ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

HYBRID CONTROLLER APPROACH FOR AN AUTONOMOUS GROUND VEHICLE PATH TRACKING PROBLEM

M.Sc. THESIS

Mertcan CİBOOĞLU

Department of Control and Automation Engineering

Control and Automation Engineering Programme

DECEMBER 2016

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

HYBRID CONTROLLER APPROACH FOR AN AUTONOMOUS GROUND VEHICLE PATH TRACKING PROBLEM

M.Sc. THESIS

Mertcan CİBOOĞLU
(504141121)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. M. Turan SÖYLEMEZ

DECEMBER 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OTONOM BİR KARA ARACININ YOL TAKİBİ PROBLEMİ İÇİN HİBRİT KONTROLÖR YAKLAŞIMI

YÜKSEK LİSANS TEZİ

Mertcan CİBOOĞLU
(504141121)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. M. Turan SÖYLEMEZ

ARALIK 2016

Mertcan Cibooğlu, a M.Sc. student of ITU Graduate School of Science Engineering And Technology student ID 504141121, successfully defended the thesis/dissertation entitled "A HYBRID CONTROLLER APPROACH FOR AN AUTONOMOUS GROUND VEHICLE PATH TRACKING PROBLEM", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. M. Turan Söylemez**
Istanbul Technical University

Jury Members : **Prof. Dr. Metin Gökaşan**
Istanbul Technical University

Prof. Dr. Tankut Acarman
Galatasaray University

Date of Submission : 25 Nov 2016
Date of Defense : 23 Dec 2016

Stars are only visible in darkness…

FOREWORD

I would like to thank my supervisor, Prof. Dr. Mehmet Turan Söylemez, for sharing wealth of his knowledge and leadership. It was a great pleasure for me to take his endless support in my research

My biggest gratitude goes to my mother, Meral Cibooğlu for being with me and supporting me at every moment of my life. She always tried to do her best to give me a big support during the preparation of this thesis.

Also I would like to thank Asiye Altay for her support and for her wishes.

Finally, I also would like to thank my friends and 002, especially Mr. Yasin Çalışkan and Mr. Atakan Şahin for their support.

November 2016 Mertcan Cibooğlu

TABLE OF CONTENTS

Page

ABBREVIATIONS

CG	**:** Center of Gravity
MPC	**:** Model Predictive Control
PI	**:** Proportional-Integral
PID	**:** Proportional-Integral-Derivative

SYMBOLS

x_f, y_f	: The global coordinates of the front wheel
x, y	: The global coordinates of the rear wheel
L	: The distance between the front wheel and the rear wheel
δ	: Steering angle
ψ	: Yaw angle / Vehicle orientation
v_x	: Longitudinal velocity
F_{xf}	: Longitudinal force on front wheel
F_{yr}	: Lateral force on the rear wheel
F_{yf}	: Lateral force on the front wheel
m:	: Mass of the vehicle or slope of the line
v_y	: Lateral velocity
I_z	: Yaw inertia
L_f	: The distance between front axle to CG
L_f	: The distance between rear axle to CG
α_r, α_f	: Rear wheel and Front wheel tire side slip angles
C_r, C_f	: Combined cornering stifnesses of the front and rear tires
α	: The angle between Pure Pursuit goal point and vehicle heading
l_d	: Look ahead distance of Pure Pursuit
x_f, y_f	: Goal point coordinates of Pure Pursuit
R	: Turning radius
δ_{PP}	: Pure Pursuit desired steering angle
e	: Crosstrack error of the front wheel
k	: Track error gain
δ_{Sta}	: Stanley method desired angle
o	: Steady state look ahead offset
δ_{SSC}	: Steady State Cornering method desired angle
$d, d_{look\text{-}ahead}$	: Steady-State Cornering method look ahead distance
$d_{constant}$	: Constant distance
t_{driver}	: Reaction time of the driver
S_1	: Start point of the segment
S_2	: End point of the segment
P	: Search Point
$\vec{V}$	: Vector from S_1 to S_2
$\overrightarrow{W}$	: Vector from S_1 to S_2
x_c, y_c	: Closest Point
$x_b\, y_b, P_x\, P_y,$	: The global coordinates of the Search Point
i	: index
$i_{updated\ index}$	: Updated index
d_s	: The smallest distance
d_{max}	: Max distance
C	: Closest Point
C_p	: Updated Closest Point

$\mathbf{A,B,C,\Delta}$	**:** Solution parameters
$\mathbf{T,T_1,T_2}$	**:** Pure Pursuit goal point algoritm goal points
$\mathbf{x_f', y_f'}$	**:** Coordinates of the front wheel with respect to vehicle frame
$\mathbf{x_c', y_c'}$	**:** Coordinates of the closest point with respect to vehicle frame
$\mathbf{\theta_{diff}}$	**:** The angle difference of the path segment
$\mathbf{\delta_{Hybrşd}}$	**:** Hybrid controller desired angle
$\mathbf{t_h}$	**:** timer value of hybrid controller
$\mathbf{e_{CG}}$	**:** Crosstrack error of the CG
$\mathbf{k}$	**:** time steps

LIST OF TABLES

Page

LIST OF FIGURES

Page

HYBRID CONTROLLER APPROACH FOR AUTONOMOUS GROUND VEHICLE PATH TRACKING PROBLEM

SUMMARY

Automotive industry is the one of the most important economic sectors according to its circulation. Starting from the last part of the 18th century, the industry keeps its up-to-dateness with the keep tracking the future technology very closely. According to these development, autonomous driving function is become hot topic when the range of the automotive industry is under consideration. Autonomous function is mainly based on driving without any labor which try to reduce faults cause by the humans. Thanks to this aim, safety, comfortable and effective transportation will offer by the future self-driver.

