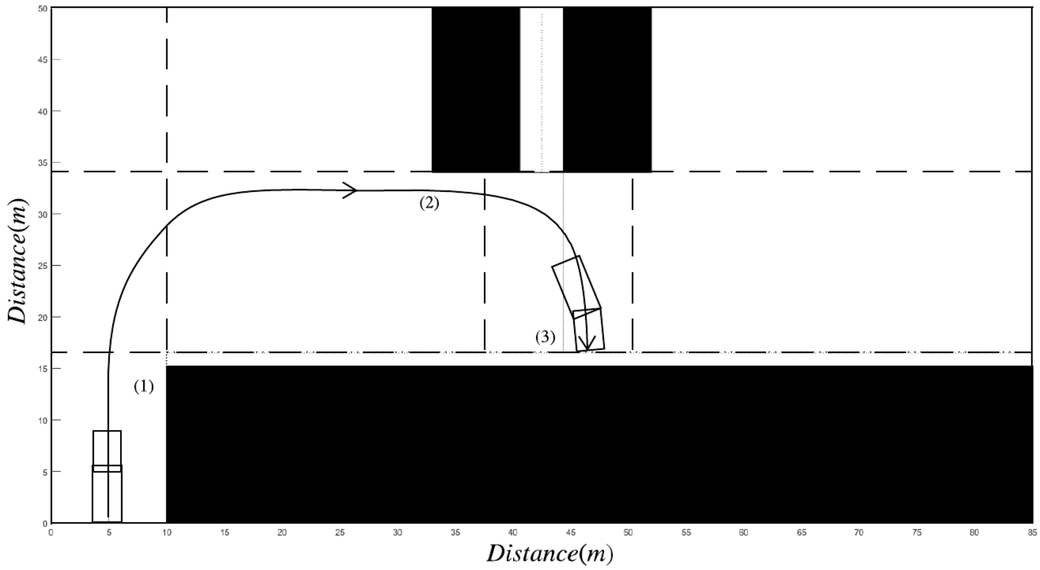


Figure 20, the trajectory of the vehicle from simple initial conditions.

From Figure 20 it can be seen that the vehicle now doesn't go over the parking space, this change also allows the vehicle to better align itself with 90° at the desired $x = 43.5m$ point. This desired x position was chosen between the desired overshoot position and the centre line of the parking space. The reason for this is to allow for the manoeuvre to the pedestrian crossing to achieve a desirable position, around $x = 47m$. However, if the vehicle fails to reverse and the rules are strong the vehicle will drive back to $x = 47m$, this could cause a loop of forwards and backwards manoeuvres that wouldn't park the vehicle. If the rules are less strong then the vehicle will overshoot $x = 43.5m$ for the first manoeuvre but then drive to this location after a failed reversing manoeuvre, this allows the vehicle to then attempt the reversing manoeuvre from a more desirable location. This better alignment of the truck then results in the angle of the trailer being larger, this will make the parking manoeuvre easier for the controller. The control strategy was also tested for a forward's manoeuvre after a failed reversing attempt, the trajectory can be seen in Figure 21.

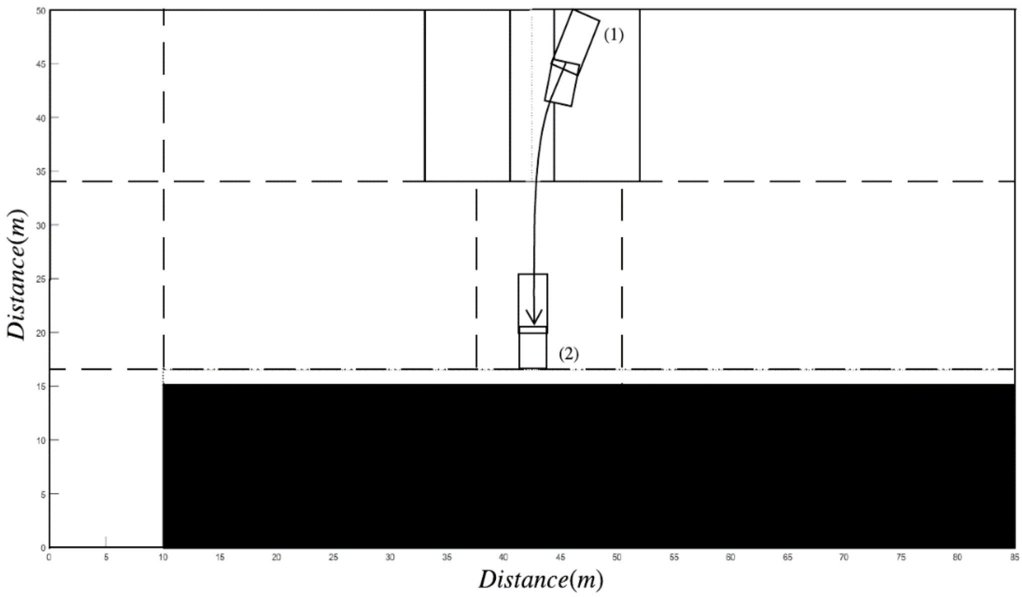


Figure 21, the trajectory of the vehicle from an imaginary missed reversing scenario.

Nevertheless, the FFPP controller has limitations. For the trajectories used in the above simulations the initial truck and trailer orientations have been simple, the initial x, y positions haven't been changed and the orientation is $-90°$ for truck and trailer. This iteration of the controller is unable to safely complete the forward manoeuvre when it is either on the extreme right or left of the parking space entrance. When in the extreme left the vehicle goes over the parking space, when on the extreme right is collides with the edge of the parking boundary ((1) in Figure 20). The vehicle also has problems when the initial orientation is far from $-90°$, if the initial orientation angle deviated far from this value it would crash into the sides of the parking lots boundaries.

The way to correct the initial orientation sensitivity is to have fuzzy rules added to the x position input membership functions. These rules will request a truck orientation that pushes the vehicle away from the boundaries found on the extreme left and right of the entrance. The problem where the vehicle drives over the parking space occurs when its initial x position is to the extreme left, this can be fixed by adding additional y position membership functions to push the vehicle down onto the desired $y = 32.7m$ line before reaching the parking space. The difficulty in doing this is that by changing the rules or membership functions the previously tested simulations might be unsuccessful. For this reason, additional membership functions and rules were added instead of editing the existing ones. The final membership functions can be seen in Figure 22, Figure 23 shows the surface of the FFPP, and Table 5 shows the rules.

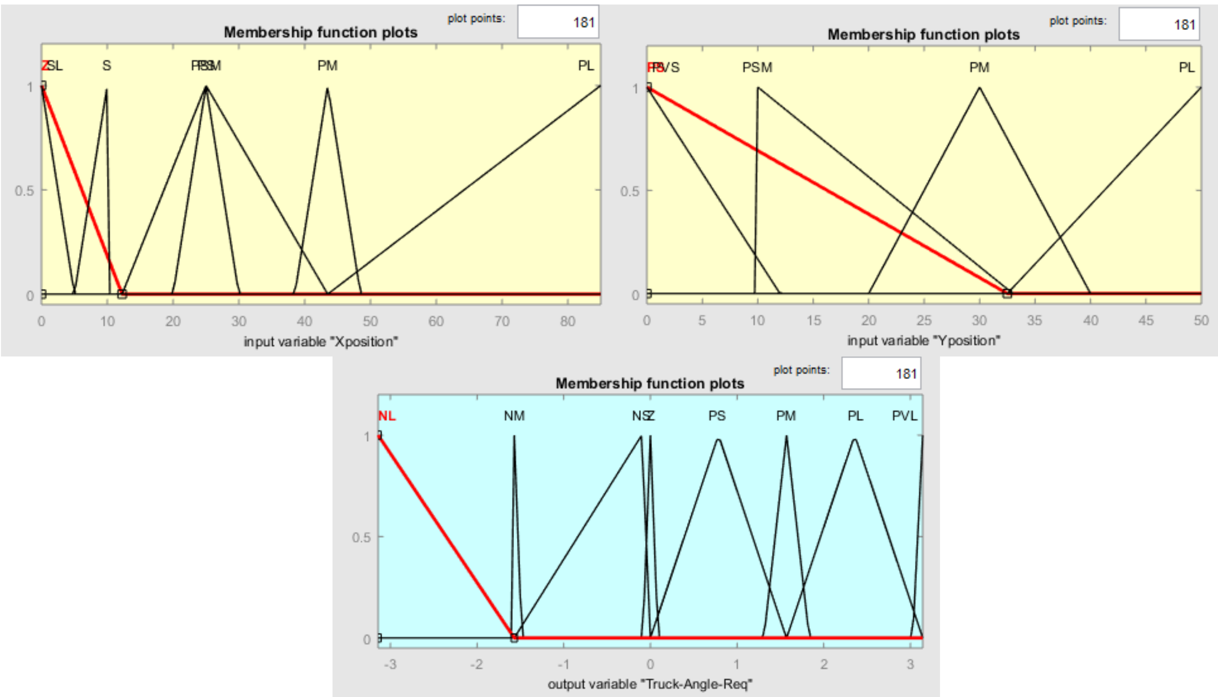


Figure 22, the final membership functions for inputs and output. Radians and Meters used.

Table 5, the final rules for the Fuzzy Path Planner.

$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PS\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NM \quad 1$$
$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NS \quad 2$$
$$\boldsymbol{IF}\ Xposition\ is\ Z\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ Z \quad 3$$
$$\boldsymbol{IF}\ Xposition\ is\ PS\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ Z \quad 4$$
$$\boldsymbol{IF}\ Xposition\ is\ PS\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PS \quad 5$$
$$\boldsymbol{IF}\ Xposition\ is\ PM\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PM \quad 6$$
$$\boldsymbol{IF}\ Xposition\ is\ PM\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PM \quad 7$$
$$\boldsymbol{IF}\ Xposition\ is\ PL\ \boldsymbol{AND}\ Yposition\ is\ PM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PVL \quad 8$$
$$\boldsymbol{IF}\ Xposition\ is\ PL\ \boldsymbol{AND}\ Yposition\ is\ PL\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ PL \quad 9$$
$$\boldsymbol{IF}\ Xposition\ is\ PSM\ \boldsymbol{AND}\ Yposition\ is\ PSM\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NS \quad 10$$
$$\boldsymbol{IF}\ Xposition\ is\ S\ \boldsymbol{AND}\ Yposition\ is\ PVS\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ NL \quad 11$$
$$\boldsymbol{IF}\ Xposition\ is\ SL\ \boldsymbol{AND}\ Yposition\ is\ PVS\ \boldsymbol{THEN}\ Truck\_Angle\_Reg\ is\ Z \quad 12$$

The addition of SL and S membership functions in the x position input, PVS and PSM in the y position inputs and NL, NS in the output membership function were used to push the vehicle away from the sides of the parking entrance, using rules 10, 11 and 12. These membership functions can be seen in Figure 22, these affects can also be seen in the fuzzy surface Figure 23. The effects can be seen in the quadrant $Xposition\ [0,20]\quad Yposition\ [0,20]$. The steep surface slops on either side push the vehicle away from the parking boundaries.

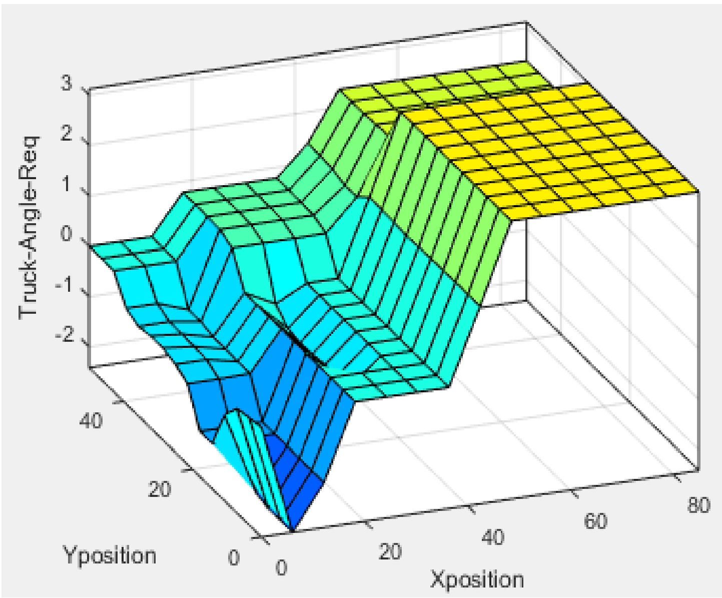


Figure 23, the fuzzy path planning controller surface. The entire controller behaviour can be analysed from here. Units: Radians and meters.

In Figure 23 the vehicles initial position is somewhere in-between $Xposition\ [0,20], Yposition\ [0,20]$ with a truck and trailer orientation angle. In this position the FFPP pushes the vehicle onto the $x = 5m$ line between the two slopes until around $Ypsotion\ \geq 17m$. Once the y position exceeds this value the vehicle reaches a slope that gives the vehicle a new orientation of around $-1rad\ \equiv -57°$ this starts to turn the vehicle into the main area of the parking space. The slope described can be found in Figure 23 in the area $Xposition\ [0, 10]$, $Yposition\ [20, 35]$. This slope directs the vehicle to the right of the parking space making the x position increase, as the x position increases the truck orientation desired reduces to 0°. If the vehicle starts on the L.H.S of the entrance the vehicle will enter into the dip described in $Xposition$ $[20, 25]$, $Yposition$ $[15, 40]$ this will stop the vehicle from over shooting the desired $y = 32.7m$ line. This dip is also used to help reduce the trailer orientation lag experienced by the vehicle when it reaches the pedestrian walkway. The vehicle then reaches the large slope described in the rough area $Xposition$ $[39, 46]$, $Yposition$ $[0, 36]$. This slope is used to align the vehicle with the desired position $x = 43.5m$, if the vehicle overshoots this line the controller will request an angle that points it back towards this position.

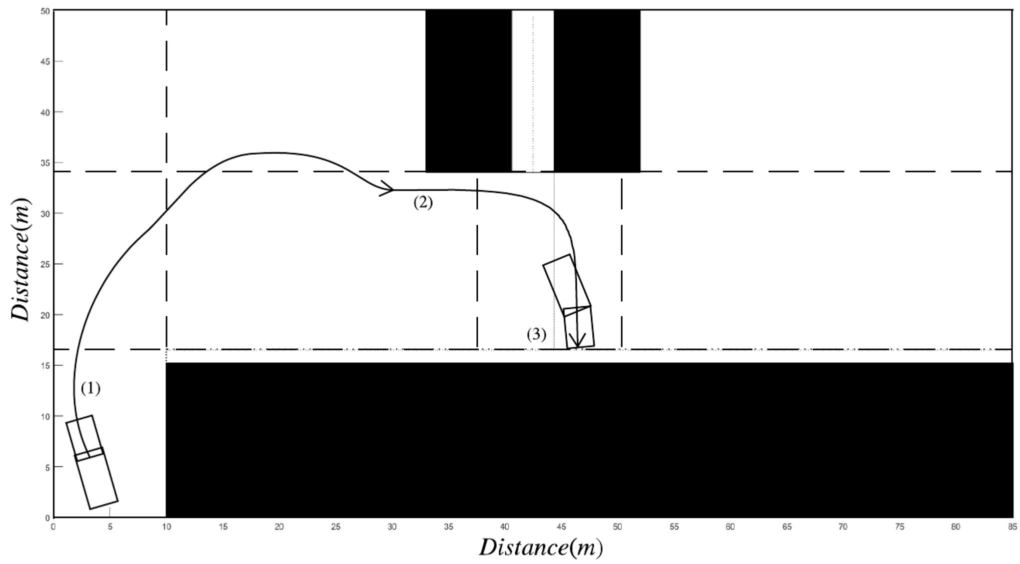


Figure 24, the trajectory of the vehicle using the final membership functions and rules. Initial conditions have the vehicle pointing towards the left side of the parking entrance.

Figure 24 shows the trajectory described in the above paragraph. The initial conditions have the vehicle pointing toward the left side of the parking entrance. At (1) in Figure 24 the controller works to push the vehicle away from the side of the parking entrance, the vehicle then slowly turns clockwise to get to position (2). While getting to position (2) the vehicle overshoots the desired y position in this area, this overshoot was difficult to avoid while having the vehicle successfully drive forwards for all initial conditions. However, the slight overshoot increases the final trailers orientation making it easier to complete the reversing manoeuvre. The position (3) shows the vehicle reaching the pedestrian walkway with the desired position and orientations. The entire manoeuvre is completed without hitting or going over the parking spaces boundaries.

4.3. Reversing Controller

The aim of the reversing controller is to drive the vehicle backwards into the desired parking spaces central line, the controller should actively reduce the hitch angle while simultaneously achieving the desired truck and trailer orientations. The desired final position of the vehicle is $x_D = 42.5m, y_D = 49.85m$, this leaves 15cm between the trailer rear and the dock. The vehicle should have a desired trailer orientation within $\pm 2°$ of the desired 90° orientation, the truck should have an orientation error less than $\pm 5°$, also for 90°. The truck can have a larger orientation error than the trailer because the orientation of the truck isn't as important, so long as it fits inside the parking space and allows for the driver to easily exit, $\pm 5°$ allows this. Figure 25 shows the schematic layout of the reversing fuzzy control system. The required states to control the system are: inputs $x_D, y_D, \theta_2, \theta_1$, outputs δ.

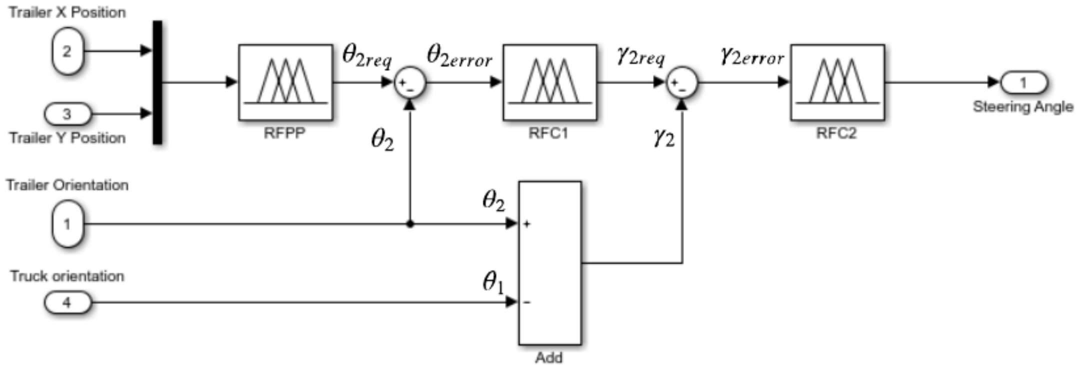


Figure 25, the fuzzy control system used to reverse the vehicle.

The first aim of the control system is to generate a reference trajectory for the vehicle. This reference trajectory is generated by the Reversing Fuzzy Path Planner (RFPP) seen in the figure above, this path planner uses the same principles as the forwards path planner.

Figure 26 shows the path planner for the reversing maneouvre, there only needs to be trailer requested orientation inside a small region of the parking lot, this is because the vehicle never exceeds these limits after the first forward maneouvre. The path planner uses the trailer's orientation rather than the truck's orientation to control the trailer, this is because the primary object to control is the trailer. The final position of the trailers rear needs to have a final position in the following rages: $42.3m \leq x_D \leq 42.8m$, $49.7m \leq y_D \leq 49.85m$.

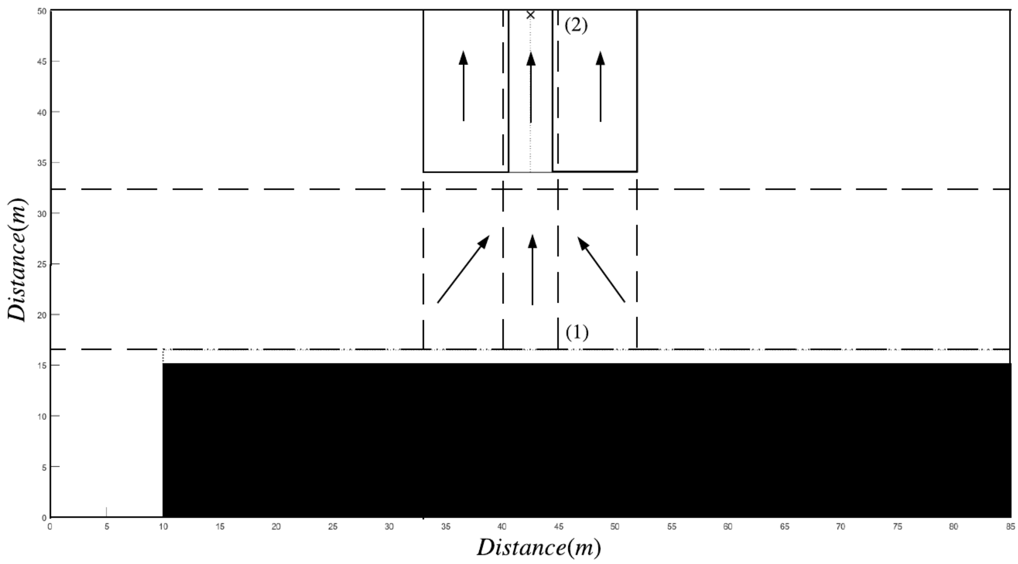


Figure 26, the requested trailer orientation for the vehicle in different areas of the parking space.