Automated driving requires deep understanding and cooperation of many different disciplines and topics, such as sensor technologies, localization and mapping technologies, estimation and fusion algorithms, image processing algorithms, decision making and trajectory generation algorithms, vehicle controls theory and automotive engineering An ordinary driver just steers the steering wheel and apply brake or gas pedal to follow the lane and adjust the speed of the vehicle even without thinking. Nevertheless, this path following problem is under research for years as can be observed from the literature..

Path tracking methods can be divided into two main groups: Geometric and model based control methods. Geometric methods use only the geometrical relation between the path and the vehicle. In this thesis two geometric based selected Pure Pursuit and Stanley method. Then one model based method is selected as Steady State Cornering method which devoleped from linearized bicycled model. First two methods are based on the conception of an ordinary driver trying to track a given path. While the third approach is based on a simplified mathematical model of the vehicle that is trying to follow a given path. In that sense, it can be said that first two is more intuitive than the latter one, and it is obvious that each method has its own strengths and weaknesses.

In contrast, Pure-Pursuit and Steady State Cornering methods look forward in order to maneuver. For that reason, the latter two methods can preview sudden changes on the path beforehand. But on a smooth path, these two methods cut corners and their performances are not as good as Stanley

In order to use advantages of different methods at the same time, innovative approach that based on the combination of two method is proposed as hybrid controller. The proposed hybrid controller is using Pure-Pursuit and Stanley Method at the same time. A weight factor is adjusted depending on the smoothness of the path ahead. As the path gets smoother, the weight of the Stanley method is increased, if a sharp change is ahead the weight of the pure pursuit method is increased. To decide if the path is smooth or not, the look ahead strategy used in steady state method is implemented.

After that, the proposed hybrid controller and three path tracking method performances are examined with three different path sceniros. These paths are 50 meter radius circle path, rectangular shaped path and mixed of rectangular and circle path. In comparisons, in order to see effect of speed changes on methods, simulations are done with three different speed such as 20 km/h 50km/h and 80 km/h.

To sum up in this thesis, the first part will cover the literature survey on the wide range path following problem. Afterwards, the vehicle model that will be used on whole studies about the thesis will introduce at the Section-2. In Section-3, the tracking methodologies will be summarized with their mathematical backgrounds. The proposed hybrid methodology will present in Section-4. Accordingly, three different simulation studies and their comparisons will discuss in Section-5. Finally, the conclusion and planning future works on these topics will represent at the last part.

.

OTONOM KARA ARAÇLARININ YOL TAKİBİ PROBLEMİ İÇİN HİBRİT KONTROLÖR YAKLAŞIMI

ÖZET

Otomotiv endüstrisi iç döngüsünden dolayı en önemli ekonomik sektörlerden birisidir. 18. Yüzyılın sonlarından itibaren, otomotiv endüstrisi gelecek teknolojilerini yakından takip edip kendini güncellemiştir. Bu gelişmelere göre, otomobil sektörü göz önüne alındığında, otonom sürüş fonksiyonu olarak adlandırabileceğimiz otopilot sistemi en önemli konulardan birisi haline gelmiştir. Otonom olma fonksiyonu, her hangi bir insan gücü kullanmama prensibine dayanır, bu sayede insandan kaynaklı sürüş hatalarını en aza indirmek mümkün olabilmektedir. Bu amaç ile birlikte, teknolojinin gelişmesi ile gelecekte güvenli, rahat ve etkili bir seyahat, otonom araçlar tarafından sunulacaktır.

Sıradan bir sürücü yolu takip etmek için sadece direksiyon, gaz ve freni kontrol eder ve aracın hızını düşünmeden ayarlar. Yine de, yol takip sorunu, literatürden de görüleceği gibi, yıllardır araştırılmaktadır. Çünkü otonom sürüş fonksiyonu derinlemesine analiz edilirse, birçok disiplinin ve konuların iş birliğinin gerektiği görülmektedir. Örnekleyecek olursak, bu disiplin ve konular; sensör teknolojileri, konum bulma ve haritalama teknolojileri, görüntü işleme algoritmaları, karar verme ve yol oluşturma algoritmaları, araç kontrol teorileri ve otomotiv mühendisliği gibi sıralanabilir.

Yol takip metotları geometrik ve model tabanlı takip metotlar olmak üzere iki ana gruba ayrılabilir:. Geometrik metotlar yol takibi için sadece araç ve yol arasındaki geometrik ilişkiyi kullanılır. Bu metodlar için araç dinamikleri önemsizdir. Bu tezde incelenmek üzere *'Pure Pursuit'* ve *'Stanley Metod'*. olarak iki geometrik tabanlı metod seçildi. Ardından da bir tane model tabanlı metod olan *'Steady State Cornering'* seçildi. Bu metod doğrusallaştırılmış bisiklet modelinden geliştirilmiştir. İlk iki metot sıradan bir sürücünün verilen bir yolu takip etmesi prensibine dayandırılır, bu açıdan ilk iki metot sonraki metoda göre daha sezgiseldir. Ancak model tabanlı metodun tasarımı için model parametrelerinin bilinmesi gerekmektedir.

Model tabanlı yöntemin daha iyi açıklanması için tezin ikinci bölümü araç modelleme üzerine oluşturuldu. Bir araç ön iki tekerleğin ve arka iki tekerleğinin doğrusal eksen hizasında tek tekerleğe indirilerek bisiklet şekline indirgenebilmektedir. Literatürde bisiklet modeli olarak geçen bu model kinematik, dinamik ile doğrusal ve doğrusal olmayan bisiklet modeli olarak yer almaktadır. Bu modellerden kinematik model ile dinamik model arasındaki en temel fark şudur; kinematik model tekerlekteki kayma etkilerini dikkate almaz ama dinamik modelde kaymadan dolayı oluşan yatay kuvvet ve ivme hesaba katılır. Araç modellerinin çıkarımı bölümünün sonunda dinamik bisiklet modelinin küçük açılar yaklaşımı ile doğrusallaştırılmış hali de verilmiştir. Ardından yol takip modelleri anlatılmıştır.

Yol takip algoritmalarından *'Pure Pursuit'* ve *'Steady State Cornering'* metotları manevra yapabilmek için ileri bakarlar ancak *'Stanley'* metodu ön tekerleğe en yakın yolun durumuna bakar. Yani bir nevi araç hizasını kontrol etmektedir. Bu sebepten dolayı, *'Pure Pursuit'* ve *'Steady State Cornering'* yöntemleri ani değişimleri önceden algılayabilirler ve manevra yapabilmek için daha önceden harekete geçerler. Ancak, viraj derecesi düşük ve/veya düzgün bir yolda bu iki metot köşeleri keser ve performansları *'Stanley'* metoduna göre daha düşüktür. Seçilen bu üç ayrı metod tezin üçüncü bölümünde detaylı bir şekilde anlatılmaktadır. Ayrıca bu bölümde, bu algoritmaların uygulanma aşamasında kullanılacak bazı önemli hesaplama algoritmaları detaylıca anlatılmıştır. Bunlar; belirlenmiş bir noktaya yol üzerinde en yakın nokta hesabı, *Pure Pursuit* metodu için gerekli olan hedef noktası hesabı ve yola olan uzaklık hesapları için gerekli algoritmalardır.

Bu tez çalışmasının ana konusu olarak yenilikçi bir yaklaşım olan iki farklı metodun birleşimi olarak hibrid kontrolör tasarımı önerildi. Önerilen tasarımda bahsedilen farklı yol takip yöntemlerinin güçlü taraflarını aynı anda kullanıldı. Bu önerilen hibrid kontrolör temel olarak *Pure Pursuit* ve *Stanley* metotlarını kullanır. *Pure Pursuit* yolun ilerisine bakarak önceden dönmeye başlamaktadır, *Stanley* ise ön tekere en yakın yol bilgisine göre hareket ettiği için dönüşleri önceden göremez ve sert dönüşlerde yoldan daha çok uzaklaşır. Temel amaç sert olmayan virajlarda *Stanley* metodunun performansı ile , sert dönüşlerdeki *Pure Pursuit* metodunun performanslarını birleştirmektir. Bunu yapmak için iki metod aynı anda çalışmaktadır ve direksiyon işaretleri belirli ağırlıklar ile çarpılarak toplanır ve hibrit kontrolörün direksiyon işareti elde edilir. Burada temel nokta, virajların eğriliğine göre bu kontrol işaretlerinin ağırlıklarının değiştirilmesidir. Eğer sert dönüş var ise, kontrol işareti olarak *Pure Pursuit* in ağırlığı arttırılır, sert dönüş yoksa, yol takibi performansı çok iyi olduğu için *Stanley* metodun kontrol işareti ağırlıklandırılır. Kontrolör ağırlıklarını belirlemekte kullanılan yolun düzlük durumuna karar verebilmek için Steady State metotta kullanılan ileri yol stratejisi uygulanmıştır. Bu strateji, aracın boylamasına hizasından belirli mesafe ileriye bakarak, bu noktaya en yakın yolun değişimine bakar.