Once the requested trailer orientation has been output from the RFPP it is subtracted from the actual trailer orientation to give the orientation error. This orientation error is then fed into the Reverse Fuzzy Controller 1 (RFC1) that decides the optimium hitch angle required to ensure this trailer orientation can be achieved. The output of the RFC1 is a desired hitch angle. The desired hitch angle is then subtracted from the actual hitch angle to give the hitch angle error, this is then input into Reverse Fuzzy Controller 2 (RCF2). RCF2 takes the hitch angle error and outputs a steering angle that allows for the desired hitch angle to be achieved, this allows for the desired trailer orientation to be achieved. By using RFC1 and RFC2 the hitch angle can be controlled to prevent the jack-knifing phenomenon and achieve a desired final position and orientation.

Similar to the FFPP the RFPP involved an iterative design approach. Initial membership functions and linguistic rules were formulated, the membership functions and the surface can be seen in Figure 27. Inguistic rules can be found in Table 6.

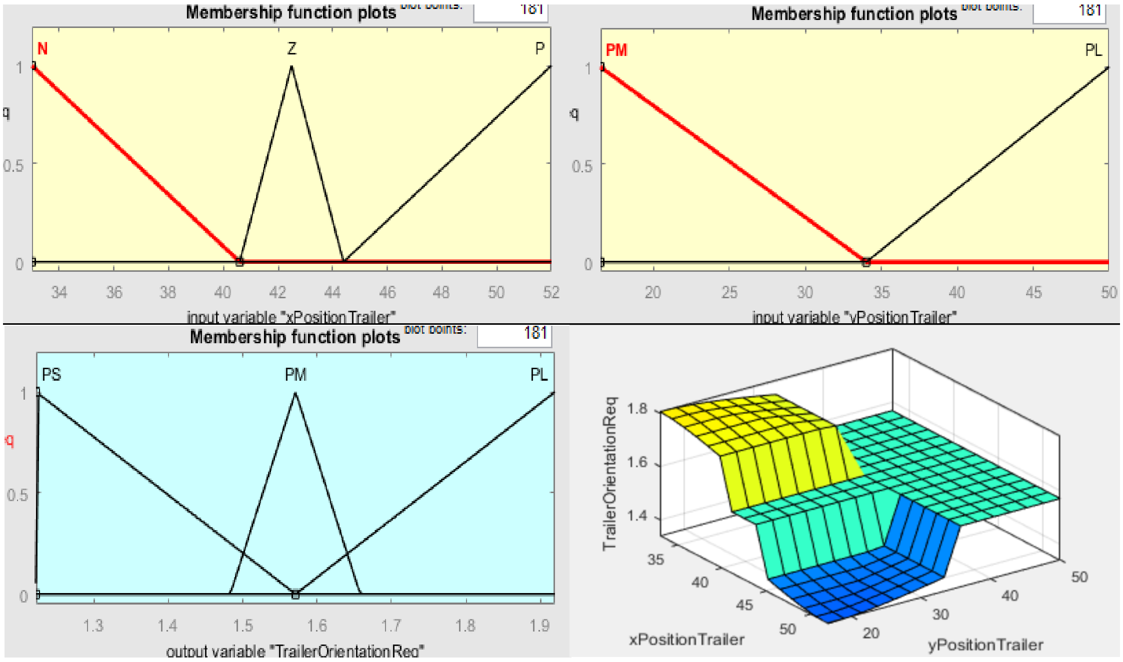


Figure 27, the two input variables, the output variable function, and the shape function of the RFPP. The output membership functions are in radians and meters.

Table 6, the linguistic rules for the RFPP

$$\boldsymbol{IF}\ xPositionTrailer\ is\ N\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PL$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ N\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PM$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ Z\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PM$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ Z\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PM$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ P\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PS$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ P\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \boldsymbol{THEN}\ TrailerOrientation\ is\ PM$$

The control surface shown in the bottom right of Figure 27 summarises the input and output behaviour of the membership functions and linguistic rules. It can be seen in the fuzzy surface that if the vehicle is in the range of $xPositionTrailer\,[41, 45]$ the trailer orientation request is 90° for all $yPositionTrailer$. This flat area will ensure the trailers orientation is 90°, however, it will not allow for the vehicle to align itself with a specific $xPositionTrailer$ value, rather a range of values. Another problem with the RFPP is the large flat area after the trailer has entered the parking space, this prevents the vehicle from achieving the desired final value $xPositionTrailer = 42.5m$. Figure 28 and Table 7 shows the final RFPP design.

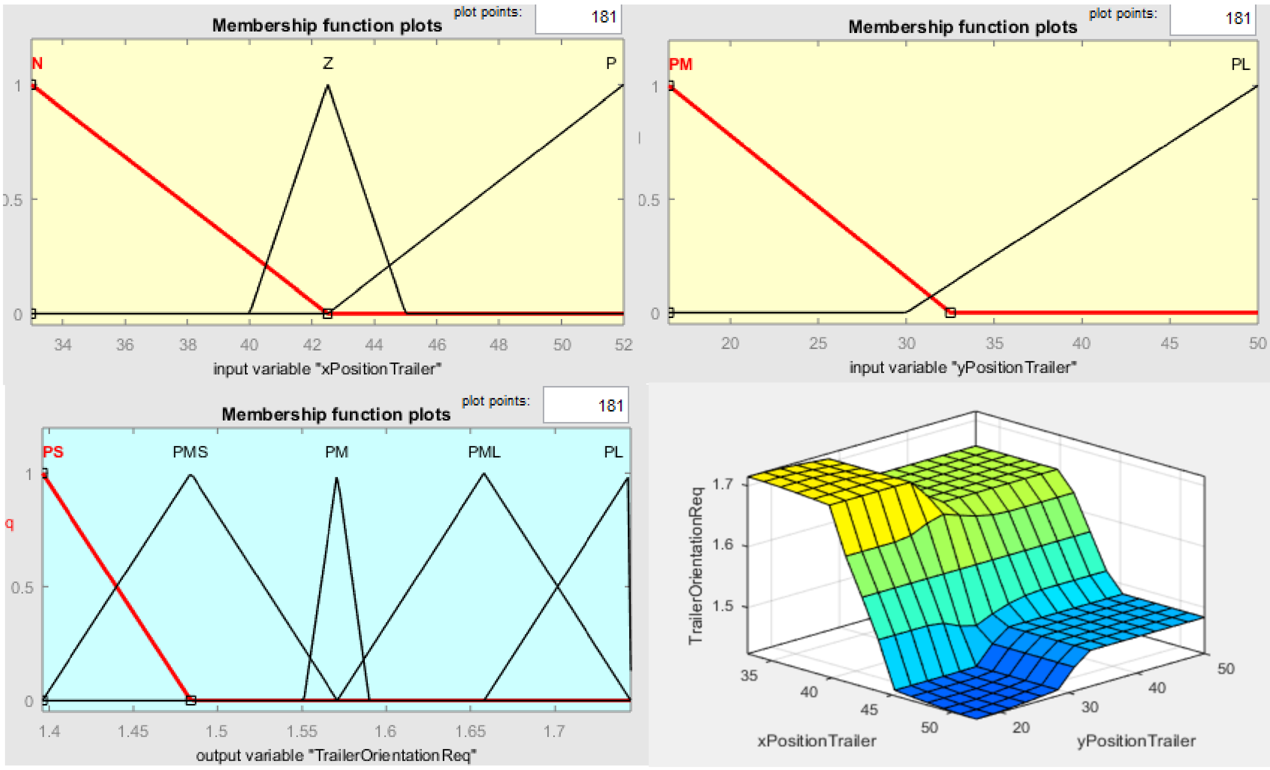


Figure 28, the final RFPP input and output membership functions, and the controller surface. Meters and Radians used

Table 7, the linguistic rules of the RFPP

$$\boldsymbol{IF}\ xPositionTrailer\ is\ N\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PL$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ N\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PML$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ Z\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PM$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ Z\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PM$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ P\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PM\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PS$$
$$\boldsymbol{IF}\ xPositionTrailer\ is\ P\ \ \boldsymbol{AND}\ \ yPositionTrailer\ is\ PL\ \ \boldsymbol{THEN}\ \ TrailerOrientation\ is\ PMS$$

From the fuzzy surface seen in Figure 28 it can be seen that the previously flat $xPositionTrailer$ $[41, 45]$ zone has now been replaced with a slope. This slope allows for the desired $xPositionTrailer = 42.5m$ to be achieved by requesting a trailer orientation that always brings the vehicle back to this desired position. This change was made by allowing for the $xPositionTrailer$ membership functions to cross over the Z membership function. The overlap of the $yPositionTrailer$ membership functions was to allow for a smooth transition when the vehicle's y position increased, this can be seen in the surface at $yPositionTrailer$ $[30, 33]$. Finally, two additional rules for the output trailer orientation membership functions were added, this was to allow for a trailer orientation angle closer to the end of the dock to be smaller. This stops the RFPP from requesting large changes in trailer orientation close to the edge of the parking space, if large orientations were requested the vehicle could destabilise itself. In addition to this, if the vehicle reverses inside the desired parking space its orientation should already be close to the desired 90°, if the vehicle doesn't fit into the space or goes over the parking space lines it will drive forwards.

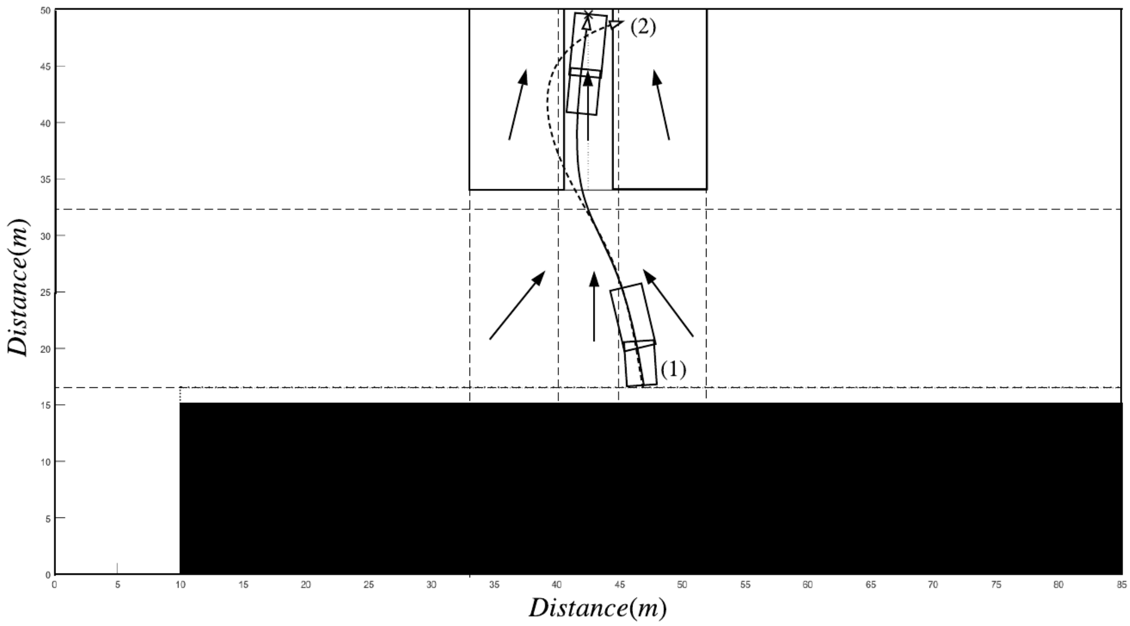


Figure 29, the trajectory of the final RFPP is the solid line, the dotted line is the previous iteration.

In Figure 29 the dotted trajectory shows the first iteration of the RFPP and the solid curve shows the final iteration, the vehicle is shown in its initial and final position when following the optimised RFPP trajectory. The final position of the vehicle is outside of the acceptable truck and trailer orientation, the RFPP gives the correct orientations so the fault must be in either RFC1 or RFC2. The arrows show the desired trailer orientation in the spaces surrounding the parking spot, the two areas on either side of the parking space now have slight trailer orientation requests.

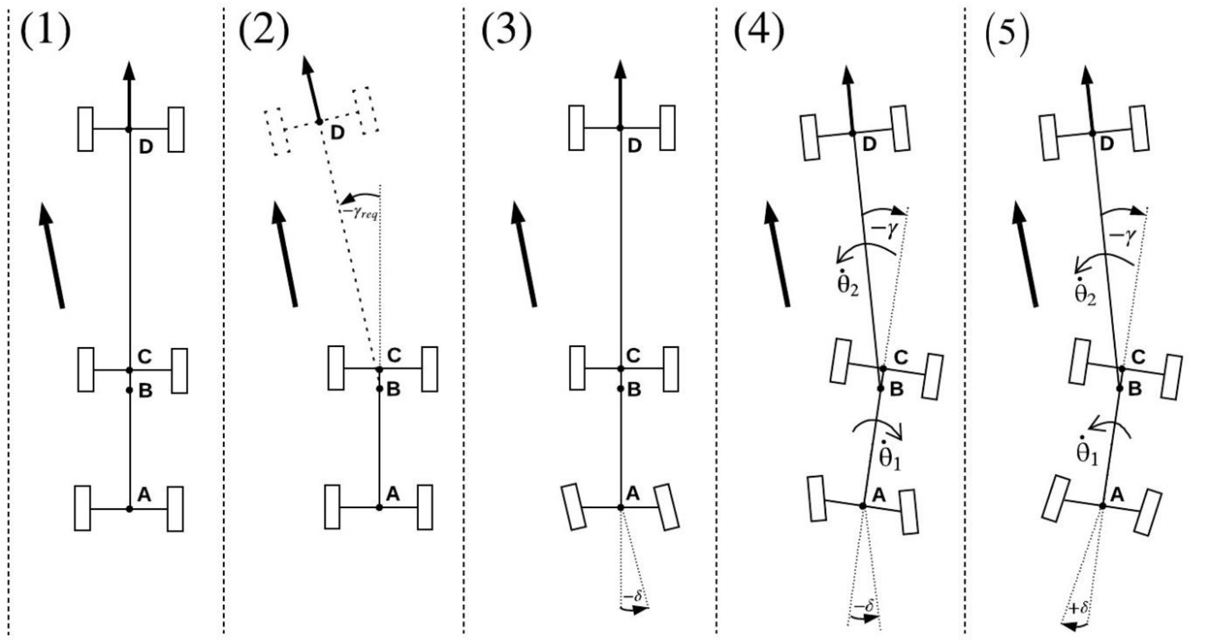


Figure 30, (1) to (5) shows how RFC1 and RFC2 work together to guide the point D on the trailer to the desired reference line.

The RFC1 and RFC2s operation has been described in Figure 30. The vehicle begins at (1) with its trailer orientation parallel to the desired reference line (the dotted line on the left). The trailer has a requested trailer orientation shown by the thicker arrow not attached to the vehicle, the smaller arrow attached to D shows the trailer's current velocity vector. At (1) the vehicle has a positive trailer orientation error, (2) shows the vehicle having the correct trailer orientation, this can be achieved by first reaching the desired hitch angle $-\gamma_{req}$. In order to achieve this requested hitch angle the vehicle is required to first have a negative steering angle (3). This rotates the truck clockwise about B and the trailer anticlockwise about B, creating the desired hitch angle (4). This negative steering angle will grow the hitch angle past the requested hitch angle making the error negative, when the hitch angle error is negative the steering angle should be positive to prevent jack-knifing and to direct the vehicle to the reference path. (5) Shows the vehicle having a negative hitch angle error, consequently the steering angle becomes positive. This process is how the controllers are able to guide the trailer into the parking space and prevent jack-knifing.

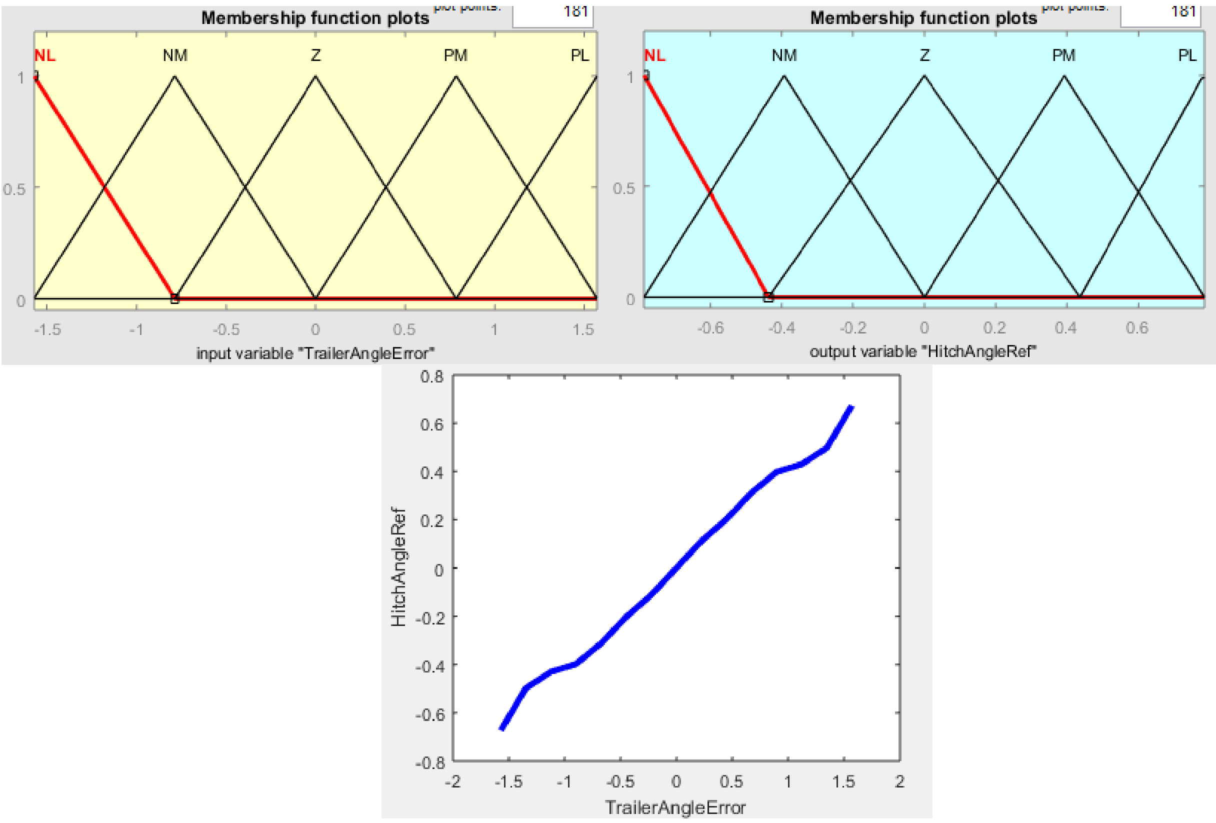


Figure 31, the membership functions of the RFC1 controller responsible for the requested hitch angle. Radians used.

Figure 31 shows the inputs and outputs of the RFC1, the fuzzy surface can be seen at the bottom. When the trailer orientation error is negative the requested hitch angle is also negative, vice vera for positive trailer orientations. This relationship can be summerised in the linguistic rules Table 8. This corrasponds to steps (1) to (2) in Figure 30.

Table 8, the linguistic rules for RFC1.

$$\boldsymbol{IF}\ TrailerOrientationError\ is\ NL\ \boldsymbol{THEN}\ HitchAngleRequest\ is\ NL$$
$$\boldsymbol{IF}\ TrailerOrientationError\ is\ NM\ \boldsymbol{THEN}\ HitchAngleRequest\ is\ NM$$
$$\boldsymbol{IF}\ TrailerOrientationError\ is\ Z\ \boldsymbol{THEN}\ HitchAngleRequest\ is\ Z$$
$$\boldsymbol{IF}\ TrailerOrientationError\ is\ PM\ \boldsymbol{THEN}\ HitchAngleRequest\ is\ PM$$
$$\boldsymbol{IF}\ TrailerOrientationError\ is\ PL\ \boldsymbol{THEN}\ HitchAngleRequest\ is\ PL$$

The RFC2 in used to input a steering angle into the plant to achieved the correct trailer orientation, trailer position, and to manipulating the hitch angle without jack-knifing. The fuzzy surface of this controller refelects the works done on stability in section 3.4, more specifically Figure 7. However this figure was derived with a very strict stability region, for this reason the fuzzy surface (see in Figure 32) of RFC2 isnt exactly the same shape, instead roughtly the same curved shape. Figure 32 shows the membership functions and fuzzy surface for the final RFC2, they control the process (3) to (5) in Figure 30.