Hibrit kontrolör tasarımından sonra, önerilen hibrid kontrolör ve 3 farklı yol takip metotlarının performansları üç farklı yol senaryosunda test edildi. Bu yollar; 50 metre çapında daire, dikdörtgen ve karmaşık yol dizaynı şeklindedir. Daire şeklinde yol, sert dönüş olmayan bir yoldaki performansları ölçerken, dikdörtgen şeklindeki yol sahip olduğu 90 derece dönüşler sayesinde sert virajları test etmek için kullanılmıştır. Karışık yol senaryosu ise, hem sert olmayan hem de sert virajlara karşı yol takip algoritmalarının performanslarını dengeli bir şekilde test etmek için tasarlanmıştır. Farklı hızların metodlarının performanslarını nasıl etkilediğini görmek amacı ile bütün senaryolar düşük hız orta hız ve yüksek hız olmak üzere üç farklı hız ile test edilmiştir. Bu hızlar, 20 km/h, 50km/h ve 80km/h olarak belirlenmiştir. Performansların karşılaştırılması olarak ise L1 ve L2 hata normları kullanılmıştır. L1 hata normu hataların toplamının ortalaması olarak hesaplanmış, L2 hata normu , bir diğer adı ile öklit normu, hataların karelerinin toplamının karekökü olarak hesaplanmıştır. Sonuçlara bakıldığında hibrit kontrolör L1 normuna göre dikdörtgen yol ve karışık yol senaryolarında en iyi sonucu vermiştir. Ayrıca grafikler incelendiğinde önerilen metodun, sert virajlarda diğer metodlara göre çok daha hızlı yol takibine devam ettiği görülmektedir.

Özetlenecek olursa, bu tezin ilk bölümünde yol takibi problemi ile ilgili geniş çaplı bir literatür araştırması yapılmıştır. Günümüz otomotiv dünyasında hali hazırda firmaların çabaları, değişik yöntemler ve bu alanda yapılan çalışmaların artarak devam ettiği

gösterilmiştir. Daha sonrasında Bölüm 2'de araç simülasyonlarda kullanılan araç modelleri incelenmiş ve detaylıca modeller arası farklar incelenmiştir. Sonrasında bu tez çalışması için seçilmiş olan yol takibi algoritmaları matematiksel olarak ele alınmıştır. Bölüm 4'e gelindiğinde ise önerilen yaklaşım olan hibrit metodun ortaya çıkışı uygulaması ve kalibrasyonu hakkında bilgiler yer almaktadır. Bütün metodlar toparlandıktan sonra Bölüm 5'te, metodların performanslarını test etmek üzere hazırlanmış pistler, hata kriterleri ve simülasyon sonuçlarına yer verilmiştir. Son bölümde ise bütün çalışma özetlenmiş, gelecekte ne tür geliştirmeler yapılabileceği hakkında bilgiler sunulmuştur.

1. INTRODUCTION

Automotive industry is the one of the most important economic sectors according to its circulation. Starting from the last part of the 18th century, the industry keeps its up-to-dateness with the keep tracking the future technology very closely [1]. According to these development, autonomous driving function is become hot topic when the range of the automotive industry is under consideration. Autonomous function is mainly based on driving without any labor which try to reduce faults cause by the humans. Thanks to this aim, safety, comfortable and effective transportation will offer by the future self-driver [2].

First autonomous car idea suggested very earlier dates of the automotive industry. However, the first self-sufficient prototype of the automotive are appeared in the 1980s by the collaboration of Carnegie Mellon University and ALV in 1984 [3,4], and Mercedes-Benz and University of Munich in 1987 [4]. Since then, the projects about the automaticity is increase day by day.

It is well-known information that the big automotive companies have several successfully projects or trials in that days. Tesla, one of the most innovative car producer with its reliable technologies, has tried its highway autopilot at 20th of October in 2016 which was quite close update to our literature [5]. This was not their first attempt. The company has ongoing software program for their cars which includes self-parking, road follow. It has need to be assigned by driver only on limited-access highways or quite inhabitant locations. Traffic signals or stop signs, and pedestrian or cyclist's actions are the next problems of the project which needs to be solved for fully self-drive experience.

Google, the biggest internet company in our era, has also announced their accident-free 500.000 kilometers with their autonomous car in August 2012. This successfully story has ended in February 2016 with their first obstacle avoidance and traffic light crashes. Nevertheless, they have still ongoing progress on their project [6].

Uber, the public transportation company as globally, and its owned initiative Otto sent their autonomous truck on a two-hour 200 kilometers delivery, successfully [7]. In addition to these progress, Audi, Ford, Mobileye and Delphi, Mercedes-Daimler, Volvo have their unique project in this area. They aim the Level-5 autonomous devices which is the car that can handle all driving tasks and go anywhere. However, there is no Level-3 car which will allows to take control as itself [8]. That's why the autonomous car idea is extremely open to innovation in this days.

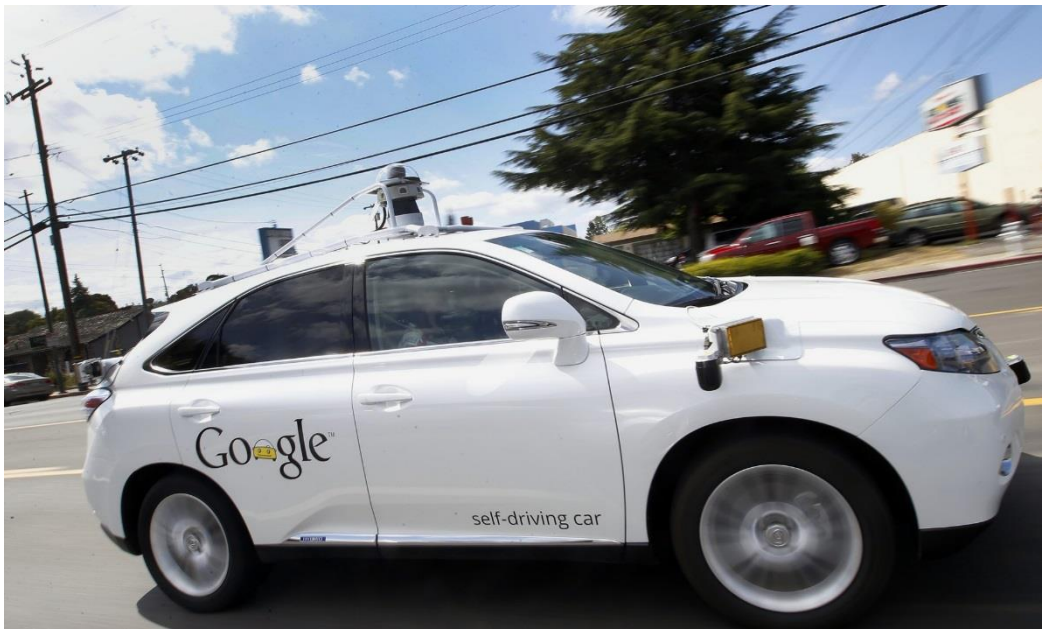

Figure 1.1 : Google's Self Driving Car [6].

Automated driving requires deep understanding and cooperation of many different disciplines and topics, such as sensor technologies, localization and mapping technologies, estimation and fusion algorithms, image processing algorithms, decision making and trajectory generation algorithms, vehicle controls theory and automotive engineering.

An ordinary driver just steers the steering wheel and apply brake or gas pedal to follow the lane and adjust the speed of the vehicle even without thinking. Nevertheless, this path following problem is under research for years as can be observed from the literature. According to Scopus, the published paper about path tracking and following are shown with rising slope which can also be proof of the popularity in academic side[9].

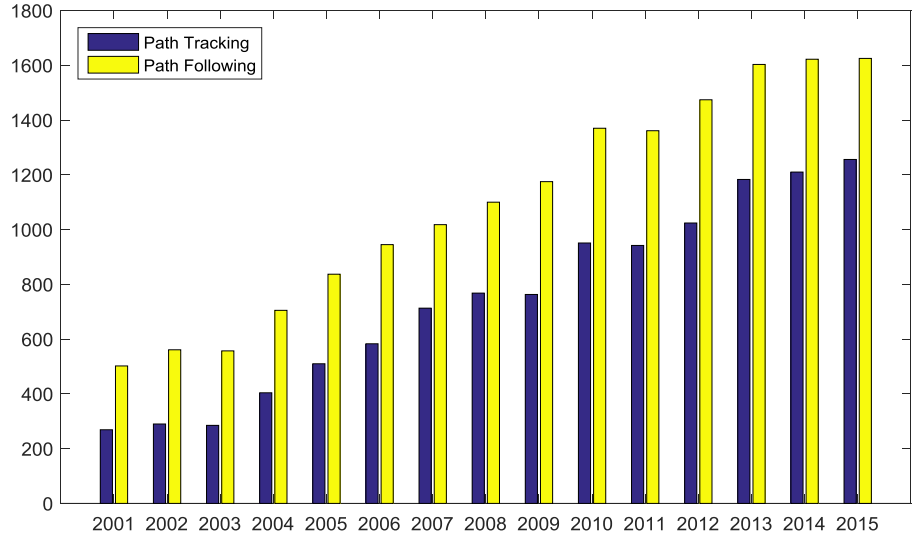

Figure 1.2 : Scopus datas about "path tracking" and "path following"[9].

Path tracking methods can be divided into two main groups: Geometric and model based control methods. Geometric methods use only the geometrical relation between the path and the vehicle. The most popular geometric methods include pure pursuit, vector pursuit and Stanley method [10]. Geometric methods are better at disturbance rejection in general. However they are relatively problematic at high speeds [10, 11]. This problem is tried to be overcome by proposing adaptive look forward methods or gain scaling.

Each methods has its own weak points, while the pure pursuit methods starts to cutting corners at high speed[12], Stanley method overshoots from the path at sharp corners[11]. Vector control, which is another geometric control method, depends on the screw theory [13]. It is similar to pure pursuit method but it also considers the target orientation at the look ahead point.

Model based methods uses the vehicles mathematical model to decide the control strategy. In the literature almost all classical control methods are proposed for solving the vehicle trajectory control problem. Kinematic, dynamic linear and nonlinear models are employed in several works. They are relatively complicated with respect to geometrical methods.

A method that is combining the look ahead type control with steady state cornering vehicle model is proposed by Jalali et al. [14]. In this method a steering angle, under the steady state dynamic steering condition, that will bring the vehicle to target position

is calculated. In another work, gain scheduled PID controller is proposed. Scheduling of the gains is determined by minimizing lateral deviation and heading error with respect to target path [15].

A kinematic vehicle model based controller is developed in a dissertation [11]. However, the method includes a complicated mathematical derivation that is hard to implement. A dynamical model based controller called optimal preview control is also proposed by Snider with favorable results [11].

Some fuzzy logic based control strategies are also proposed in the literature. A control strategy is proposed by P. Ping et al. such that the cross-track error is compensated by a PI controller and the yaw angle error is compensated with a fuzzy logic controller [16]. In another work a full state feedback controller method is supported by a fuzzy logic based gain scheduler to compensate parameter changes in the model, and implementation on a real vehicle is considered [17].