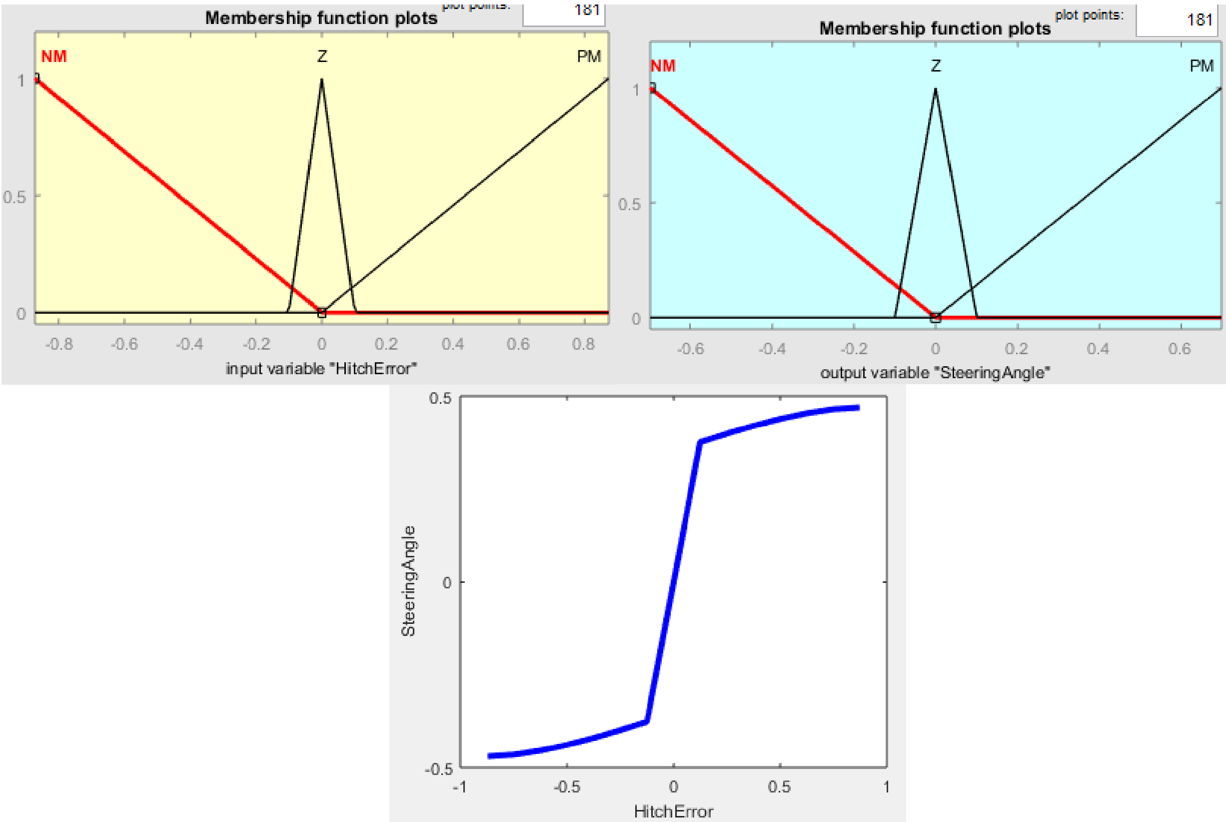


Figure 32, the membership functions for the input and output, as well as the fuzzy surface. Figures in Radians.

It's worth noticing that the fuzzy surface seen in the figure above is very similar to Figure 7 but rotated anticlockwise, this is because the axes have been swapped. It was found that increasing the gradient of the middle line on the fuzzy surface improved the controller's behaviour. By increasing the gradient the overshoot was reduced by 24% and the setteling time was reduced from around 60 seconds to 23 seconds, this was for a steering angle step response test.

The fuzzy surface in Figure 32 has been limited to the range discussed in Figure 7, the steering angle output universe is in the range $-30° \leq \delta \leq 30°$ and the resulting hitch angle error universe is $-49° \leq$

$\gamma_{error} \leq 49°$. It's important to note that the hitch angle never exceeds $\pm 46°$ as discussed in seciton 4.5, while the hitch angle error is permitted to exceed this value as the hitch angle error can be larger than the maximum hitch angle without the hitch angle exceeding its limits. The linguistic rules that produce this fuzzy surface from the memberhsip functions can be seen in Table 9.

Table 9, the table showing the linguistic rules for the RFC2.

$$\boldsymbol{IF}\ \ HitchAngleError\ is\ NM\ \ \boldsymbol{THEN}\ \ SteeringAngle\ is\ NM$$
$$\boldsymbol{IF}\ \ HitchAngleError\ is\ Z\ \ \boldsymbol{THEN}\ \ SteeringAngle\ is\ Z$$
$$\boldsymbol{IF}\ \ HitchAngleError\ is\ PM\ \ \boldsymbol{THEN}\ \ SteeringAngle\ is\ PM$$

In summary:

The design process of the fuzzy path planners and fuzzy controller has been outlined, including how both systems work. In the forward fuzzy control system the FFPP uses 12 rules, the FFC uses 5 rules. In the reverse fuzzy control system the RFPP uses 6 rules, the RFC1 uses 5 rules, and the RFC2 uses 3 rules. The total number of rules used is **31**.

5. Simulation Design

5.1. Introduction

This section will look at how the above analysis and controller design was implemented into Simulink. This will include how the kinematic model was implemented. It will include how the controller switch works, how to prevent jack-knifing if the vehicle exceeds the critical hitch angle. The velocity simulation used to help model loaded and unloaded conditions will be addressed, as well as analysis of vehicle acceleration and deceleration. Finally, the automated method to count the number of manoeuvres will be mentioned. Figure 33 shows the Simulink model built to model the autonomous parking of the T&T vehicle.

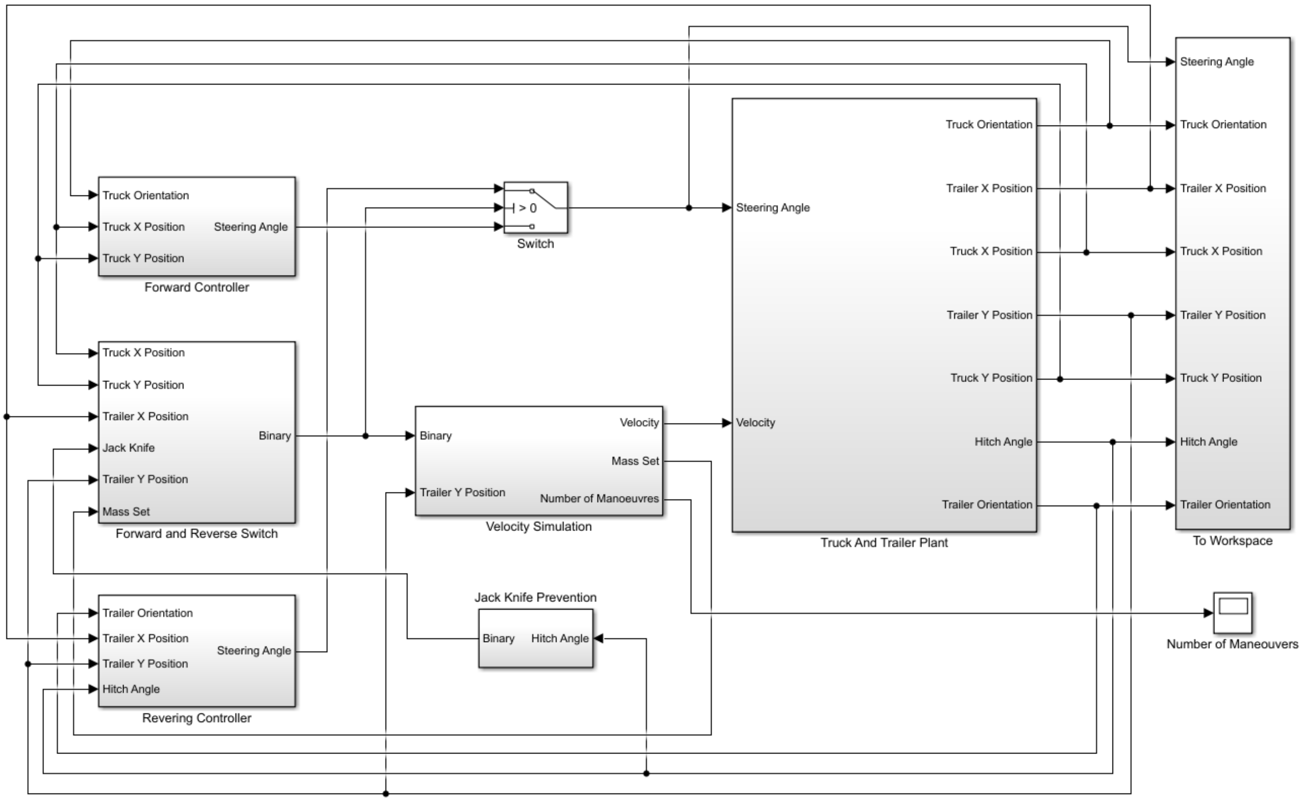


Figure 33, the Simulink model built to simulate the autonomous parking to the T&T vehicle.

5.2. Plant Model

The equations from Table 2 need to be input into Simulink. There was no need to linearize the equations as fuzzy controllers can handle both linear and nonlinear, this allows for a more accurate plant model. Due to not being linearized the equations are not put into matrix form and then solved, instead the equations can be thought of as a state flow process. This process can be seen in Figure 34.

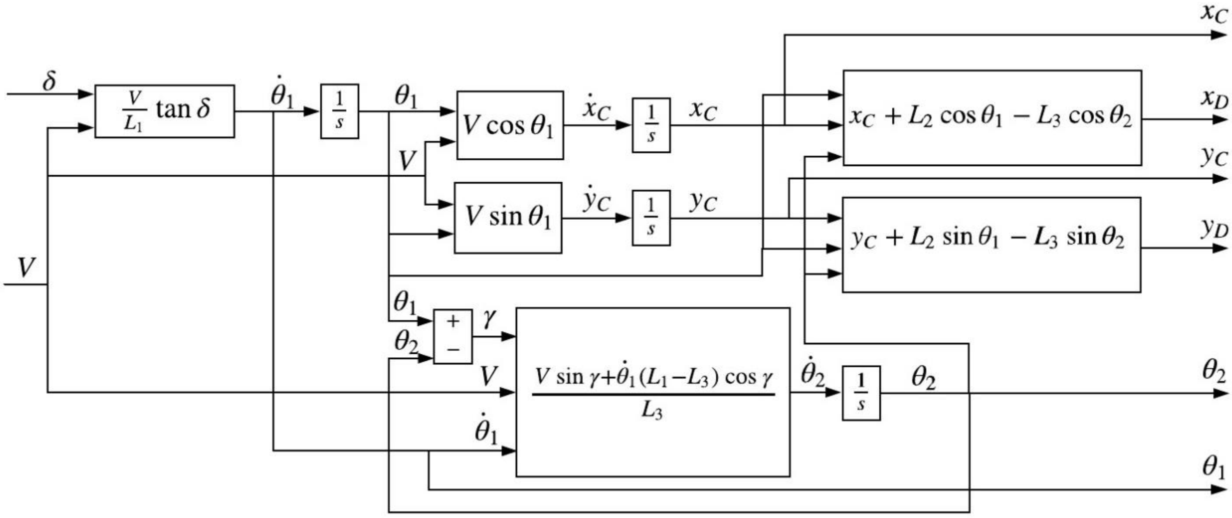


Figure 34, the kinematic model in Simulink. Please refer to Table 2 for equations. L_1, L_2 *and* L_3 *are all constant.*

The above figure shows the input and output states: $input$: δ, V $Output$: $x_c, x_D, y_c, y_D, \theta_1, \theta_2$. Integrators are used to find the truck and trailer orientations as well as the x, y positions of the vehicle at points D and C. The process works by the independent variable flowing into the equation boxes, the dependent variable is then computed and manipulated in the following processes.

Validation of this method was carried out by running various simulations with constant steering angles. The expected behaviour of the vehicle is to drive in a circle, while the trailer should follow in a circular path with a delay. The reason the trailer will follow with a delay is due to the lag in the hitch angle, the vehicle starts of having a hitch angle of 0° and then grows over time to a steady state value that is related to the constant steering angle. The results for this observation can be found in Appendix 5.2.1. As well as general trajectory shape, the states of the vehicle were analysed to ensure that points C and D didn't move unexpectedly. For example, the distance between two points should be constant throughout the circular trajectory (apart from at the very beginning).

5.3. Inertia Simulation

In order to understand how the vehicle decides to swap from a forwards to backwards manoeuvre the acceleration of the vehicle needs to be discussed. The acceleration of the vehicle will depend on the mass of the vehicle (unloaded or loaded), it has been assumed that the vehicle has a constant maximum force it can generate from the engine. For simplicity the vehicle's acceleration and deceleration will be the same. In reality the deceleration of the vehicle would be much larger than the acceleration, therefore, this assumption is a worst-case scenario.

From section 3.3 it was shown that the response of the vehicle to a step input velocity was a first order transfer function. It is realistic to assign maximum accelerations the vehicle can achieve for loaded and unloaded conditions. The gain and time constant for these varying masses can then be found (see Equation 11). From this analysis the vehicle will have a decreasing acceleration for an increasing mass, this relationship between acceleration and mass is related to Equation 9. Different maximum accelerations were chosen for a variety of masses. These chosen accelerations can be seen in Table 10, the masses vary between 16 tonnes and 40 tonnes. The accelerations will be compared against a journal reporting accelerations for light trucks (class 5) to heavy trucks (class 8 & 9) for validation of these selected accelerations.

Table 10, the mass sets and maximum accelerations of the vehicle.

Vehicle Mass Sets (tonnes)	**Maximum Acceleration and Deceleration (ms^{-2})**
$[16, 20.8)$	± 5
$[20.8, 25.6)$	± 4.25
$[25.6, 30.4)$	± 3.5
$[30.4, 35.2)$	± 2.75
$[35.4, 40]$	± 2

To achieve these accelerations a value for both variables in Equation 11, K, τ needs to be found. The transfer function has a maximum input of $\pm 4ms^{-1}$ $\left(\frac{4}{s}\right)$, this occurs when the vehicle swaps from a forwards input signal of $2ms^{-1}$ to a reversing input signal of $-2ms^{-1}$ (or vice versa). For this situation, the largest input signal will produce the largest acceleration. A block diagram of the problem can be seen in Figure 35.

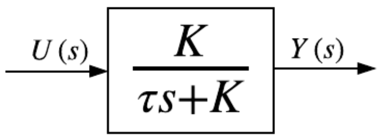


Figure 35, the block diagram showing the input velocity U(s), the transfer function (from Equation 11) and the velocity response Y(s).

From the above figure the relationship of $Y(s)$ can be determined. We already know that the largest input signal is $\pm 4ms^{-1}$ this translates to a S domain value of $U(s) = \frac{4}{s}$. The following analysis to find velocity function $y(t)$ can be seen below.

$$Y(s) = U(s)G(s) = \frac{4}{s}\left(\frac{K}{s+K}\right)$$

$$L^{-1}\{Y(s)\} \rightarrow y(t)$$

$$y(t) = 4 - 4e^{-kt}$$

Equation 17

Equation 17 describes the velocity profile of the vehicle as it accelerates from either $2ms^{-1}$ to $-2ms^{-1}$ or vice versa. As the function is a velocity it can be differentiated with respect to time to find the acceleration. This relationship for acceleration can then be put equal to the maximum acceleration defined in Table 10 and solved for the values of k. The value of k can then be substituted into $G(s)$ to find the value of $K \; and \; \tau$, this is seen in Equation 18. The process can be repeated for all the masses, this is seen in Table 11, Appendix 5.3.1 shows how to get the time constant.

$$\frac{dy}{dt} = 4ke^{-kt} = 5 \;\therefore\; k = 5/4$$

$$G(s) = \frac{5/4}{s+5/4} = \frac{5}{4s+5} \therefore \tau(t) = 0.88s \text{ and } K = 5 \qquad \textit{Equation 18}$$

Table 11, the mass sets and the corresponding time constant and gain.

Vehicle Mass Sets (tonnes)	**Maximum acceleration (ms^{-2})**	**Time constant (s)**	**Gain (k)**
$[16, 20.8)$	± 5	0.88	$5/4$
$[20.8, 25.6)$	± 4.25	1.03	$17/16$
$[25.6, 30.4)$	± 3.5	1.26	$7/8$
$[30.4, 35.2)$	± 2.75	1.6	$11/16$
$[35.4, 40]$	± 2	2.2	$1/2$

From (Yang et al., 2016) it is seen that for a light weight truck and trailer (class 5 ≈ 16 tonnes) the average acceleration over 6.1m (20 feet) was found to be $1.5ms^{-2}$, the heavy vehicles value was $0.65ms^{-2}$. It was found that by using the time constants and gains above the lightest vehicle would also have an average acceleration over this distance of $1.03ms^{-2}$, the heavy vehicle's average was $0.51ms^{-2}$, verifying the above maximum accelerations chosen. For details on how this average velocity was found please see Appendix 5.3.2.

Based on the selected vehicle mass, one of the above mass sets will be chosen with the appropriate transfer function that determines the maximum acceleration. This velocity profile generated from the transfer function is then input into the plant. This is done in Simulink by having the selected mass and velocity reference input into a Matlab function (1), this then determines which transfer function to input the velocity reference into (2). The velocity response generated from this transfer function is then input into another function (3) that takes only non-null values of velocity and then inputs them into the plant. This can be seen in Figure 36. Inside the Transfer Function (between (2) and (3)) can be seen in Appendix 6.3.3.

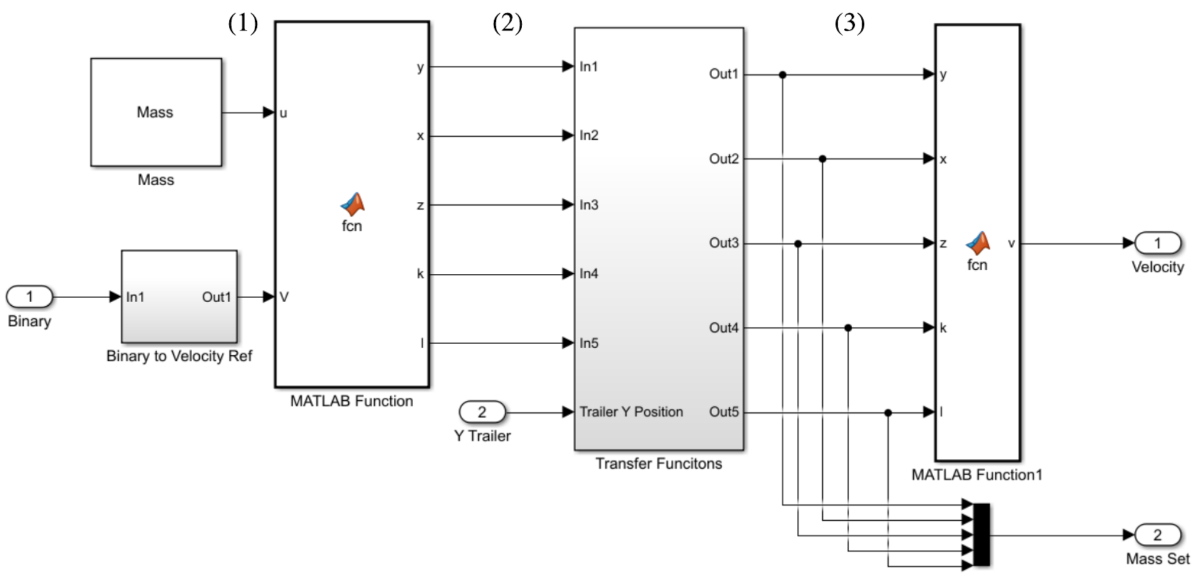


Figure 36, the Simulink model to generate a velocity response for different masses.

5.4. Switch System

The workings of the switching system are fairly simple in principle, when the vehicles automated process is engaged by the driver the vehicle will always drive forwards first. The FFPP then guides the vehicle to the pedestrian walkway and the vehicle will start to decelerate and then stop just before it goes onto the walkway. Just before the vehicle reaches the walkway the forward controller disengages, and the reversing controller engages, this occurs at a stopping distance $s(m)$. The reversing controller then reverses the vehicle into the parking space, if the vehicle doesn't fit into the parking space the forwards controller will be engaged and the vehicle will brake and then accelerate forwards to the walk way. This process is carried out until the vehicle fits inside the parking space. The reversing controller will engage the brakes when the vehicle is close to the desired docking position.

The difficulty in simulating this behaviour is due to the controller's uses of different accelerations and decelerations for different masses, and the process of switching between the forwards and reversing controller. The varying acceleration for different masses is difficult because for each acceleration the vehicle has to start to brake at different distances from the walkway and parking space, furthermore, when the vehicle reverses into the parking space at some point the system has to decide if the vehicle will fit into the parking space before it actually gets there. Braking once the vehicle doesn't fit inside the parking space will mean the vehicle will drive into the other spaces before slowing down. For this reason, a predictive method had to be developed to ensure the vehicle never goes into the pedestrian walkway and never goes over the parking spaces boundaries. The code and model can be found in Appendix 5.4.1.