Model Predictive Control (MPC) methods is one of the most popular methods in the literature. Path tracking problem very well complies with the model predictive control strategy. Reference, which is the trajectory, is predetermined in most of the cases and a detailed vehicle model is very effective at state estimation. A combined longitudinal and lateral control structure with a nonlinear MPC is proposed by Song et al. [18]. Another nonlinear MPC method is proposed for path following and collision avoidance by Braghin et al. [19]. A robust learning based model predictive control is proposed to decouple robustness and optimal performance by introducing an online learning algorithm to traditional MPC algorithm [20].

In this thesis, an innovative approach that based on the combination of Pure-Pursuit and and Stanley Method is proposed for the path following problem. According to following parts of thesis, the first part will cover the literature survey on the wide range path following problem. Afterwards, the vehicle model that will be used on whole studies about the thesis will introduce at the Section-2. In Section-3, the tracking methodologies will be summarized with their mathematical backgrounds. The proposed hybrid methodology will present in Section-4. Accordingly, three different simulation studies and their comparisons will discuss in Section-5. Finally, the conclusion and planning future works on these topics will represent at the last part.

2. VEHICLE MODEL

In this section the vehicle model that is used in the simulations is described. In order to model vehicle lateral dynamics, vehicle front and rear wheels can be presented as one wheel such as in Figure 2.1. This method is so called bicycle method. A bicycle model can be kinematic, dynamic, linear or nonlinear.

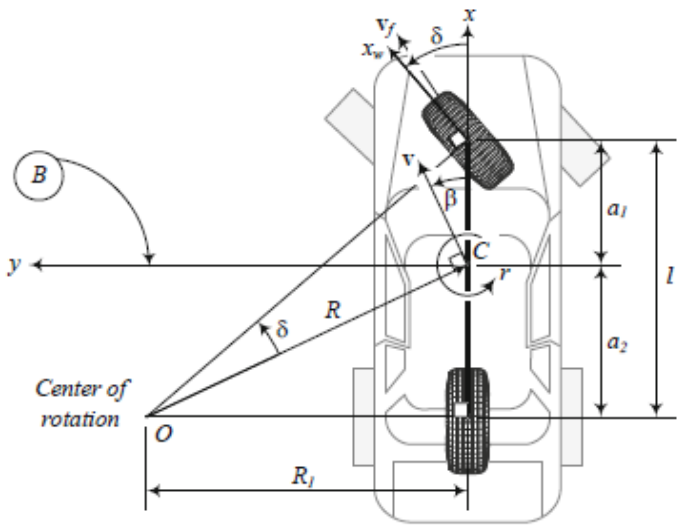

Figure 2.1 : Bicycle model representation [22].

Tire slip effect is the main difference between kinematic and dynamic model. Kinematic model is derived from assumption of zero slip on tires.

2.1 Kinematic Bicycle Model

In the literature [11, 13]. the equations of motion for the kinematic bicycle model of vehicle is given. In the kinematic bcycle model, the left and right wheels are merged at the centre of the vehicle, and front and rear axles are kept as it is shown in Figure 2.2.

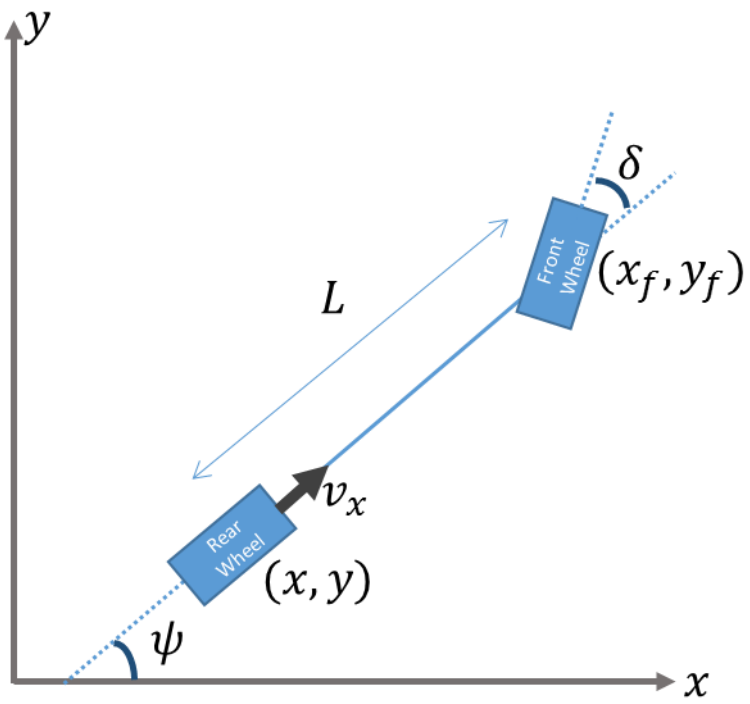

Figure 2.2 : Kinematic Bicycle Model.