To do this the vehicle must have a pre-emptive system that calculates the distance required to stop the vehicle in time before it reaches any boundaries. As the vehicle's initial velocity before acceleration/decelerating is known, as well as the maximum acceleration/deceleration, then calculations can be carried out to predict the stopping distance. This stopping distance is more complicated for reversing into the parking space as a prediction of the controller's actions needs to be made. This prediction is analysed when the vehicle gets within the stopping distance of the parking space, the vehicle's angle and position at this point is then used to determine if the vehicle can fit inside the space or if it needs to drive forwards correcting its orientation and position for another reversing attempt. Equation 19 was used to predict the stopping distance (s) required for vehicle of mass set $[16, 20.8)$ to stop from a forward manoeuvre. The following derivation was used to determine the stopping distances for all the mass sets. Equation 19 comes from analysis seen in Figure 35 but for an input signal of $-4ms^{-1}$ $\left(\frac{-4}{s}\right)$. The stopping distances vary from 0.49m to 1.5m. The stopping distance when the vehicle approaches the walkway has a safety factor of 1.5 ensure the vehicle always stops before this area. All the other stopping distance have a safety factor of 1.1, as there is less risk to human life in these manoeuvres.

$$v(t) = -2 + 4e^{-\left(\frac{5}{4}\right)t} \quad \text{Equation 19}$$

$$find\ t\ when\ v = 0{:}\quad 0 = -2 + 4e^{-\left(\frac{5}{4}\right)t} \ \therefore t = 0.55s$$

$$find\ expression\ for\ distance{:}\quad x(t) = \int v(t)dt = -2t - \frac{16}{5}e^{-\left(\frac{5}{4}\right)t} + c$$

$$x(0) = 0 \ \therefore c = \frac{16}{5}$$

$$find\ x(0.55){:}\ x(0.55) = \frac{16}{5} - 2(0) - \frac{16}{5}e^{0} = \mathbf{0.49m}$$

These stopping distances need to be adjusted again because the vehicle isn't always perpendicular to the boundary. The manoeuvre when the vehicle reverses into the parking space also uses this stopping distances. The vehicle decides to brake and go forwards if the vehicle doesn't fit into a smaller defined area of the parking space at the point within the stopping distance. This is explained in Figure 37.

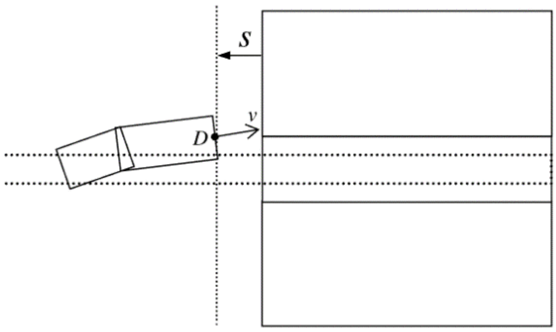


Figure 37, the vehicle at the decision point needs to decide whether it should drive forwards or continue reversing.

The above figure shows the vehicle attempting to reverse into the parking space, the vehicle gets to a distance that is equal to the stopping distance s and has to decide whether to continue or drive forwards and try again. For this situation the point D is outside of the dotted lines that extending from inside the parking space, therefore the vehicle will drive forwards. If D is inside these lines, then the vehicle will continue reversing. The dotted lines are exactly half the width of the trailer away from the parking space's side edges.

The system decides when to switch between forwards and backwards manoeuvres at these s distances away from boundaries. Once the system has made a choice a binary output is generated, zero for forwards and one for reversing. This binary output is then combined with the reversing steering angle from RFC2, or the forwards steering angle from FFC using a Simulink switch. The Simulink switch works by changing to the forwards steering angle if the binary output is zero, and swapping to the reversing steering angle if the binary output is one. This switch then outputs the selected steering angle into the plant, this can be seen in Figure 33.

5.5. Jack-Knife Prevention

As discussed in section 4.5, the jack-knife phenomenon occurs in the Ford T&T when the hitch angle exceeds $\pm 46°$. If this condition occurs the vehicle will automatically switch to the forwards control system. This requires the jack-knife signal in Figure 33 to override any other controller switch actions. As well as this override capability the jack-knife prevention system is required to stop the vehicle if it ever exceeds a critical value, when the corners of the trailer hit the back of the truck due to an excessive hitch angle. However, not enough information about the truck's passenger cabin was provided to determine the angel, an arbitrary hitch angle of 60° was chosen. This can be updated easily by considering the vehicle geometry at a later point. Figure 38 shows the jack-knife prevention system. "Ultimate JK prevention" is used to stop the vehicle when the hitch angle exceeds 60°, and the bottom Matlab function is used to send a signal that will override the switches functionality and make the vehicle drive forwards until the hitch angle is $< 30°$ (arbitrary number). When the vehicle drives forwards the hitch angle is automatically reduced.

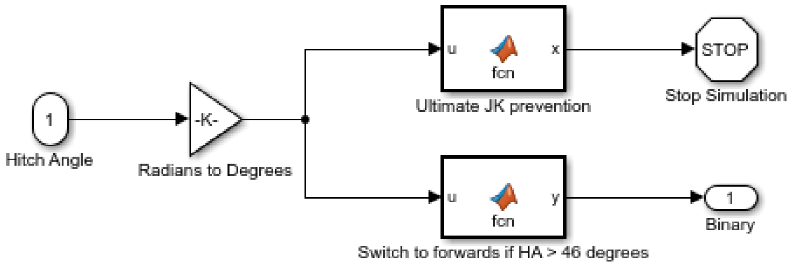


Figure 38, the jack-knife prevention system.

5.6. Number of Manoeuvres

One of the key performance indicators of this project is the number of manoeuvres the vehicle has to undertake in order to successfully park. A manoeuvre is classified in this context as either forwards or reversing. For example, if the vehicle drives forwards from the parking space entrance, reaches the walkway, and then reverses into the parking space with the desired orientation and position, then only two manoeuvres were carried out. Consider Figure 39.

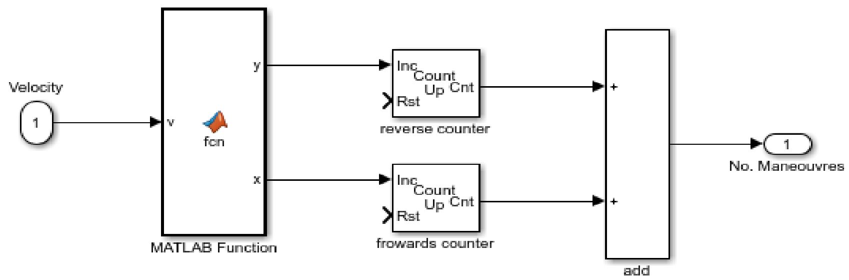


Figure 39, the system that prevents jack-knifing.

The above figure takes the velocity of the vehicle into the Matlab function. This function turns the velocity of the vehicle into a binary output, either zero for a positive velocity or one for a negative velocity. The y output of the matlab function is the same binary signal from the switch process, the x output is the inverse of this signal. The two signals then pass into counters that count the number of times the x, y signals reach one, the two output are then summed together to give the total number of manoeuvres.

6. Results and Discussion

The path planner and control system has been designed, this section will review the simulations run to compare the system to the Key Performance Indicators (KPIs), seen in Table 1. The section will look at initial positions that a sensible driver would stop the vehicle in before engaging the automated parking system. A sensible initial condition would involve small hitch angles, should not be too close to the sides of the entrance, and with truck and trailer orientations of around $-90°$. After this randomly selected initial conditions will be tested. To better analysis the controllers developed, both the forwards and reversing controllers will be analysed using different reference inputs. These reference inputs replicate some of the demands that could be made by the fuzzy path planner. This sections will then find the limits of the designed system and test for robustness.

6.1. Time Step Convergence

The objective of this section is to choose a time step that shows convergent behaviour for critical KPIs. The time step that achieves the convergent accuracy and has the fastest computational time will be used. Before the tests can be carried out a convergent time step should be found. The convergent time step is the time step that allows the simulation to run without the vehicle exceeding the boundaries of the space and also shows that it has converged to critical KPIs. The KPIs considered are seen in the legend of Figure 40. The test to find the convergent step time was carried out with the same initial conditions

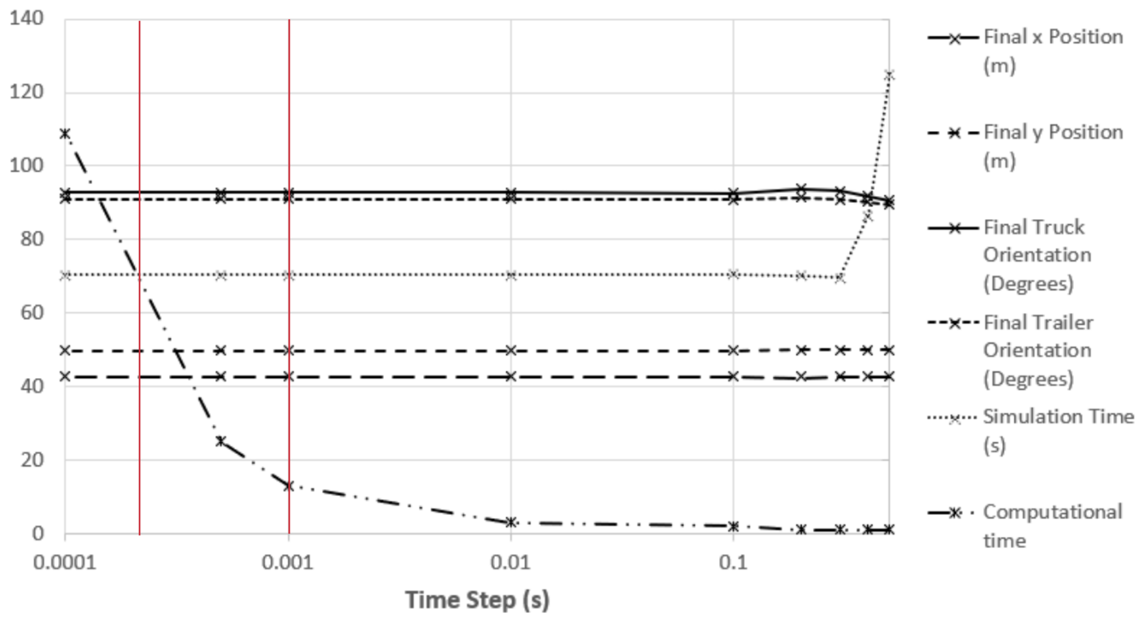


Figure 40, the convergent behaviour of the time step. X-axis is Log base 10.

$(x_C = 5m, y_C = 9.75m, \theta_1 = -90°, \theta_2 = -90°, m = 28\ tonnes)$. These values were chosen as they are the initial conditions that place the vehicle in the centre of the entrance with the mean vehicle mass.

Figure 40 shows that by changing the time step the final position and orientation of the vehicle change. It can be seen that as the time step is reduced the computational time increases, the computational time can't exceed the simulation time ($70.31s$) otherwise the system will experience lag, this correlates to a minimum time step of around $0.0003s$. From Figure 40 it can be seen that the KPIs converge at the $0.1s$

time step, however, this time step can't be used as the vehicle drives $21cm$ into the pedestrian walkway. Only when the time step is less than or equal to $0.001s$ does the vehicle stop before the parking space boundaries. This behaviour is expected, the stopping distance was calculated using continuous methods, Simulink solves discreetly. This causes the vehicle to overshoot onto the walkway. To ensure this doesn't happen a time step between the two red lines should be selected. However, the best choice is the largest time step, this is because this time step has the smallest computational time. This computational time (12s) was found using an Intel i7 3.7GHz CPU, normal microcontrollers usually only have around 1.4GHz (Industries, 2017), meaning that the computational time would be much longer for these processors. Assuming a linear relationship between CPU speed and computational time the microcontroller's solving time would be 32s. For these reasons mentioned above the time step used will be $0.001s$.

If the selected microcontroller experiences lag with a time step of $0.001s$ then the stopping distance could have an additional safety factor added onto it. The problem with this is that the relationship between how close the vehicle gets to the walkway and the number of manoeuvres required to park is very sensitive to change.

6.2. Error Analysis

This section will explain how error analysis will be carried out for the final position and orientation of the vehicle. The maximum error is the largest possible error in position and orientation the vehicle can have and still be considered an acceptable parking manoeuvre. For the following data to be meaningfully compared to the maximum allowable error), and errors found in the literature, an error analysis formula must be constructed. As the focus in the literature has been on the final vehicle position and orientation, only the error for this final configuration will be used. In (Riid, Lei and Rüstern, 2009) the authors make use of the very rear point of the trailer and its orientation, there was no consideration to the orientation of the truck. Equation 20 shows their equation.

$$\epsilon = \epsilon_p - 0.0267\varepsilon_\theta$$
$$\epsilon_p = \sqrt{{x_{De}}^2 + {y_{De}}^2}\ and\ \varepsilon_\theta = |\theta_{2e}| \qquad \text{Equation 20}$$

This equation has a scaling factor 0.0267 added to the orientations to give an even weighting to both position and orientation. From (Riid, Lei and Rüstern, 2009) it was found that for a sample size of 10 the maximum error was $\epsilon_{max} = 0.065$, the minimum error was $\epsilon_{min} = 0.013$, the mean error of $\epsilon_{mean} = 0.037$, and the standard deviation $\sigma = 0.015$.

As well as comparing the results to the literature an expression for the final vehicle position and orientation should also be found. This expression should include the truck's orientation as this is a KPI, the expression can be seen in Equation 21.

$$\epsilon = 10\sqrt{10}\epsilon_p - \varepsilon_\theta$$

$$\epsilon_p = \sqrt{{x_{De}}^2 + {y_{De}}^2} \; and \; \epsilon_\theta = \sqrt{6.25{\theta_{2e}}^2 + {\theta_{1e}}^2} \quad \text{Equation 21}$$

To understand the values $10\sqrt{10}$ and 6.25, consider the maximum acceptable error. The acceptable range of the vehicle's position is defined by: $49.95m \leq y_D \leq 49.75m$, and $42.3m \leq x_D \leq 42.7m$. The acceptable range of vehicle's orientation are defined by: $85° \leq \theta_1 \leq 95°$, $88°m \leq \theta_2 \leq 92°$. Therefore, the maximum errors are: $y_{emax} = 0.1m$, $x_{emax} = 0.2m$, $\theta_{1emax} = 5°$, $y_{emax} = 2°$. This gives a maximum error of $\epsilon = 14.14$. The values $10\sqrt{10}$ and 6.25 are used to give equal weighting to position and orientation in the error equation when considering the maximum error.

6.3. Basic Simulations

This section will look at how the final position and orientation of the vehicle is affected by simple initial conditions. Simple initial conditions referrer to initial conditions that only vary in x position and the mass of the entire vehicle (m). The basic simulations analysed will look at the truck initial x position varying along the entrance of the parking space. The limits of these initial conditions are selected from the areas where the truck won't scrap against the entrance side boundaries $1.3m \leq x_C \leq 8.7m$, the limit the vehicle can drive in is $1.275m \leq x_C \leq 8.725m$. The initial y positions will stay at a constant value of $y_C = 4.5m$ meaning the entire vehicle is inside the parking space before the automated parking system is engaged. Simulations to test the behaviour of the varying mass of the vehicle will also be tested. These tests will be carried out at $x_C = 5m, x_C = 1.3m \; and \; x_C = 8.7m$ with the largest and smallest mass sets tested for these initial x positions. Table 12 and Table 13 details all the simple simulations to run.

Table 12, the basic simulations to run. Only the x position varies.

Test Number	$x_C\ (m)$	$y_C\ (m)$	$\theta_1\ (°)$	$\theta_2\ (°)$	$m\ (tonnes)$
1	1.3	4.5	-90	-90	28
2	2	4.5	-90	-90	28
3	3	4.5	-90	-90	28
4	4	4.5	-90	-90	28
5	5	4.5	-90	-90	28
6	6	4.5	-90	-90	28
7	7	4.5	-90	-90	28
8	8	4.5	-90	-90	28
9	8.7	4.5	-90	-90	28

Table 13, the basic simulations for varying x position and the mass.

Test Number	$x_C\ (m)$	$y_C\ (m)$	$\theta_1\ (°)$	$\theta_2\ (°)$	$m\ (tonnes)$
10	1.3	4.5	-90	-90	16
11	1.3	4.5	-90	-90	40
12	5	4.5	-90	-90	16
13	5	4.5	-90	-90	40
14	8.7	4.5	-90	-90	16
15	8.7	4.5	-90	-90	40

The reason for only changing some of the variable initial conditions is to limit the total number of test carried out. If all initial conditions were varied 9 times (like for x in Table 13) then just under 60,000 simulations would need to be run. From testing experience, it has been found that the sensitive parameters are the x position and the vehicle's mass. The x position impacts the path chosen more than other initial conditions and the mass impacts the total time and number of manoeuvres required. Table 14 and Table 15 shows the test results of all 16 test in relation to the KPIs, all of the test successfully parked within the allowed errors, without crossing any boundaries, exceeding the manoeuvre limit, or the time limit.

Table 14, the test results for four of the KPIs.

Test Number	Final Trailer Orientation Error (°)	Final Truck Orientation Error (°)	Final x Position Error (cm)	Final y Position Error (cm)
1	-0.96	-2.91	-15.8	0
2	-1	-3	-14.7	0
3	-1	-3.02	-14.6	0
4	-0.96	-2.93	-15.7	0
5	-0.96	-2.93	-15.7	0
6	-0.96	-2.93	-15.7	0
7	-0.20	-0.85	-5.8	0
8	-0.47	-1.52	-8.5	0
9	0.64	0.43	-7.6	0
10	-0.48	-1.8	-15.4	0
11	-1.07	-3.05	-14.1	0
12	-0.44	-1.67	-15.1	0
13	-1.05	-3.00	-14.6	0
14	-0.67	-2.48	-16.6	0
15	-0.13	-0.90	-7.6	0

Table 15, the test results for the other four KPIs

Test Number	Final Hitch Angle Error (°)	Time Taken (s)	Number of Manoeuvres	Maximum Hitch Angle (°)
1	-1.95	70.5	4	23.6
2	-2.01	70	4	23.8
3	-2.01	69.7	4	23.8
4	-1.96	70.1	4	23.6
5	-1.96	70.1	4	23.6
6	-1.96	70.1	4	23.6
7	-0.64	86.5	6	25
8	-1.05	85.3	6	25.4
9	-0.21	99.5	10	24.9
10	-1.32	67.1	4	20.5
11	-1.98	91.5	6	31.5
12	-1.23	66.9	4	20.4
13	-1.95	91.3	6	31.4
14	-1.81	68.5	4	21.7
15	-0.77	110	8	33.2

Using Equation 20 and Equation 21 error analysis can be carried out comparing the values in Table 14. This will allow for the system designed in this thesis to be compared to the results found in (Riid, Lei and Rüstern, 2009). (Riid, Lei and Rüstern, 2009) is the most similar paper to this thesis as the controller and path planning structure was adapted from the author's ideas. The errors can be seen in Table 16.

Table 16, the error analysis of final position and orientation.

Test Number	Error (using Equation 20)	Error (using Equation 21)
1	0.055	8.77
2	0.057	8.55
3	0.057	8.54
4	0.055	8.75
5	0.055	8.75
6	0.055	8.75
7	0.014	2.82
8	0.028	4.61
9	0.021	4.06
10	0.031	7.03
11	0.059	8.52
12	0.028	6.77
13	0.058	8.60
14	0.043	8.24
15	0.012	3.36
	$\boldsymbol{Mean = 0.042}$	$\boldsymbol{Mean = 7.08}$

The above table shows that the error analysis from the paper and the error analysis derived show similar results with the minimum error and maximum error occurring for the same tests. The test case that shows the highest position and orientation error was found from test 11, with an error of 60% of the maximum allowable error. This result seems reasonable as both orientation errors were the largest out of all the tests. This result is unsurprising as test 11 has two extreme initial conditions; one the vehicle is fully loaded, and the other is the vehicle is on the extreme left of the entrance. To better understand the effects of starting position and mass Table 17 was constructed.

Table 17, the comparison of simulations on the left and right, and the comparison of mass on either side of the mean value.

	L.H.S Average	R.H.S Average	$m < 28\ tonnes$ Average	$m > 28\ tonnes$ Average
Error	8.65	5.06	6.83	7.35
Percentage Increase in Mean Error (7.08)	22%	-29%	-3.5%	3.8%

Table 17 shows the mean error (from Equation 21) given for all the values found on the L.H.S of the entrance, with a mean load of 28 tonnes. It shows the same for the R.H.S. An average of all loaded simulations under 28 $tonnes$, and an average of all the simulations over 28 $tonnes$. It can be seen that when starting on the L.H.S the vehicle's error is 22% larger than the mean. When starting on the R.H.S the vehicles error is 29% smaller than the mean. When the vehicle's mass is less than 28 $tonnes$ the vehicle error is 3.5% smaller than the average. Finally, When the vehicle's mass is larger than 28 $tonnes$ the error is 3.8% larger than the mean. This shows that the effect of varying the initial x position has a large impact on the error of the final vehicle's position and orientation. This analysis backs up the maximum error being found in test 11, as it occurs when the vehicle is on the extreme L.H.S and is maximally loaded.