In this model wheels are assumed to have no tire slip and only the front wheel is steerable. Restricting the model to motion in a plane, the nonholonomic constraint equations for the front and rear wheels are:

$$\dot{x}_f \sin(\delta + \psi) - \dot{y}_f \cos(\delta + \psi) = 0 \tag{2.1}$$

$$\dot{x} \sin(\psi) - \dot{y}\cos(\psi) = 0 \tag{2.2}$$

Where (x, y) is the global coordinate of the rear wheel, (x_f, y_f) is the global coordinate of the front wheel, ψ is yaw angle of the vehicle in global frame, and δ is the steering angle. As the front wheel is located at distance L from the rear wheel along the orientation of the vehicle, (x_f, y_f) may be calculated as:

$$x_f = x + Lcos(\psi) \tag{2.3}$$

$$y_f = y + Lsin(\psi) \tag{2.4}$$

Eliminating (x_f, y_f) from equation 2.1 will give

$$0 = \frac{\partial\big(x + Lcos(\psi)\big)}{\partial t} \sin(\delta + \psi) - \frac{\partial\big(y + Lsin(\psi)\big)}{\partial t} \cos(\delta + \psi) \tag{2.5}$$

$$0 = \left(\dot{x} - \dot{\psi}Lsin(\psi)\right)\sin(\delta + \psi) - (\dot{y} + \dot{\psi}Lcos(\psi))\cos(\delta + \psi) \tag{2.6}$$

$$\begin{aligned} 0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) - \dot{\psi}Lsin(\psi)\sin(\delta + \psi) \\ - \dot{\psi}Lcos(\psi)\cos(\delta + \psi) \end{aligned} \tag{2.7}$$

If the trigonometric terms expands:

$$\begin{aligned} 0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) \\ - \dot{\psi}Lsin(\psi)(\sin(\psi)\cos(\delta) + \sin(\delta)\cos(\psi)) \\ - \dot{\psi}Lcos(\psi)(\cos(\psi)\cos(\delta) - \sin(\delta)\sin(\psi)) \end{aligned} \tag{2.8}$$

$$\begin{aligned} 0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) \\ - \dot{\psi}L(\sin^2(\psi)\cos(\delta) + \sin(\psi)\sin(\delta)\cos(\psi)) \\ - \dot{\psi}L(\cos^2(\psi)\cos(\delta) - \cos(\psi)\sin(\delta)\sin(\psi)) \end{aligned} \tag{2.9}$$

$$\begin{aligned} 0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) \\ - \dot{\psi}L(\sin^2(\psi)\cos(\delta)) - \dot{\psi}L\,(\cos^2(\psi)\cos(\delta)) \\ - \dot{\psi}L(\cos(\psi)\sin(\delta)\sin(\psi)) \\ + \dot{\psi}\mathrm{L}(\cos(\psi)\sin(\delta)\sin(\psi)) \end{aligned} \tag{2.10}$$

$$0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) - \dot{\psi}L\cos(\delta)\,(\sin^2(\psi)) + \cos^2(\psi)) \tag{2.11}$$

$$0 = \dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) - \dot{\psi}L\cos(\delta) \tag{2.12}$$

The nonholonomic constraint on the rear wheel, equation 2.2, is satisfied by $\dot{x} = \cos(\psi)$ and $\dot{y} = \sin(\psi)$ and scalar corresponds to the longitudinal velocity v_x , such that:

$$\dot{x} = v_x\cos(\psi) \tag{2.13}$$

$$\dot{y} = v_x\sin(\psi) \tag{2.14}$$

Applying this to the constraint on the front wheel, equation 2.1 gives a solution for the yaw rate, $\dot{\psi}$:

$$\dot{\psi} = \frac{\dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi)}{L\cos(\delta)} \tag{2.15}$$

$$\dot{\psi} = \frac{v_x \cos(\psi)\sin(\delta+\psi) - v_x \sin(\psi)\cos(\delta+\psi)}{L\cos(\delta)} \tag{2.16}$$

If trigonometric terms expands again

$$\dot{\psi} = \frac{v_x \cos(\psi)\,(\sin(\psi)\cos(\delta) + \cos(\psi)\sin(\delta))}{L\cos(\delta)} - \frac{v_x \sin(\psi)\,(\cos(\psi)\cos(\delta) - \sin(\psi)\sin(\delta))}{L\cos(\delta)} \tag{2.17}$$

$$\dot{\psi} = \frac{v_x(\sin(\psi)\cos(\delta)\cos(\psi) + \cos^2(\psi)\sin(\delta))}{L\cos(\delta)} - \frac{v_x(\sin(\psi)\cos(\psi)\cos(\delta) - \sin^2(\psi)\sin(\delta))}{L\cos(\delta)} \tag{2.18}$$

$$\dot{\psi} = \frac{v_x((\sin^2(\psi) + \cos^2(\psi))\sin(\delta))}{L\cos(\delta)} - \frac{v_x(\sin(\psi)\cos(\psi)\cos(\delta) - \sin(\psi)\cos(\delta)\cos(\psi))}{L\cos(\delta)} \tag{2.19}$$

$$\dot{\psi} = \frac{v_x \sin(\delta)}{L\cos(\delta)} = \frac{v_x \tan(\delta)}{L} \tag{2.20}$$

So we can summarize kinematic bicycle model equations with equations 2.21, 2.22 and 2.23

$$\dot{x} = v_x \cos(\psi) \tag{2.21}$$

$$\dot{y} = v_x \sin(\psi) \tag{2.22}$$

$$\dot{\psi} = \frac{v_x \tan(\delta)}{L} \tag{2.23}$$

2.2 Dynamic Bicycle Model

The model introduced in the kinematic bicycle model section can be extended to include dynamics. Lateral forces on the vehicle are of primary concern in dynamic bicycle model shown in Figure 2.3.