The above analysis is true for most of the simulations run, however, some anomalies are found. From the above analysis the smallest error should be seen in test 14, which has the largest value on the R.H.S and is unloaded. Instead test 15, which carries the maximum load on the extreme R.H.S shows the lowest error. The reason for this behaviour is due to the number of manoeuvres; test 14 carried out 4 manoeuvres, and test 15 carried out 8. The reason more manoeuvres gives a better result is because the vehicle has more opportunities to align itself with the centre of the parking space. However, just because the vehicle has more opportunities to align itself doesn't necessarily mean that its error will always be smaller, rather it has more chance to be smaller. The reason for this is based on the random position of the second to last reversing manoeuvre. If the second to last reversing manoeuvre doesn't fit inside the parking space by a very small margin, then the vehicle will drive forwards and align itself very accurately to the central parking space. This allows for the last reversing manoeuvre to ensure high accuracy. The other extreme is that the second to last reversing manoeuvre is far away from the desired parking centre line, the vehicle then drives forwards to a better position to start another reversing manoeuvre. This reversing manoeuvre only just fits inside the constraints and parks the vehicle with larger errors. This phenomena adds a slight randomness to the errors seen in Table 16.

How easily the automatic system parked the vehicle is analysed by comparing the number of manoeuvres and the time taken to complete the manoeuvre. It can be seen in Table 15, tests 1-9, that for initial positions on the left the vehicle can park more easily than on the right. Both the number of manoeuvres and the total time taken increases as the vehicle starts from a position close to the R.H.S of the entrance. In general, the vehicles initial position shows improved accuracy on the R.H.S of the entrance but takes a longer time to achieve the final position, whereas the initial positions on the L.H.S reaches the desired position faster but with less accuracy.

The maximum hitch angle in all 15 tests never gets close to the uncontrollable $\pm 46°$, with the average maximum hitch angle being $|25.1|°$. The largest hitch angle always occurs during the vehicle's first reversing manoeuvre, subsequent manoeuvres have a reduced maximum hitch angle, this shows that the forward manoeuvres reduce the hitch angle. The largest hitch angle $|31.5|°$ was found in test 11 during the first reversing manoeuvre. The vehicle would require a hitch angle $14.5°$ larger to ever activate the emergency hitch angle prevention system. It is unlikely any initial position would cause jack-knifing.

It can be seen that the y_D error is always zero. The reason for this is that the vehicle's inertia simulation uses a tuned stopping distance to allow for the vehicle to slow down and then stop at exactly $y_D = 49.85m$ for all loaded and unloaded conditions. The vehicle's velocity just before this position is very small, and when the vehicle reaches $y_D = 49.85m$ the simulation is stopped. This stop at such a small velocity is equivalent to applying the brakes. In reality the vehicle would have sensors attached to the rear of the trailer that would be used by a controller to brake, this would give an error.

From Table 16 the mean error using analysis found in the literature can be calculated. The mean is $\epsilon_{mean} = 0.042$, the maximum was $\epsilon_{max} = 0.059$, the minimum was $\varepsilon_{min} = 0.012$, and standard deviation $\sigma = 0.017$. The mean error for the final position is 17% larger than the error seen in the literature, the literature also shows a standard deviation 13% smaller than the proposed system. However, in this paper the constraints on the vehicle's movement is very relaxed and no boundaries request tight manoeuvres. The paper also has several reversing manoeuvres that are longer than 30m (the maximum distance possible to reverse in Ford's docking bay), this allows the vehicle to achieve a steady state value before reaching the final position reducing the final error. The author's also only use a sample size of 10, however, having such a small standard deviation shows that the mean value is unlikely to change if the sample size was increased. If only the R.H.S simulation are considered the system proposed has similar error results as the paper mentioned. The mean error is now $\epsilon_{mean} = 0.029$, this is 22% smaller than the paper's mean. The standard deviation for the R.H.S is now similar to the paper's value $\sigma = 0.014$, which is 7% smaller.

6.4. Randomised Initial Conditions

This section will address testing of random combinations of initial positions and orientations inside the parking space entrance. After this, specific initial conditions will be considered that might occur outside of the parking space entrance. Table 18 shows the randomly selected initial conditions, Table 19 shows the vehicle's final position and orientation errors, and Table 20 shows the other KPIs. After this a single randomly selected simulations will be used to analysis the trajectory.

Table 18, the randomly selected values for the initial vehicle conditions.

Test Number	$x_C\ (m)$	$y_C\ (m)$	θ_1 (°)	θ_2 (°)	$m\ (tonnes)$
1	4.63	8.18	-105.00	-75.00	35.20
2	6.49	7.65	-81.00	-69.00	35.20
3	7.24	6.08	-63.00	-66.00	37.60
4	5.37	11.33	-72.00	-105.00	23.20
5	2.77	11.85	-108.00	-84.00	17.20

Table 19, the vehicles errors in the final position. Included is the overall error.

Test Number	Final Trailer Orientation Error (°)	Final Truck Orientation Error (°)	Final x Position Error (cm)	Error
1	-1.06	-3.00	-14.8	4.95
2	-1.07	-3.05	-14.0	5.02
3	-1.14	-3.16	-16.3	5.25
4	-1.05	-3.07	-15.00	5.01
5	n/a	n/a	n/a	n/a

Table 20, the vehicles other KPIs.

Test Number	Final Hitch Angle Error (°)	Time Taken (s)	Number of Manoeuvres	Maximum Hitch Angle (°)
1	-1.94	89.9	6	31.4
2	-1.98	89.5	6	31.6
3	-2.03	89.1	6	31.9
4	-2.02	63.8	4	22.5
5	n/a	(120)	21	24.0

The random initial conditions were selected using a random variable selection function found in Excel. It can be seen that out of the 5 tests all but one ran successfully. This test failed due to the vehicle getting stuck in the parking space. The vehicle reversed into the space successfully, but due to its angle on entry it gets close to the sides of the parking space and is required to drive forwards. However, past a certain point in the parking space the vehicle is also requested to reverse. These conflicting system requests cause the vehicle to sway between forwards and backwards manoeuvres rapidly. This could be avoided by implementing a system that drives the vehicle forwards when it exceeds a certain number of manoeuvres in a period of time. This system should override the other two requests and would stop the vehicle from getting stuck in this area.

The randomly selected simulations all have smaller errors than the mean error found from the simple simulations ($\varepsilon_{Simple\ mean} = 7.08$), this is because the randomly selected initial positions were predominately on the R.H.S of the entrance. As the simulations are on the R.H.S of the parking space the mean simulation's time is larger than the mean found for the simple simulations, $t_{simple\ mean} = 79.1s$ and $t_{complex\ mean} = 83.1s$.

6.5. Transient Behaviour

6.5.1 The Trajectory

This section will look at one randomly selected initial position, the controller's performance and the vehicle's transient behaviour will be examined as well as the vehicle's trajectory. Figure 41 shows the trajectory of the truck and trailer in the parking space. Initial conditions: $x_C = 3.51\ m, y_C = 13.95m, \theta_1 = -66°, \theta_2 = -78°, and\ m = 27.2\ tonnes.$

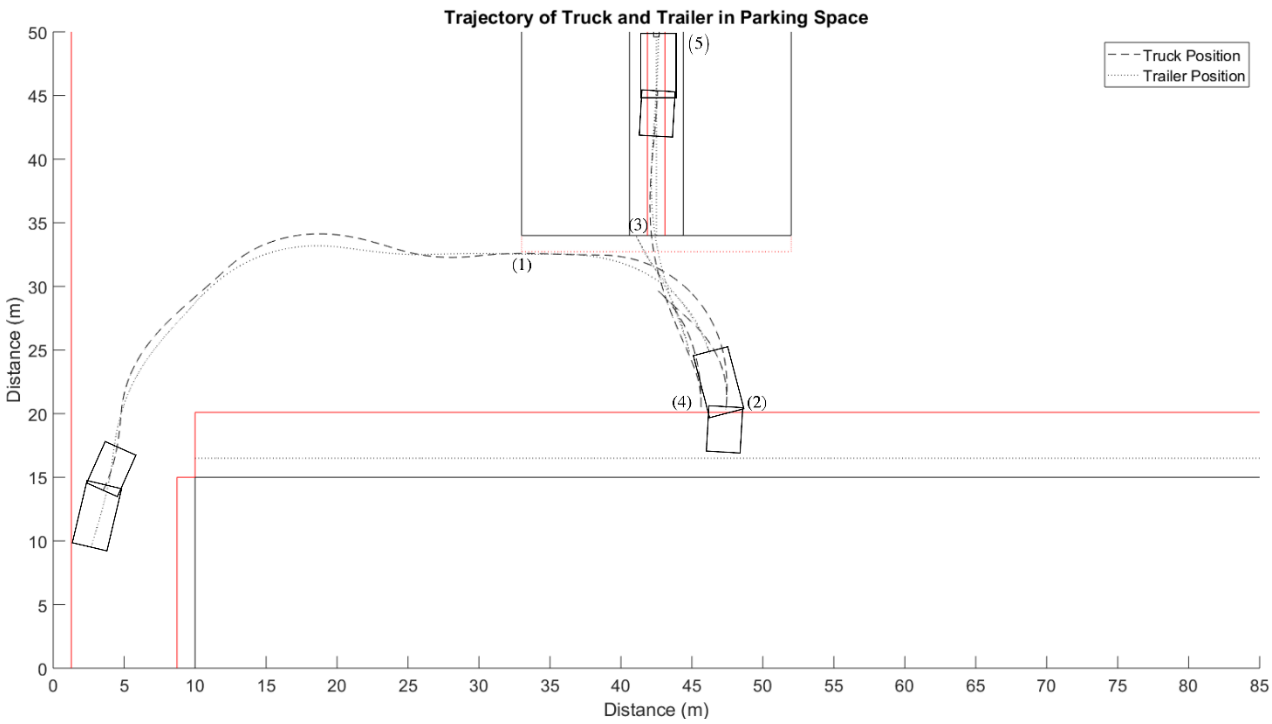


Figure 41, the trajectory from the initial positions above. 4 manoeuvres are carried out.

Figure 41 shows the trajectory of the vehicle form the starting position to the final parked position (5). In this figure the solid red lines show the boundaries that should never be crossed by the vehicle's trajectory lines. The dotted red lines on front of the parking space are used to show how close the vehicle at position (1) gets to driving over other parking spaces, this line can be crossed when trying to reverse into the desired space. It can be seen that the vehicle doesn't cross the boundary at (1). From (1) to (2) the vehicle makes a hard-right turn to try and align itself with $x = 47.5m$, ending up at (2) where the vehicle stops just before the pedestrian walkway. The vehicle then reverses from (2) to (3), just before (3) the system predicts that the vehicle won't fit inside the space, it decides to brake and drive forwards. The vehicle stops at position (3), this line would fit inside the parking space, but the sides of the vehicle would go into the other spaces. From (3) the vehicle drives forwards stopping in front of the walkway (4). The vehicle then reverses from (4), fits inside the parking space with no overlap and stops just before the back of the space at (5).

The trajectory carried out by the system is respectable, allowing for the vehicle to be efficient and accurately parked without going into other parking spaces or into the walkway. However, there are several safety factors what should be considered. The first safety risk occurs at (1) were the vehicle comes close to the desired parking spaces. This is the smallest risk as parking spaces are designed to give vehicles additional space, if the vehicle crossed this line by a small amount it is unlikely to result in any damage. The largest safety risks are at (2) and (4), where the vehicle gets close to the walkway. If the vehicle goes over this walkway by any amount the test has failed. For this reason, a 1.5 safety factor was added to the stopping distance when the vehicle approaches the walkway. It can be seen that the vehicle leaves a gap of 20cm between the front of the vehicle and the walkway. For all other stopping distances, a safety factor of 1.1 was used.

6.5.2 Vehicles States

The vehicle is completely controlled by the use of the steering angle (δ) and the longitudinal velocity (V). This section all look at and discuss the behaviour of both this states. The two output states can be seen in Figure 42.

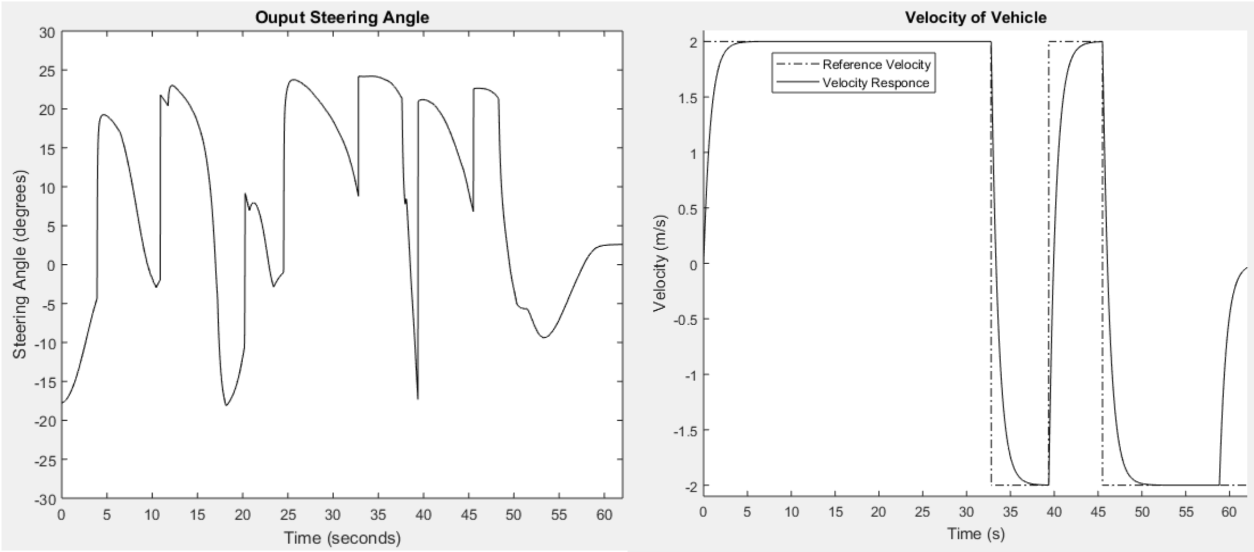


Figure 42, the transient behaviour of the vehicle's inputs.

Figure 42 shows the steering angle input on the left. The steering angle input never exceeds $\pm 30°$, this is critical as Ford's vehicle can only operate in this region. It can be seen that the steering angle often has large spikes requiring the steering wheel to be turned very quickly. In reality these steering wheel rates produced from the large change in steering input would have to be achieved by an actuator. These spikes could cause large actuation overshoots and other errors. However, some of the spikes are due to the switch from forwards to backwards steering input. Both controllers produce steering inputs for the vehicle throughout the simulation, the switch chooses which one based on the velocity. For example, the largest steering angle spike occurs when the vehicle switches from reversing to forwards at around $39s$.

In the "Velocity of Vehicle" figure it can be seen that the vehicle starts its automated parking system from stational ($0ms^{-1}$). The vehicle then responds to the reference input signal (2), and reaches 67% of this value in the time constant stated in Table 11, for the mass set $[16, 20.8)$. The reference velocity is the binary forwards and backwards signal that has passed through a gain ($-2Ref + 2$), when the forward signal is output (0) it is converted into $2ms^{-1}$. When the vehicle approaches the desired position in the parking space the vehicle stops using the velocity reference and engages the brakes, stopping at $y_D = 49.85m$. The acceleration response producing this velocity profile can be seen in Figure 44.

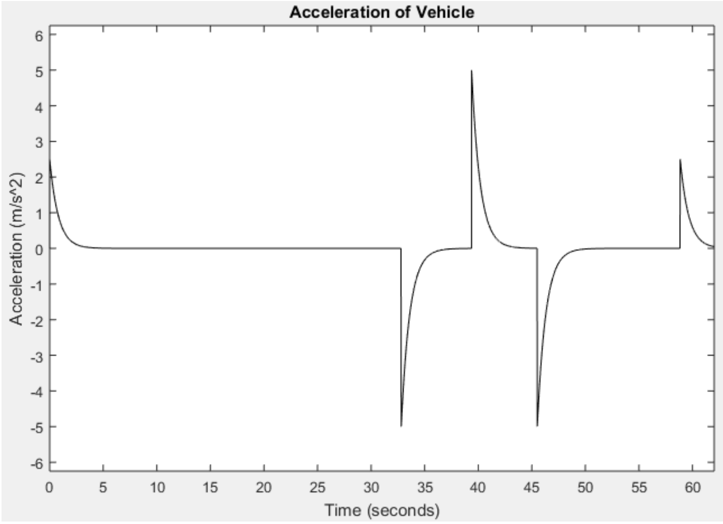


Figure 44, the acceleration of the vehicle throughout the simulation.

Figure 44 shows that the acceleration of the vehicle doesn't exceed $\pm 5ms^{-2}$. This maximum acceleration only occurs when the vehicle swaps from a forwards manoeuvre to a reversing manoeuvre, half the acceleration is experienced when the vehicle either accelerated from a stopped position or is braking to a stopped position. In (Hoberock, 1977) uncomfortable acceleration was found to be anything over 0.3g/s. From the above acceleration profile, the passenger would be uncomfortable for the first 1.3s of the two largest peaks. However, this data is taken from the acceleration of an undamped system, the driver is likely to have a comfortable chair which damps out some acceleration, the vehicle suspension would also damp out some of this acceleration. Therefore, it is unlikely the passenger will ever be in this uncomfortable range. The above acceleration is also the worst-case scenario, as heavier vehicles will accelerate less.

Figure 43, the hitch angle throughout the simulation.

Figure 43 shows the vehicle's hitch angle throughout the trajectory. The hitch angle never exceeds the maximum allowable value, and therefore it can be said that the reversing controller ensures that the vehicle remains asymptotically stable when reversing. Consider the final reversing manoeuvre, indicated by the data on the R.H.S of the red line. The vehicle's hitch angle is reduced to zero, then overshoots and is brought back to an acceptable value at the end of the simulation. The hitch angle looks like it will converge to a value below zero, however, this flattening of the hitch angles curve is due to the reducing in velocity. If the vehicle continued to reverse the hitch angle would converge to zero. The two dotted lines are the acceptable final hitch angle errors, if the hitch angle is between these lines at the end of the simulation then it has passed.

6.5.3 Controller Performance

The lateral error of the controller is very difficult to predict due to the unconventional reference generated, this reference can't easily be converted into one continuous signal. The error is based on the fuzzy rules desired truck or trailer orientation in a specific area of the parking space. Due to some of the steep fuzzy surfaces large errors could be shown when the vehicle crosses over certain areas of the parking space, this is not representative of the controller's abilities. It's clear from Figure 41 that the controller follows a smooth and efficient trajectory. The only way to test the controller's performance is to remove the FFPP and RFPP from the controller systems and test the controller's response to different inputs, step and sinusoidal.

Consider the vehicle having an initial orientation of $\theta_1 = -90°, \theta_2 = -90°$, the vehicle drives with a constant velocity of $2ms^{-1}$. The forward controller has a step reference input requesting a truck orientation angle of $\theta_{1req} = 0°$ after 10 seconds. This situation is analogous to the vehicle's first 30 seconds using the FFPP system, as the vehicle starts at the same angle in the parking entrance and then is asked to turn left once it has left the entrance. Figure 45 shows the response of the FFC to this step input.

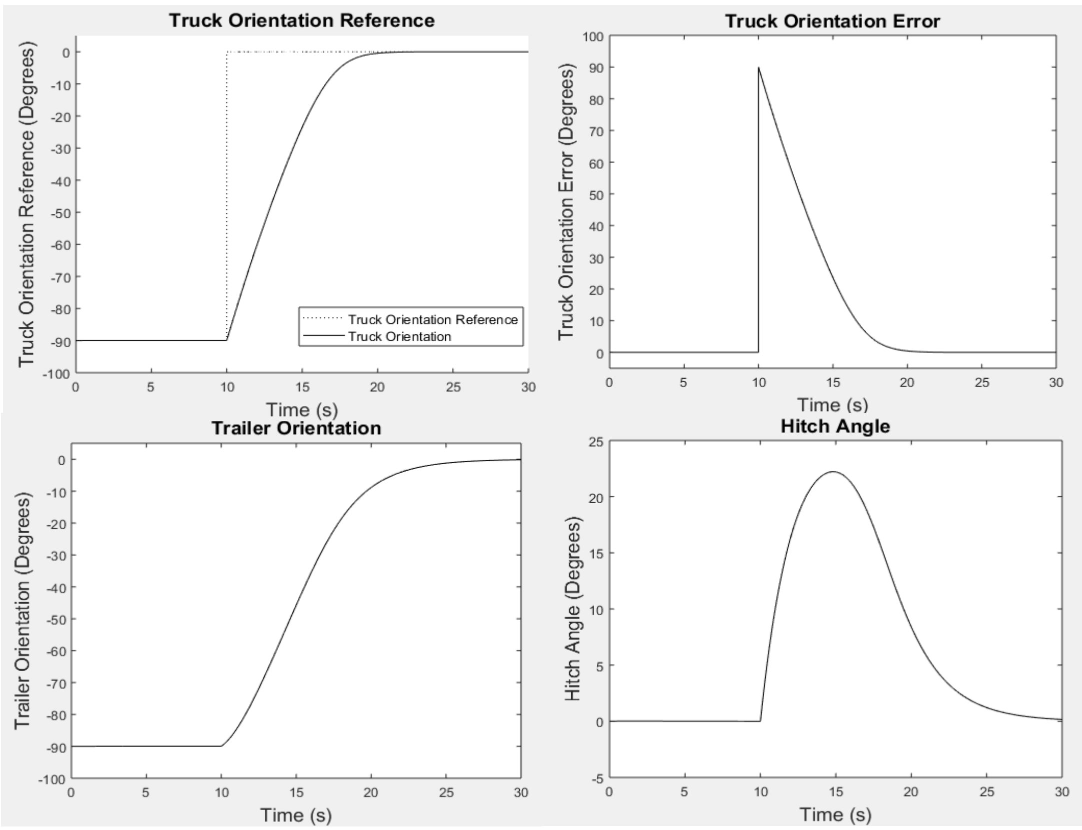


Figure 45, the response to the vehicle to a step input requesting a truck orientation of 0°.

From Figure 45 the controller response to the step input change can be seen in the top left figure. The controller manages to achieve the desired orientation with zero overshoot, zero steady state error, a time constant of $\tau = 4.6s$, and a settling time of $18.7s$. It can be seen that the trailer's orientation also achieves the same final result but with a larger settling time and time constant. The bottom left figure shows that the vehicles ability to self-stabilise the hitch angle without any control action for the forwards manoeuvre. A similar test was carried out with a sinusoidal wave as the input, with amplitude between $[-120, -60]$ and a time period of $65s$, it was found that the maximum truck orientation error was 3.2°. This shows that the FFC is able to follow a sensible reference trajectory (see Appendix 6.5.3.1 for the reference signal and the vehicle's response).

The overshoot seen in Figure 41, when the vehicle drives forwards from the starting position to (1) can be explained by the response time of the FFC. When the vehicle leaves the entrance the FFPP requests an angle of $-45°$, the vehicle achieves this. The vehicle is then requested to turn to 0° quickly and align with $y = 32.8m$, this is where the vehicle overshoots the desired path. However, the vehicle quickly recovered and ensures that it doesn't drive into other parking spaces. If the FFC was tuned to allow for a faster response time this overshoot could be avoided.

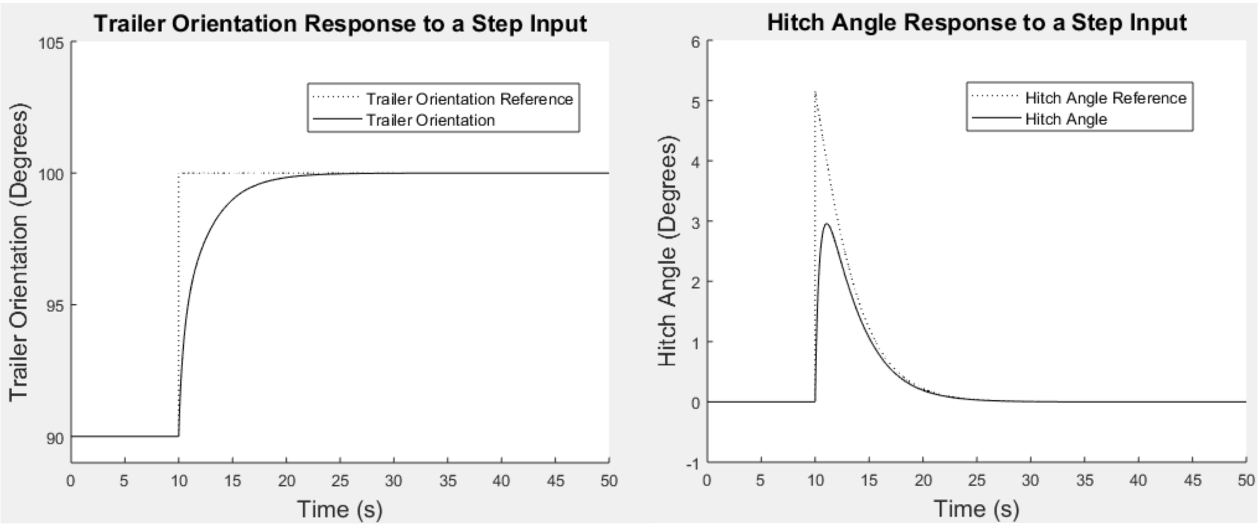


Figure 46, The response of the RFC1 and RFC2 to a trailer orientation angle change of 10°.

Figure 46 shows the reversing controller's response to the vehicle being required to change its trailer orientation from an initial angle of 90° to $100°$. The truck's orientation is 90°, the velocity of the vehicle is $-2ms^{-1}$, and the hitch angle is $0°$. It can be seen from the figure on the left that the reversing fuzzy controller is capable of changing the orientation of the trailer without any overshoot, the controller time constant is $11.8s$ and has a settling time of $19.6s$. The figure on the right shows how the trailer orientation error produces a large spike in reference hitch angle. The controller responds by outputting a steering angle that grows this hitch angle reducing the hitch angle error. It can be seen that the desired hitch angle is achieved after around $12s$.

The response of the controllers to this trailer orientation change is relatively slow. Consider the manoeuvre (2) to (3) in Figure 41. The trailer orientation is roughly $105°$, the RFPP requests that the orientation slowly changes over the x and y distance from the parking space. If the RFPP requested the trailer orientation to instantly be 90° the vehicle would take longer than $19.6s$ to achieve this. In that time the vehicle would have travelled 39.2m and gone well into the parking space. This is why the vehicle overshoots the desired parking position, and then has to drive forwards and reattempt. To reduce the number of manoeuvres and time spent parking the RFC1 and RFC2 could be tuned to allow for faster rise times.

6.6. System Robustness and Limits

The purpose of this section is to find the limits of what initial conditions it can still safely park from, To test the robustness of the automated system the vehicle's trailer length can be increased, and a maximum allowable trailer length can be found. From running simulations using randomly generated initial conditions it was found that the maximum allowable trailer length was $6.1m$, a $1.09m$ increase of Ford's Vehicle. If the trailer length is longer the vehicle either exceeds the time limit of 2 minutes or does more than 10 manoeuvres. The initial conditions are: $x_C = 5.78m, y_C = 6.84m, \theta_1 = -95.2°, \theta_2 = -102.4°, and\ m = 19.45\ tonnes$. The Error using Equation 21 was 4.87, which is well below the mean error. This shows that the by increasing the length of the trailer the final position error is not affected, only the simulation time and number of manoeuvres. The trajectory can be found in Appendix 6.6.1.

The initial orientations of the vehicle are limited depending on how close the vehicle is to the entrance sides. No automated system was designed to prevent vehicle orientations largely different to $-90°$ close to the entrance sides, this is because it is unlikely that a sensible driver would stop the vehicle close to and facing the sides of the entrance. If more time was permitted a reversing manoeuvre could be added in case the vehicle ever gets to close too the entrance walls.

The automated parking system was designed for the vehicle to be stopped somewhere in the parking entrance and then the system would be engaged by the driver. It is important to know where the automated system can be engaged for successful docking. Once situation is if the driver forgets to use the system and drives too far forward, resulting in the truck being in line with the beginning of the parking space. Initial conditions for this can be: $x_C = 5\ m, y_C = 32.4m, \theta_1 = -90°, \theta_2 = -90°, and\ m = 20\ tonnes$. The vehicle successfully docks well within the acceptable final position and orientation error. The trajectory can be seen in Figure 47.

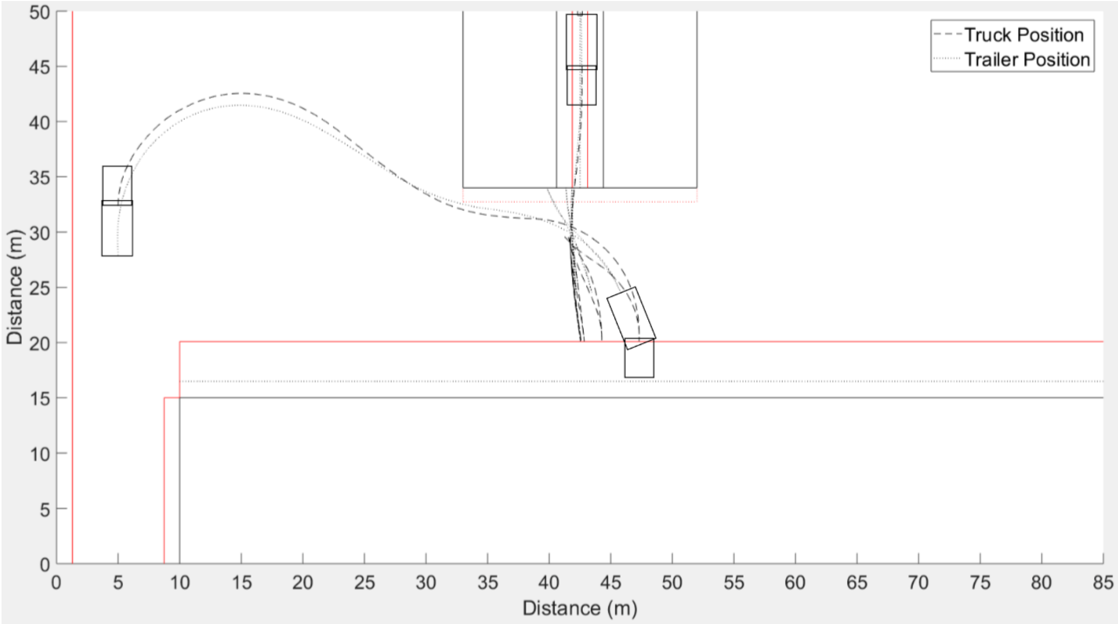


Figure 47, the trajectory outside the parking entrance.

It can be seen that the vehicle is very close to the parking space just before it turns to head towards the walkway. If the vehicle was stopped any further on the right, then the vehicle would drive over the other parking spaces. Again, this could be avoided by adding another system that reverses the vehicle if it gets too close to the parking spaces. Its clear from the above analysis that the vehicle can be automatically parked for all sensible initial conditions in the parking entrance as well as initial conditions outside of this designed area.

(Kong and Kosko, 1992) looks at testing the robustness of the fuzzy controllers by randomly deleting rules from each controller. It was found that the system still worked after removing 50% of the fuzzy rules. It was found that randomly deleting 50% of the rules from the path planners and controller resulted in a complete failure of the system, the vehicle drove forwards until it hits the back of the parking area. It was found that even deleting 15% of the FFPP systems rules result in failure to park, with the vehicle getting stuck trying to reverse. The reason the paper mentioned had so much success was due to its use of a neural network controller and fuzzy controller, so if the fuzzy controller's performance was impaired the neural controller could correct the behaviour. Also, the fuzzy controller mentioned in their work had 735 rules to allow for increase precision. Removing these rules reduced their precision but the key behaviour of the system with so many rules remained.

6.7. Improvements and Future Work

The system that calculates the velocity and acceleration of the vehicle is split into 5 different mass sets. This means that the vehicle's acceleration is exactly the same for a range of 4.8 tonnes. If more time was allowed a linear interpolation calculation would have been introduced that used the accelerations form other mass sets to more accurately predict the vehicle's acceleration and velocity response.

For the automated system to work in a real vehicle the vehicle's steering angle rate needs to be reduced. Currently the vehicle uses extremely high steering angle rates to achieve the correct trajectory. If filters are added to reduce the maximum steering angle rate to $200°/s$ the vehicle's trajectory is affected, and it is not guaranteed that it can park safely from all initial positions. The reason these steering rates are so large is due to the sensitivity of the FFC, RFC1 and RCF2. In future work, these controllers need to redesign to allow for a more sensible steering angle rate of around $100°/s$, currently the system cannot park the vehicle with this filter in place.

The final position and orientation errors of the vehicle always is inside the specified ranges; however, they could easily be improved. By adding a system that only accepts the vehicle's final position with a much tighter position and orientation tolerance. This could be achieved by making the vehicle drive forwards if the vehicle is not within this region. The distance the vehicle drives forwards could also be improved, as it doesn't have to drive to the walkway every time the position and orientation aren't sufficient. Instead the vehicle could drive forwards until the hitch angle is less than 1°, this would save time.

As discussed in section 6.5.3 the controller could be retuned to allow for better response times, this better response time could reduce the total amount of manoeuvre required. If the number of manoeuvres required to get the vehicle into the parking space is reduced, then more manoeuvres can be used to ensure the final position and orientations are more accurate.

The path planning system used for both forwards and backwards manoeuvres is unsafe. This is because it doesn't include an obstacle avoidance system. (Rohani, Sharafi and Zare, 2011) looked at using an obstacle avoidance system combined with a fuzzy path planner to ensure the vehicle could dock and avoid potential hazards, both moving and stationary, in the parking space. The obstacle avoidance system overrides the fuzzy path planning system anytime an obstacle is detected. If more time was permitted this could be added. The system hierarchy would be in this order: Obstacle avoidance, avoid parking boundary limits, follow parking trajectory. Another safety feature that could be added is a system that either drives the vehicle forwards or backwards when it gets too close to the parking space boundaries. Currently this is only done for the walkway, if the vehicle doesn't fit inside the parking space when reversing, and when the vehicle gets too close to the end of the parking space.

In order for the FFPP and RFPP to work in a real vehicle, the vehicle would be required to have a GPS system. This system should be calibrated to allow for the same coordinate system described above to be used, this will allow for the vehicle's path planners to have constantly updated coordinates allowing for a safe and controlled parking manoeuvre.

7. Conclusion

In this thesis an automated parking system was designed using fuzzy logic, this system was then simulated using Simulink. It was found in the results that the system was capable of parking all of the simple initial positions within the range of acceptable predefined KPIs. These simple simulations are supposed to represent the majority of initial conditions the system will encounter from sensible drivers. It was found that the vehicle parked faster when the system was engaged on the L.H.S of the parking space but achieved higher precision when on the R.H.S. The vehicle also showed realistic results regarding loaded and unloaded simulations as the vehicle would take longer to complete the docking when fully loaded and could require more manoeuvres.

The fuzzy path planners used offer a simple and effective way of generating a reference signal for the fuzzy controllers. However, for this system to work the vehicle would need to be fitted with a GPS tracking system that has been calibrated so that the bottom left of the parking space is the origin. The path planners are limited to only working inside this specific docking area and this specific parking space. Future work would involve making the path planners more robust, allowing for different parking space geometries and parking positions. This was done in (Riid, Lei and Rüstern, 2009).

The complex simulations used random initial conditions that were all selected from a wide range of values. This allowed the vehicle to simulate a potentially less sensible initial condition that could occur due to driver negligence. It was found that 4/5 of these simulations were successful, completing the simulation within all the acceptable predefined KPIs. However, one of the simulations failed due to the vehicle getting stuck in the parking space. This failure doesn't nullify the use of the parking system as it managed to control the vehicle safely up until the trailer is fully submerged in the desired parking space. If the driver senses a fault in the system, they could disengage the system and complete the small section of the parking manoeuvre themselves. Of course, this is not desirable, but it occurs very seldom. To fix this problem the system discussed in section 6.7 could be developed.

Overall the automated parking system designed shows promising results. The worst position and orientation error found in the 20 test was only within 60% of the maximum allowable error. The system was then compared to a similar design found in the literature, the results showed a slightly larger error. However, the system design in this thesis has a much more complex parking space.

Finally, the robustness of the controller was reviewed, and it was found that the system was able to park the vehicle even when the driver had purposely sabotaged the system by driving a considerable distance from the docking entrance. Nevertheless, the system still required considerable improvements to allow for difficult parking situations to be considered safe, these have all been addressed in section 6.7.

8. Bibliography

Guo, J., Luo, Y. and Li, K. (2018). Adaptive non-linear trajectory tracking control for lane change of autonomous four-wheel independently drive electric vehicles. IET Intelligent Transport Systems, 12(7), pp.712-720.

Moran, A. (2017). Autonomous path following of truck-trailer vehicles using linear-fuzzy control. [online] researchgate. Available at: https://www.researchgate.net/publication/317701705_Autonomous_path_following_of_truck-trailer_vehicles_using_linear-fuzzy_control [Accessed 20 May 2019].

Yi, J., Yubazaki, N. and Hirota, K. (2001). Backing up control of truck-trailer system - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/1007355 [Accessed 20 May 2019].

Cheng, J., Zhang, Y. and Wang, Z. (2009). Backward Tracking Control of Mobile Robot with One Trailer via Fuzzy Line-of-Sight Method - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/5359093 [Accessed 20 May 2019].

GonzBlez-Cantos, A., Maza, J. and Ollero, A. (2001). Design of a stable backing up fuzzy control of autonomous articulated vehicles for factory automation - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/996401 [Accessed 20 May 2019].

9. References

Eatherley, G. and Petriu, E. (1995). A fuzzy controller for vehicle rendezvous and docking - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/387339 [Accessed 19 May 2019].

T. Mason, M. (2007). Lecture 5. Nonholonomic constraint. [online] Cs.cmu.edu. Available at: http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/lectures/lecture5.pdf [Accessed 19 May 2019].

Backman, J., Oksanen, T. and Visala, A. (2012). Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/pii/S0168169911003218 [Accessed 19 May 2019].

Pradalier, C. and Usher, K. (2008). Robust trajectory tracking for a reversing tractor trailer. Journal of Field Robotics, [online] 25(6-7), pp.378-399. Available at: https://go.openathens.net/redirector/surrey.ac.uk?url=https%3A%2F%2Fonlinelibrary.wiley.com%2Fdoi%2Ffull%2F10.1002%2Frob.20241.

Divelbiss, A. and Wen, J. (1997). Trajectory tracking control of a car trailer system. Estados Unidos: The Institute of Electrical and Electronics Engineers, Inc-IEEE.

Hoel, C. and Falcone, P. (2013). Low speed maneuvering assistance for long vehicle combinations - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/6629532 [Accessed 19 May 2019].

Leng, Z. and Minor, M. (2017). Curvature-Based Ground Vehicle Control of Trailer Path Following Considering Sideslip and Limited Steering Actuation. IEEE Transactions on Intelligent Transportation Systems, 18(2), pp.332-348.

Kong, S. and Kosko, B. (1992). Adaptive fuzzy systems for backing up a truck-and-trailer. IEEE Transactions on Neural Networks, 3(2), pp.211-223.

Yang, X., Yuan, J. and Yu, F. (2006). Backing Up a Truck and Trailer Using Variable Universe Based Fuzzy Controller - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/4026175 [Accessed 19 May 2019].

G. Siamantas and S. Manesis. (2009). Backing-up Fuzzy Control of a Truck-trailer Equipped with a Kingpin Sliding Mechanism. Boston Springer, pp 373-378. Available at: https://link.springer.com/chapter/10.1007/978-1-4419-0221-4_44

Tanaka, K., Taniguchi, T. and Wang, H. (1997). Fuzzy controller and observer design for backing control of a trailer-truck. [online] Science direct. Available at: https://www.sciencedirect.com/science/article/pii/S0952197697000250 [Accessed 19 May 2019].

Tanaka, K. and Sano, M. (1994). A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer. IEEE Transactions on Fuzzy Systems, 2(2), pp.119-134.

Zimic, N. and Mraz, M. (2006). Decomposition of a complex fuzzy controller for the truck-and-trailer reverse parking problem. Mathematical and Computer Modelling, 43(5-6), pp.632-645.

Kodituwakku, S. (2011). Fuzzy and ANN Control System for Backing up a Truck and Trailer. [online] researchgate. Available at: https://www.researchgate.net/publication/261288432_Fuzzy_and_ANN_Control_System_for_Backing_up_a_Truck_and_Trailer [Accessed 19 May 2019].

Riid, A., Lei, A. and Rüstern, E. (2009). Fuzzy Backing Control of Truck and Two Trailers - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/4097651 [Accessed 19 May 2019].

Cariou, C. and Lenain, R. (2010). Path following of a vehicle-trailer system in presence of sliding: Application to automatic guidance of a towed agricultural implement. [online] Sjsu.edu. Available at: http://www.sjsu.edu/people/burford.furman/docs/me190/Path%20following%20of%20a%20vehicle-trailer%20system%20in%20presence%20of%20sliding.pdf [Accessed 19 May 2019].

Altafini, C., Speranzon, A. and Wahlberg, B. (2001). A feedback control scheme for reversing a truck and trailer vehicle - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/976025 [Accessed 19 May 2019].

Mehran, K. (2008). Takagi-Sugeno Fuzzy Modeling for Process Control. [online] Staff.ncl.ac.uk. Available at: https://www.staff.ncl.ac.uk/damian.giaouris/pdf/IA%20Automation/TS%20FL%20tutorial.pdf [Accessed 19 May 2019].

Uk.mathworks.com. (2012). Comparison of Sugeno and Mamdani Systems- MATLAB & Simulink-MathWorks United Kingdom. [online] Available at: https://uk.mathworks.com/help/fuzzy/comparison-of-sugeno-and-mamdani-systems.html [Accessed 19 May 2019].

Chang, Y. and Ma, S. (2009). A kinematic modeling method for a wheeled mobile robot - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/5246109 [Accessed 19 May 2019].

Rajamani, R. (2012). Vehicle Dynamics and Control. 2nd ed. Minnesota: Springer, p.21.

Waslander, S. and Kelly, J. (2017). Lesson 2: The Kinematic Bicycle Model - Module 4: Vehicle Dynamic Modeling | Coursera. [online] Coursera. Available at: https://www.coursera.org/lecture/intro-self-driving-cars/lesson-2-the-kinematic-bicycle-model-Bi8yE [Accessed 20 May 2019].

Ren, T. (2013). Design of an Automatically Tuned Fuzzy Controller for a Truck and Multitrailer System. Advances in Mechanical Engineering, 5, pp.2,3,4.

Zadeh, L. (1965). Fuzzy Sets. [online] Science Direct. Available at: https://www.sciencedirect.com/science/article/pii/S001999586590241X?via%3Dihub [Accessed 20 May 2019].

Nainar, I. (1996). An adaptive fuzzy logic controller for intelligent networking and control. [online] Ro.ecu.edu.au. Available at: https://ro.ecu.edu.au/cgi/viewcontent.cgi?referer=https://www.google.co.uk/&httpsredir=1&article=2467&context=theses [Accessed 20 May 2019].

Sangalli, A. (2009). Fuzzy inference systems. [online] Cs.princeton.edu. Available at: http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm [Accessed 20 May 2019].

Yang, G., Xu, H., Wang, Z. and Tian, Z. (2016). Truck acceleration behavior study and acceleration lane length recommendations for metered on-ramps. International Journal of Transportation Science and Technology, [online] 5(2), pp.93-102. Available at: https://www.sciencedirect.com/science/article/pii/S204604301630034X.

Industries, A. (2017). Raspberry Pi 3 - Model B+ - 1.4GHz Cortex-A53 with 1GB RAM. [online] Adafruit.com. Available at: https://www.adafruit.com/product/3775 [Accessed 20 May 2019].

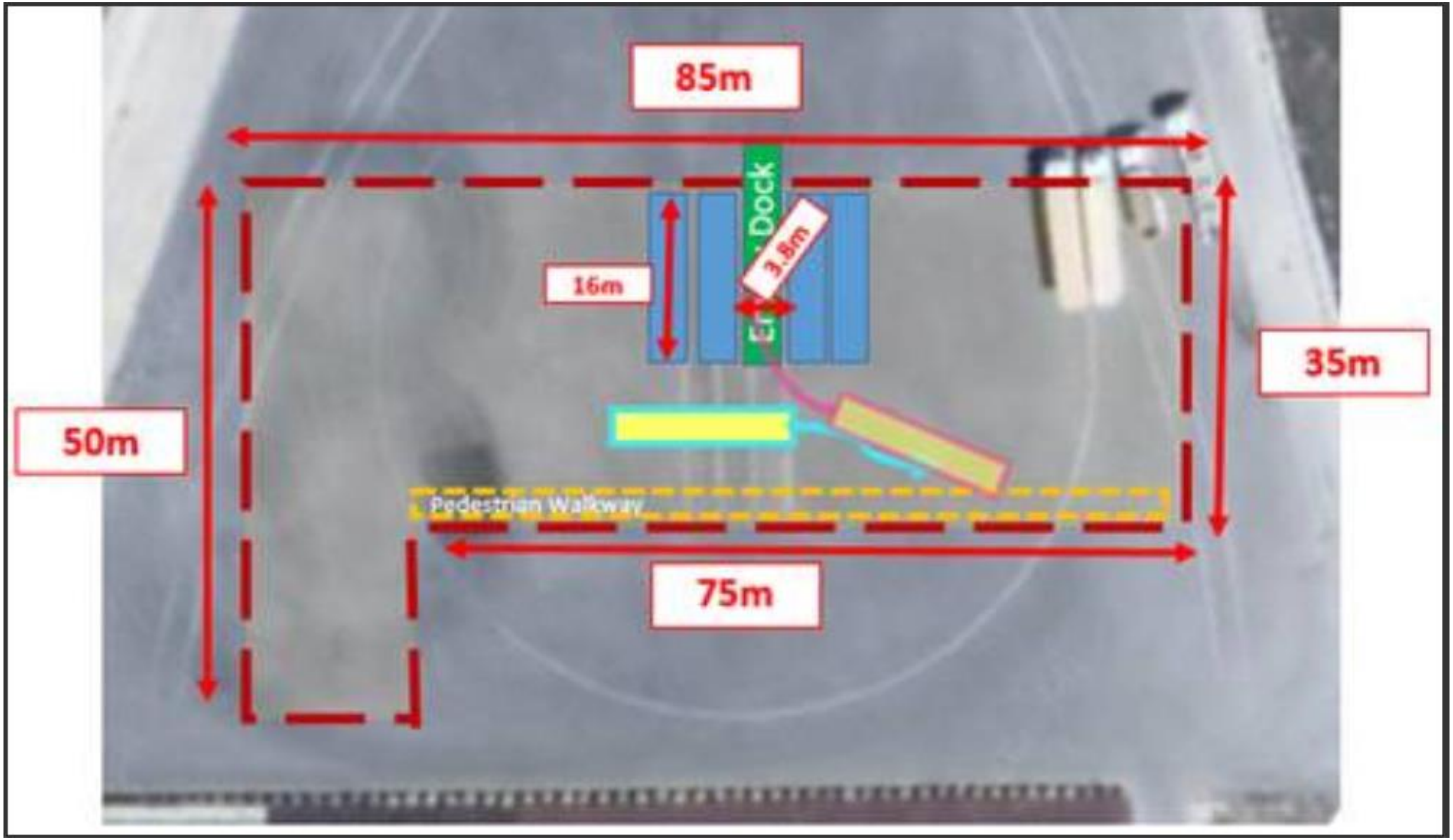
Hoberock, L. (1977). A Survey of Longitudinal Acceleration Comfort Studies in Ground Transportation Vehicles. Journal of Dynamic Systems, Measurement, and Control, 99(2), p.76.

Kong, S. and Kosko, B. (1992). Adaptive fuzzy systems for backing up a truck-and-trailer. IEEE Transactions on Neural Networks, [online] 3(2), pp.211-223. Available at: https://ieeexplore.ieee.org/document/125862.

Rohani, J., Sharafi, M. and Zare, A. (2011). Intelligent Parking Method for Trucks in Presence of Fixed and Moving Obstacles Randomly. Applied Mechanics and Materials, [online] 110-116, pp.4373-4380. Available at: http://ajbasweb.com/old/ajbas/2011/November-2011/398-416.pdf.

10. Appendix

Appendix 1.1



Appendix 2.4.1:	Summary							1: Highest 10:Lowest
Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Adaptive non-linear trajectory tracking control for lane change of autonomous four-wheel independently drive electric vehicles.	To design a controller system that allows for safe autonomous lane changing of electric four-wheel drive vehicles.	Lane changing control on straight road for four-wheeled autonomous electric vehicles	Bicycle model, here they have used a 12 DOF model, where all 4 wheels have 3DOF.	Uses a higher DOF bicycle model. Four-wheeled autonomous passenger vehicle.	Backstepping controller Adaptive fuzzy controller.	Lateral position error Longitudinal position error Orientation error Lateral force error Longitudinal force error All time based.	The control system used was compared to the reference trajectory and a SMC control strategy. The proposed controller outperformed the SMC controller and tracked the reference trajectory with less error.	*7/10*
Robust Trajectory Tracking Error Model-Based Predictive Control for Unmanned Ground Vehicles.	Design a controller that can accurately follow a reference trajectory for a tractor and trailer.	Tractor trailer at low speeds off road following a trajectory with changing curvatures.	6DOF bicycle model, truck and trailer are joined via two different revolute joints. Kinematics used!	Tractor and trailer bicycle model used. Comparison of NMPC and LMPC control performance and computational time.	Nonlinear MPC was used.	Error in the x and y directions. Time based error from the reference trajectory Tractor and trailer steering angle.	Its shows that the NMPC controller tracked the reference trajectory with a maximum Euclidian error of 39cm The NMPC was compared to a LMPC. It was shown stated that the NMPC had smaller maximum error but would take 6times longer to compute (6ms compared to 1ms)	*7/10*
Trajectory planning and yaw rate tracking control for lane changing of intelligent vehicle on curved road.	A controller that accurately tracks a trajectory of a vehicle changing planes on a constant curvature highway.	Lane changing of a road vehicle on a constant curvature road.	Modelled the system as a particle Equations for longitudinal and lateral acceleration.	Lane changing on a constant curvature road. Particle system modelling.	Use of nonlinear sliding model controller.	Yaw angle error vs time Position errors vs time Lateral velocities vs time Steering angle vs time All with a variety of road curvatures.	The controller used showed smaller errors for the road with a larger radius of curvature. The time response of showed that for the harshest curvature the error was -0.03.s	*7/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Low speed Manoeuvring Assistance for Long Vehicle Combinations.	Design of a controller to track a tight 90 degree trajectory that requires the truck and trailer to reverse in order to complete the turn.	Track a "tight" 90degree path for a truck and trailer forwards and backwards. Tight meaning that it is not possible to complete the manoeuver in one turn.	Bicycle model: Kinematic model used.	Tight cornering that requires a revers. Constant velocity (1m/s, 2m/s and 3m/s reviewed). The speed is decided by the driver. They shifted from speed domain to distance domain.	LQR controller (linear quadratic)	Lateral error to be minimalised for different constant speeds.	The LQ controller accurately kept the long vehicle within an acceptable lateral error at all speeds. 2m/s the max lateral error was 79mm. 3m/s the max error was 170mm.	*8/10*
Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer	Design a fuzzy control system (with neural networking) to be used for docking of a truck and truck-trailer vehicle.	Docking of a truck and truck-trailer in a parking lot.	Kinematic model.	The use of both a fuzzy logic and neural networking controller for backing up. A variety of different starting positions was used. The introduction of sabotage rules into the fuzzy logic. As well as purposely confusing rules. This can be used to measure robustness.	Fuzzy logic controller used with a neutral networking controller.	For the vehicle to successfully backup to a location in space. Do this while having the least amount of fuzzy logic rules.	The fuzzy controller successfully backed up the truck and truck-trailer from different start positions. The controller only started to perform badly when more than 50% of the logic rules were removed. This suggests extreme robustness.	*8/10*
Path-Tracking for articulated Vehicles with Off-axle Hitching	To develop a path tracking procedure that is based on exact linearization.	Considering both 2 and 3 body articulated vehicles moving backwards along a curving path.	Kinematic model, low speed manoeuvres. Off-axle	In the modelling an off-axle hitch point is used. The questions are also linearized using a ghost vehicle. The controller developed for the ghost vehicle is then used on the actual vehicle.	N/A	The distance that is travelled along the path until the lateral offset is smaller than 1m. The stability of the configuration vectors for which the controller is able to achieve asymptotic tracking.	Methodology put forth is sufficient for the path tracking of articulated vehicles with 2 –3 trailers in both forward and backwards motions.	*7/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Curvature-Based Ground Vehicle control of Trailer Path Following Considering Sideslip and Limited Steering Actuation.	A controller that tracks the trajectory of a trailer including considerations to sideslip and limited steering actuation.	Trailer following a set trajectory with two different bends. This is done forwards and backwards. (1.5m/s and 0.5m/s)	Kinematics used to model the behaviour. Low speeds.	The use of extended Kalman filter (EKF) sideslip eliminator and a curvature based controller used together to control trailer trajectories.	Use of a curvature-based controller.	Lateral error compared to two different types of controllers. "Pradalier controller" and "carious controller" one for backwards and one for forwards control of a trailer.	The controller considered shows reduced error for both forwards and backwards simulations. The controller performed better on the spiral curve than on the fixed radius curve. When used on a full size car it performed within a good accuracy.	*8/10*
Path following of a vehicle-trailer system in presence of sliding: Application to automatic guidance of a towed agricultural implement.	Design a controller that keeps the trailer on a pre-set trajectory for any trajectory shape or ground conditions.	Agricultural trajectory control of a trailer attached to a tractor. Constant speed of 1.4m/s was used.	Kinematics used to model the behaviour. Low speeds. Trailer was modelled as a virtual trailer.	The design of a control algorithm that minimises the lateral error of a trailer for any type of ground conditions.	The design of a control algorithm (later mentioned that a MPC could have been used).	The lateral error of both the trailer and the truck at different points.	The proposed control algorithm follows the trajectory path with a maximum overshoot of +/-20cm at the beginning of a corner and an average lateral error of +/-10cm.	*8/10*
Navigation system for agricultural machines: Nonlinear Model Predictive path tracking.	Develop a controller that can keep the lateral error of a trailer within 10cm at 12km/h	Agricultural trajectory of a trailer to ensure that the swaths are parallel and do not overlap.	Kinematic model. Longitudinal, lateral, rate of change of heading angle and the slip factor. Low speeds.	Using a NMPC to keep both the tractor and the trailer on path. (Path doesn't involve time…).	NPMC, with the combination of GPS and an extended kalman filter for global positioning. The position of the trailer used a 2D lazer scanner.	For the lateral error to be under 10cm for a constant speed of 12km/h. 10cm error based on human error, if under 10cm better than average human.	Comparison between the NMPC and the "traditional path" tracking method. The results are from experimental testing: The lateral error for the tractor and trailer are within the 10cm error limit (which is the predicted human error).	*8/10*
Trajectory Tracking Control of a Car-Trailer System.	Generate a trajectory path from experimental data, then design a controller that tracks the trajectory accurately.	Road vehicle and trailer. Various trajectories tested: parallel parking and reverse docking of the trailer.	Kinematic model used to model the longitudinal, lateral, rate of change of steering angle and the rate of change of car orientation. Low speeds.	Generates an offline path then applies the control strategy to the planned path. This is done for a road vehicle with a trailer(s), for both docking	Use of a time varying LQR (linear quadratic regulator) controller	Longitudinal and lateral position in time. (error from x-y path) Steering angle.	Parallel: Time taken to compute path was 10mins, followed path to very accurate level. 120s to execute. Docking: computational time was 14min, and execution time was 70s. Steering angles show large errors, x-y position error is minimal.	*8/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
A Feedback Control Scheme for Reversing a Truck and Trailer Vehicle	To develop a control strategy for the stabilization of backwards moving truck-trailer vehicles.	The vehicle starts in a jack-knife position, it then needs to move forwards to straighten out and then reverse on a straight and curved path.	Kinematic model, low speed manoeuvres. Off-axle	The use of two controllers, one for backwards movement and the other for forwards movement.	LQ controller	Minimise the lateral offset from the reference line. Remain in a stable configuration.	(No numbers given, expect from numbers on figures). The controller strategy successfully swaps between forward and backwards controllers to follow the reference path while minimising the lateral error.	*8/10*
Extension of Trajectory Planning in Parameterized Space to Articulated Vehicles	Develop an algorithm that allows for the safe manoeuvring of articulated vehicles.	4 scenarios considered; 1). Trajectory tracking in an open space. 2). Reverse out and then drive out of warehouse. 3). Drive into warehouse and park. 4). Reverse park in warehouse.	Kinematic model. Low speed.	The development of a trajectory planning algorithm to be used on articulated vehicles in warehouses.	(algorithm developed not controller)	Success rate of finding a correct path. Time taken to find path. Iterations to find path.	Scenarios 1,2 and 4 all had a 100% success rate when producing a path and following said path. However, 2 and 3 had very high solving times 270.14s and 244.57s respectively.	*6/10*
Robust Trajectory Tracking for a Reversing Tractor Trailer	Design a controller that is used in the reversing of a tractor trailer (note: the vehicle is also required to move forwards when in jack-knife positions.	Reversing on a circle with fixed radius Reversing on a figure of 8. A real world path, a short forward path followed by a long 160m reversing path.	Kinematic model, low speeds Off axle	Control of the vehicle using the hitch angle.	PID	Lateral error Comparison to real world driving skill.	Results show that the control strategy proposed allows for accurate control of forward and backwards trajectories. In general, the control strategy preforms better than low to medium skilled drivers. However, the human drivers complete the task much faster	*8/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Backing up a truck and trailer using variable universe based fuzzy logic controller	Design of a fuzzy controller used for the backing up of a truck-and-trailer vehicle.	The truck and trailer is made to backup no a line given as x=50.	Kinematic bicycle model, low speed. On axle Constant speed (-3m/s)	The uses of a VUBFC fuzzy controller (variable universe based fuzzy controller)	Fuzzy controller (VUBFC)	Lateral error, settling time, response time	VUBFC has a faster response time, settling time and less lateral error compared to another fuzzy controller mentioned in reference [2]. Reduces the SSE to zero and less overshoot, whereas the other controller has more overshoot and cannot reduce SSE to zero. VUBFC satisfies conditions in 18.7s whereas the other controller only does this in 31.3s. The VUBFC needs minimal tuning while at the same time achieves excellent performance.	*8/10*
Decomposition of a complex fuzzy controller for the truck-and-trailer reverse parking problem	To design a fuzzy controller for the backing up of a truck and trailer vehicle with a HFCS	Reversing of a truck and trailer onto different x=c lines from different starting positions.	Kinematic bicycle model. Instead of rates of change a time step method has been used. Constant speed.	The use of a hierarchical fuzzy controller to reduce the number of rules needed while still having the same level of control. Aka the use of a HFCS.	Fuzzy controller with a decomposition process to reduce the number of rules required.	The ability for the vehicle to reverse park from different initial points and angles. A comparison between a conventional FLC and a HFS for four initial positions was considered	It was found that the FLC only successfully parked one of the four initial positions whereas the HFS parked all four.	*8/10*
Fuzzy Backing Control of Truck and Two Trailers	Design a fuzzy logic controller to successfully reverse park a truck and two trailers into a docking bay.	Reveres a truck and two trailers into a docking bay from 10 random locations.	Bicycle model – kinematics. Use of time steps rather than differentials.	Truck with two trailers reverse docking. Use of fuzzy and PD/P controllers. The PD/P controllers are used to control the hitch angles ensuring the vehicle doesn't enter jack-knife.	Fuzzy logic controller used with PD controllers to minimise the hitch angles error.	Number of successful dockings out of the 10 (10/10). The quality is then evaluated by calculating a sum of the distance error and orientation error.	The controller system managed to dock all 10 randomly selected positions. Position 4 showed the least error, whereas position 7 showed the largest summed error.	*8/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Fuzzy Logic and Neural Network Control System for Backing up a Truck and a Trailer	The design of a controller that can reverse dock a truck and trailer vehicle without the need to drive forwards.	Constant speed reverse docking of a truck and trailer without moving forwards.	Bicycle model – kinematics. Use of time steps rather than differentials.	Docking T-T vehicle without moving forwards. Use of fuzzy controller compared to a neural network controller	Two fuzzy logic controllers and a neural network controller. Both then compared.	Comparison of the two controllers. Time taken for a successful simulation. Final x and y position from the dock. Overshoot for both of these.	The fuzzy controller out preformed the neural networking controller.	*8/10*
Backward Tracking Control of Mobile Robot with One Trailer via Fuzzy Line-of-Sight Method	Design of a controller that can successfully back up a T&T vehicle along a way point trajectory.	Backing up of a T&T vehicle along a way point trajectory	Bicycle model – kinematics. Modelled both two points on the vehicle for x,y position.	The use of a line-of-sight (LOS) fuzzy controller and trajectory based on way points.	LOS fuzzy controller	Error in the position of the distance vector at the rear of the trailer Error in the difference between the truck and trailer orientation.	“Satisfactory tracking performance” Some large position errors but only for a small amount of time 1-2 seconds.	*7/10*
Autonomous path following of truck-trailer vehicles using linear-fuzzy control	Design of a controller that backs a T&T vehicle into a desired position/ trajectory	Backing up into a y line defined position Backing up into a y=x defined position Backing up into a circular defined position. Backing up into a sinusoidal position.	Bicycle model – kinematics. Simplistic model using chained representation of operating conditions.	T&T vehicle backing up into a multiple positions and then following a trajectory after reaching that position.	Fuzzy logic controllers using both line of sight and perpendicular desired position method.	Achieve asymptotically stable positions at the desired point and then continue on a trajectory without hitting obstacles.	All scenarios achieved the desired position in from two selected initial positions. The linear trajectories showed less error than the circular and sinusoidal trajectories. Along both the circular and sinusoidal trajectories the vehicle came very close to the boundaries.	*8.5/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Backing-up Fuzzy control of a truck-trailer equipped with a kingpin sliding mechanism	Design of a control system that can back-up a T&T vehicle while avoiding jack-knifing	A T&T vehicle reversing onto a straight line position from an arbitrary initial position which avoiding jack-knifing	Bicycle model – kinematics The trailer is joined to the truck on the axle via a sliding kingpin mechanism	Fuzzy controller backing up of a T&T vehicle with a on axle kingpin mechanism	Fuzzy controller	To avoid jack-knifing and to manoeuvre to the reference path within the space provided.	The kingpin mechanism allows for the vehicle to avoid getting into the jack-knife position. One of the scenarios there is little difference between a vehicle with and without the kingpin.	*7/10*
A fuzzy controller for vehicle rendezvous and docking	Design of a controller for the reverse docking of a T&T vehicle	A T&T vehicle docking using both forward and backwards movement to avoid jack-knife.	Bicycle model – polar coordinates No mathematical description of the vehicle given…	Forward and reverse movement for the docking of a T&T vehicle. Non-constant velocity	Fuzzy controller	Ability to reduce the error between alpha and beta (dock angle and trailer angle relative to docking angle) Ability to avoid jack-knifing.	The controller designed managed to successfully dock the vehicle without jack-knifing from several starting conditions.	*8/10*
Fuzzy Controller and Observer Design for Backing Control of a Trailer-Truck	Design of a fuzzy controller and fuzzy observer to control the backing up of a truck and trailer vehicle.	A T&T vehicle reversing onto a horizontal line in space. This movement has been designed to avoid the jack-knife position.	Bicycle model – kinematics A transient model is used.	Design of the control system is based on what is realistic (at the time) to observe as states. For this reason they use a fuzzy observer to estimate the trailer angle.	Fuzzy controller with fuzzy observer	Response time Overshoot Steady state error	The controller managed to track the horizontal path accurately from a variety of different starting positions.	*7/10*

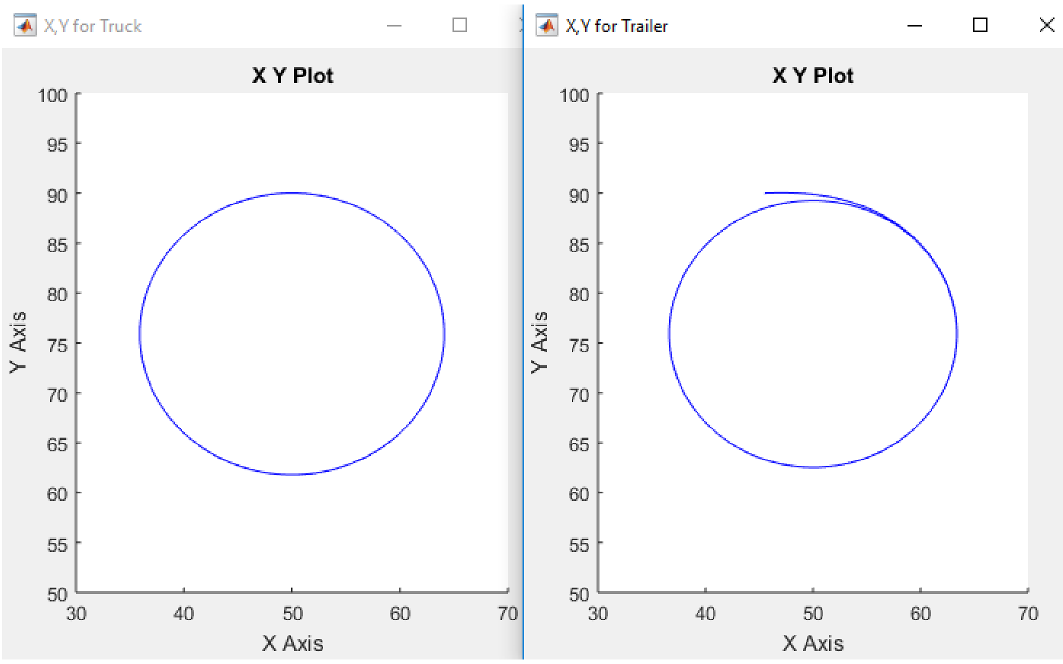
Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Design of an Automatically Tuned Fuzzy controller for a Truck and Multitrailer System	To design an algorithm that automatically tunes fuzzy controllers. This is tested on a forward moving truck and trailer(s).	The forward motion of a truck and trailer(s) from an arbitrary position to a way point.	Bicycle model – kinematic Off axle vehicle	The use of a PSO (particle swarm optimisation) algorithm to reduce the number of fuzzy rules and to auto-tune the controller.	Fuzzy controller with PSO algorithm.	Total error of the x and y position to the truck to the way point. Error in the trucks orientation to the way point desired orientation.	The algorithm successfully auto-tuned the controller and reduced the number of fuzzy rules from 243 to only 6. The resulting rules were then run giving satisfactory total errors.	*7/10*
Design of a Stable Backing up Fuzzy Control of Autonomous Articulated Vehicles for Factory Automation.	To design a fuzzy controller that is stable for large hitch angles.	Reversing (only) to dock T&T vehicles in industrial areas. E.g. using tight angles.	Kinematic bicycle model. Low speed, on axle.	Ensuring stability of the T&T for large hitch angles.	Fuzzy controller. The controller only has two inputs, both vehicle orientations (seems very limited).	Ability to avoid jack-knife position and to remain stable when the hitch angle is large.	Simulation showing that the T&T remains stable for large hitch angles.	*6.5/10*
A Robust Stabilization Problem of Fuzzy Control Systems and its application to backing up control of a truck-trailer.	To design a robust fuzzy controller using the Lyapunov approach.	Reversing of a truck and trailer vehicle. The vehicle can only reverse.	Kinematics bicycle model (discrete form). Low revering speed. Off-axle vehicle.	Ensuring stability of the controller for small hitch and steering angles.	Fuzzy controller. Controller inputs are x, y position, hitch angle, truck angle and trailer angle. Controller outputs are the steering angle.	Revering smoothly onto a constant y line (no end position required). When in jack-knife position the vehicle reduces its hitch angle and reverses to the reference stably.	Comparison of different initial position of the vehicle, including jack-knife position. Comparison between two different stability approaches. Both achieve similar results.	*7/10*
Backing up control of Truck-Trailer system.	Design a fuzzy controller with minimum rules that successfully backs up a T&T system from a variety of initial positions.	Backing up only. Different initial positions. Jack-knife included. Only reversing.	Kinematic bicycle model (discrete). Low speed revering. Off axle model.	The use of SIRM and DID in the fuzzy controller design.	Fuzzy controller using both SIRM and DID.	Time taken, position and angle of the T&T measured from the reference path (y = constant). Results taken for 4 different initial positions.	Comparison of time taken and position compared to other fuzzy controller designed in other papers. Results show either better or similar results.	*7/10*

Patent title & No	Objectives	Scenario	Vehicle modelling technique	Contributions /Novelty	Type of the trajectory planning strategy	Key Performance Indicator	Results & Conclusion	Ranking 1-10
Neuro-Fuzzy Optimal Control of Backing up a Trailer Truck.	To design a controller that makes use of both neuro and fuzzy controllers to back-up a T&T vehicle into a loading dock.	Backing up of a T&T system for docking. Only the orientation relative to the reference line was looked at, not the location along the line.	Kinematic bicycle model, discrete model used. Only reversing.	The use of both fuzzy and neuro controllers. Using Gaussian membership functions to minimise the cost function of both input and output variables.	Fuzzy and neuro controller.	Time taken, position and angle of the T&T measured from the reference path (y = constant). Results taken for 2 different initial positions.	Comparison between previous papers and this paper for position error under a selected time interval.	*7/10*
Intelligent Parking Method for Truck in Presence of Fixed and Moving Obstacles and Trailer in Presence of Fixed Obstacles.	The design of a fuzzy controller for the backward docking of a T&T vehicle while being able to avoid moving objects.	Reverse docking while avoiding moving and stationary objects.	Kinematic bicycle model, discrete version. Only reversing.	Reversing while avoiding obstacles that are stationary and moving with a constant velocity.	The use of two fuzzy controllers, one for backing up and the other for obstacle avoidance.	Ability to complete back vehicle up into the docking bay from a variety of initial positions. The vehicle must also avoid obstacles.	Out of the three initial positions considered, the vehicle managed to reach the dock safely without hitting any obstacles.	*7/10*

Appendix 3.4.1

steering angle (δ)	Hitch angle (Υ)	Hitch Minimum	Hitch Max
-38	90	90	90
-37	76	75.5	85
-36	70.9	61	81
-35	62	57	69
-30	43.4	40.7	46
-25	32.7	30.5	34.8
-20	24.5	22.7	26.6
-15	17.6	15.8	19.1
-10	11.4	9.6	13
-5	5.6	4.6	7
0	0	-2	2
5	-5.6	-7	-4.6
10	-11.4	-13	-9.6
15	-17.6	-19.1	-15.8
20	-24.5	-26.6	-22.7
25	-32.7	-34.8	-30.5
30	-43.4	-46	-40.7
35	-62	-69	-57
36	-70.9	-81	-61
37	-76	-85	-75.5
38	-90	-90	-90

Appendix 5.2.1



Appendix 5.3.1

$$y(t) = 4 - 4e^{-\left(\frac{5}{4}\right)t}$$

$$\tau = 0.67\%\ of\ reference\ signal$$

$$\therefore 4(0.67) = 4 - 4e^{-\left(\frac{5}{4}\right)\tau}$$

$$solve\ for\ \tau\ \rightarrow\ \tau = -\frac{4}{5}\log\left(\frac{1}{3}\right) = 0.88s$$

Appendix 5.3.2

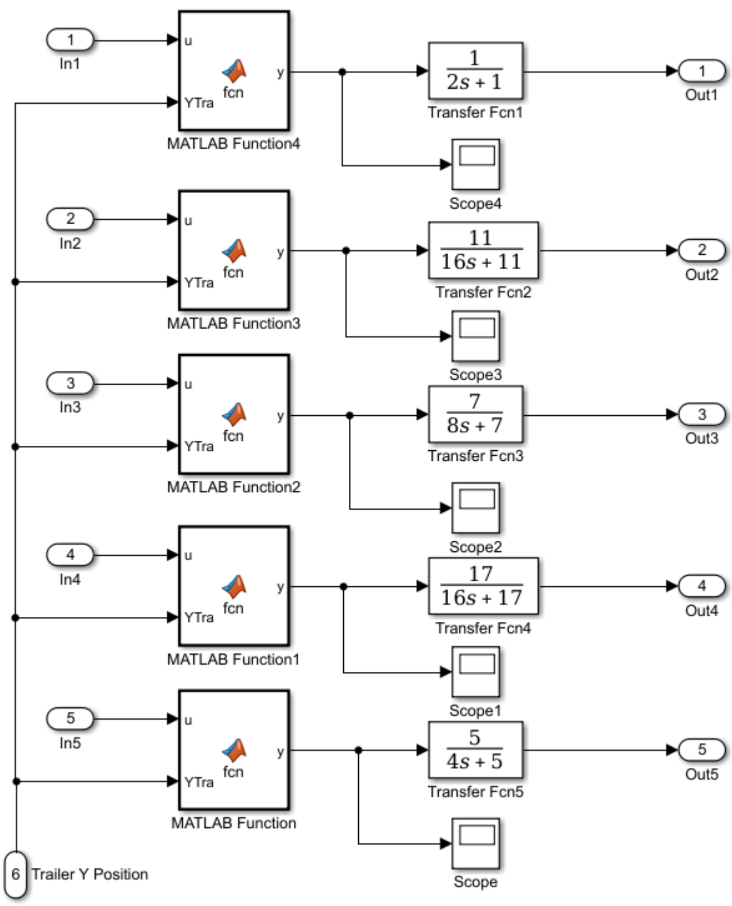
$$Average\ acceleration\colon\ \bar{a} = \frac{\int_0^t a(t)dt}{t}$$

$$from\ simulation\ it\ was\ found\ that\ it\ takes\ the\ vehicle\ 3.84s\ to\ travel\ 20\ feet.$$

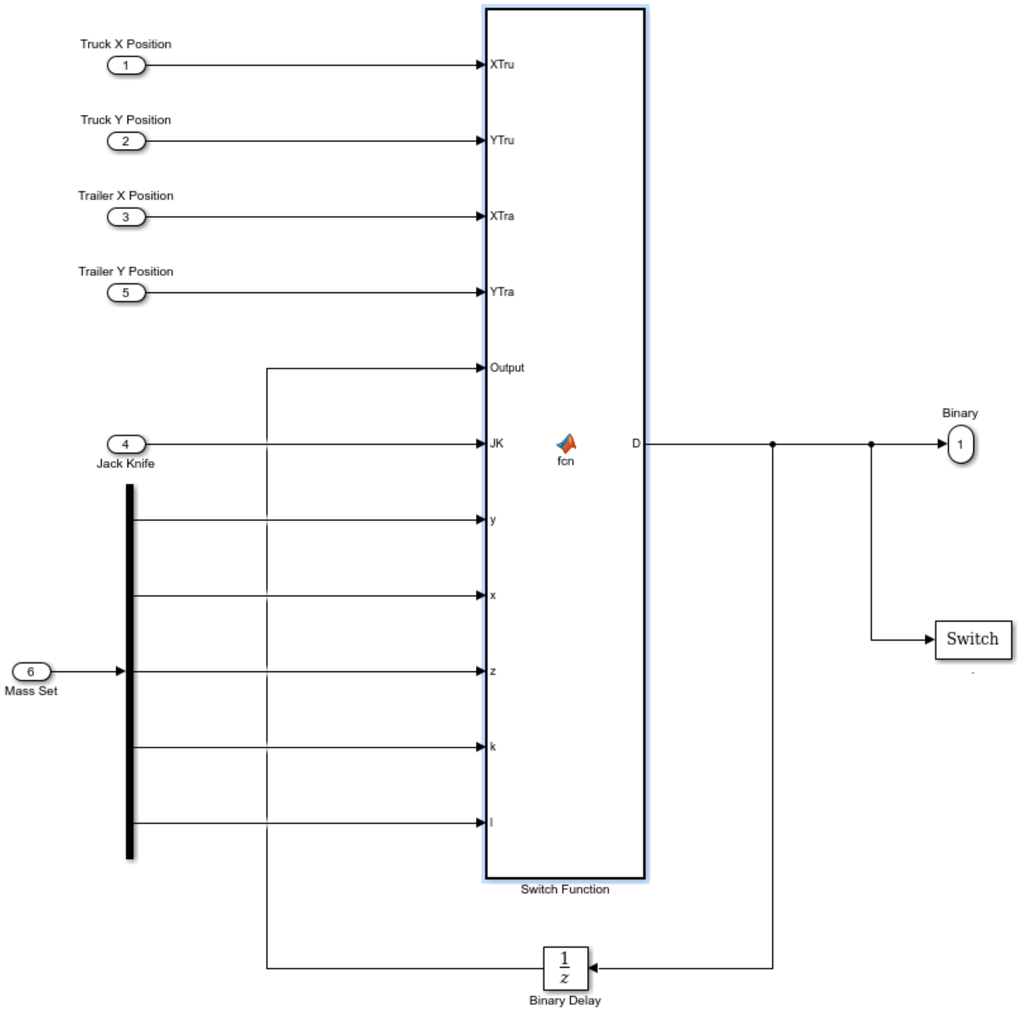
$$\bar{a} = \frac{\int_0^t a(t)dt}{t} = \frac{\int_0^{3.84} 5e^{-\left(\frac{5}{4}\right)t}}{3.84} = \mathbf{1.03}ms^{-2}$$

Appendix 6.3.3

The figure shows the transfer functions and individual codes that stop the vehicle from hitting the docking boundary.



Appendix 5.4.1



The code found in "Switch Function" can be seen below.

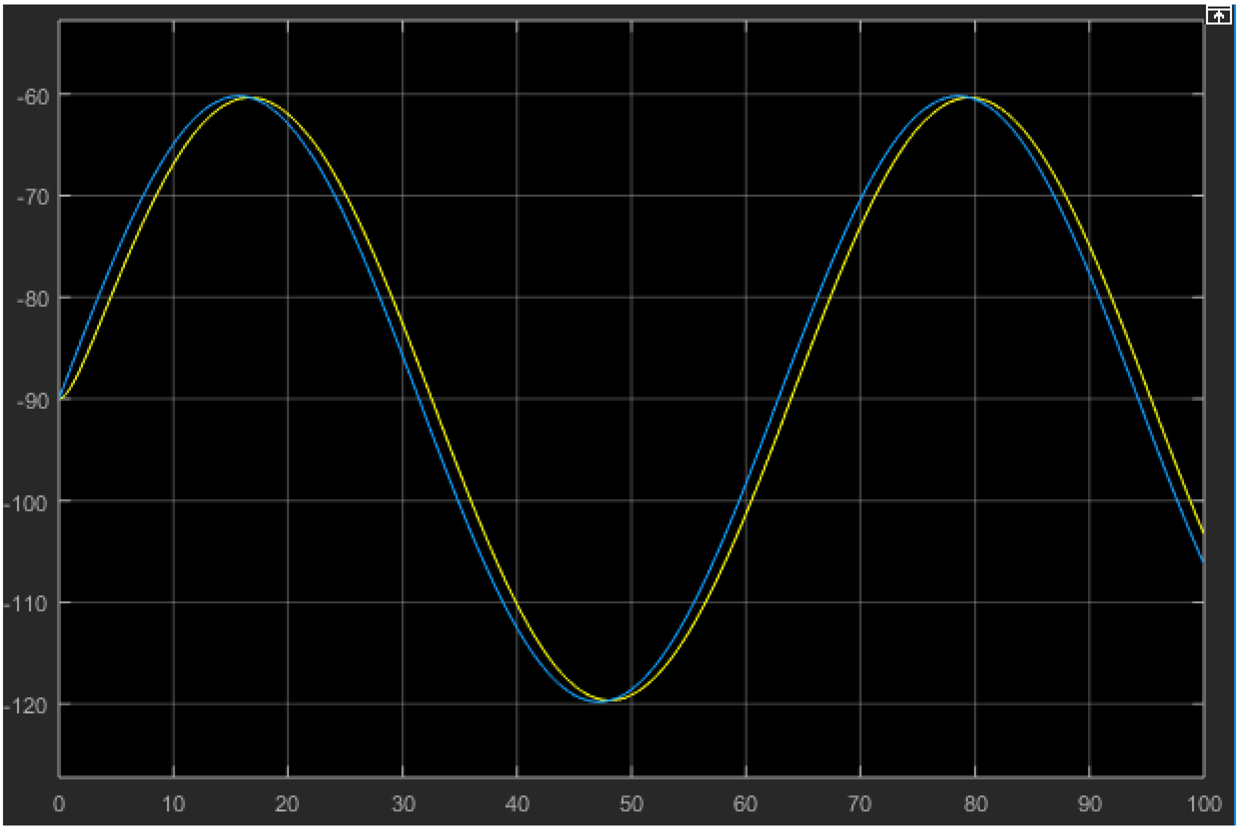
```
function D = fcn(XTru, YTru, XTra, YTra, Output, JK, y, x, z, k, l)

D = 0;

if y > -2 && x == 0 && z == 0 && k == 0 && l == 0
    if YTru <= 21.4 && XTru > 10
        D = 1;
    else
        D = 0;
    end

    if Output == 1
        D = 1;
    end

    if YTra >= 34 && (42.2 <= XTra && XTra <= 43.1)
        D = 1;
    end

    if YTra >= 32.8 && (XTra <= 42 || XTra >= 43.1)
        D = 0;
    end
```

```
    if JK == 1
        D = 0;
    end
end

if x > -2 && y == 0 && z == 0 && k == 0 && l == 0
    if YTru <= 21.2 && XTru > 10
        D = 1;
    else
        D = 0;
    end

    if Output == 1
        D = 1;
    end

    if YTra >= 34 && (42.2 <= XTra && XTra <= 43.1)
        D = 1;
    end

    if YTra >= 31.9 && (XTra <= 42 || XTra >= 43.1)
        D = 0;
    end

    if JK == 1
        D = 0;
    end
end

if z > -2 && y == 0 && x == 0 && k == 0 && l == 0
    if YTru <= 20.8 && XTru > 10
        D = 1;
    else
        D = 0;
    end

    if Output == 1
        D = 1;
    end

    if YTra >= 34 && (42.2 <= XTra && XTra <= 43.1)
        D = 1;
    end

    if YTra >= 33.4 && (XTra <= 42 || XTra >= 43.1)
        D = 0;
    end

    if JK == 1
        D = 0;
    end
end

if k > -2 && y == 0 && x == 0 && z == 0 && l == 0
    if YTru <= 20.75 && XTru > 10
        D = 1;
    else
        D = 0;
    end

    if Output == 1
```

```
        D = 1;
    end

    if YTra >= 34 && (42.2 <= XTra && XTra <= 43.1)
        D = 1;
    end

    if YTra >= 33.2 && (XTra <= 42 || XTra >= 43.1)
        D = 0;
    end

    if JK == 1
        D = 0;
    end
end

if l > -2 && y == 0 && x == 0 && k == 0 && z == 0
    if YTru <= 20.6 && XTru > 10
        D = 1;
    else
        D = 0;
    end

    if Output == 1
        D = 1;
    end

    if YTra >= 33.55 && (41.875 <= XTra && XTra <= 43.1)
        D = 1;
    end

    if YTra >= 33.55 && (XTra <= 42 || XTra >= 43.1)
        D = 0;
    end

    if JK == 1
        D = 0;
    end
end
```

Appendix 6.5.3.1

The blue curve is the reference orientation and the yellow curve is the trucks orientation.



Appendix 6.6.1

The Trajectory of the vehicle with a trailer length of 6.1m.