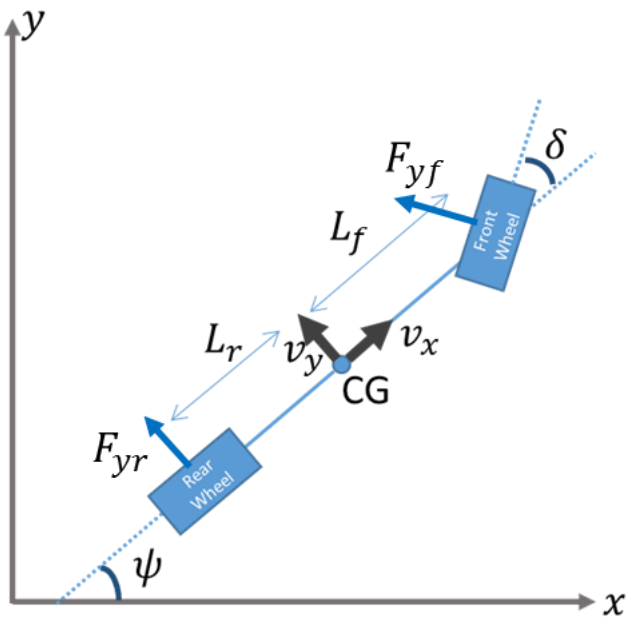

Figure 2.3 : Dynamic Bicycle Model.

For this section, longitudinal velocity is assumed to be controlled. Therefore assumption of constant velocity, $\dot{v}_x = 0$, allows the simplification:

$$F_{xf} = 0 \tag{2.24}$$

Summing the lateral forces illustrated in Figure 2.3 is giving lateral acceleration:

$$m\dot{v}_y = F_{yf}\cos(\delta) + F_{yr} + mv_x\dot{\psi} \tag{2.25}$$

Considering only motion in the plane, a center of gravity CG along the center line of the vehicle, and yaw inertia I_z, balancing the yaw moments gives:

$$I_z\ddot{\psi} = L_f F_{yf}\cos(\delta) - L_r F_{yr} \tag{2.26}$$

where $\ddot{\psi}$ is the angular rate about the yaw axis. L_r and L_f are the distances of the front and rear axles to the vehicle center of gravity. Tire side slip angles for the front and rear wheels can be derived as

$$\alpha_f = \tan^{-1}\left(\frac{v_y + L_f\dot{\psi}}{v_x}\right) - \delta \tag{2.27}$$

$$\alpha_r = \tan^{-1}\left(\frac{v_y - L_r\dot{\psi}}{v_x}\right) \tag{2.28}$$

Assuming small tire slips, the force generated by the wheels is linearly proportional to the slip angle, the lateral forces are defined as

$$F_{yf} = -C_f \alpha_f \tag{2.29}$$

$$F_{yr} = -C_r \alpha_r \tag{2.30}$$

where C_f and C_r are the combined cornering stiffnesses of the front and rear tires, respectively. If all equations are summarized:

$$\begin{aligned} m\dot{v}_y = -C_f \left[\tan^{-1}\left(\frac{v_y + L_f\dot{\psi}}{v_x}\right) - \delta\right]\cos(\delta) \\ - C_r\left[\tan^{-1}\left(\frac{v_y - L_r\dot{\psi}}{v_x}\right)\right] + m v_x \dot{\psi} \end{aligned} \tag{2.31}$$

$$\begin{aligned} I_z\ddot{\psi} = -L_r C_f \left[\tan^{-1}\left(\frac{v_y + L_f\dot{\psi}}{v_x}\right) - \delta\right]\cos(\delta) \\ + L_r C_r\left[\tan^{-1}\left(\frac{v_y - L_r\dot{\psi}}{v_x}\right)\right] \end{aligned} \tag{2.32}$$

Hence velocity of the vehicle in global coordinates can be found as follows:

$$\dot{x} = v_x \cos(\psi) - v_y \sin(\psi) \tag{2.33}$$

$$\dot{y} = v_x \sin(\psi) + v_y \cos(\psi) \tag{2.34}$$

In this thesis below parameter values are used for all models.

Table 2.1 : Dynamic Bicycle Model Parameters.

Parameter	Value	Unit
m	1400	kg
C_r	133756.05	N/rad
C_f	130756.05	N/rad
I_z	2000.24	kgm^2
L_f	1.08	m
L_r	1.620	m

Besides, lateral forces, F_{yr} and F_{yf}, are kept within 8000 N and -8000 N range for realistic results. In Figure 2.4 and 2.5, dynamic bicycle model results and kinematic bicycle model are compared. As it can be seen at the low speed, two models have similar results. On the other hand, due to slip forces acting on vehicle lateral motion in high speeds, the difference in between two models increased.

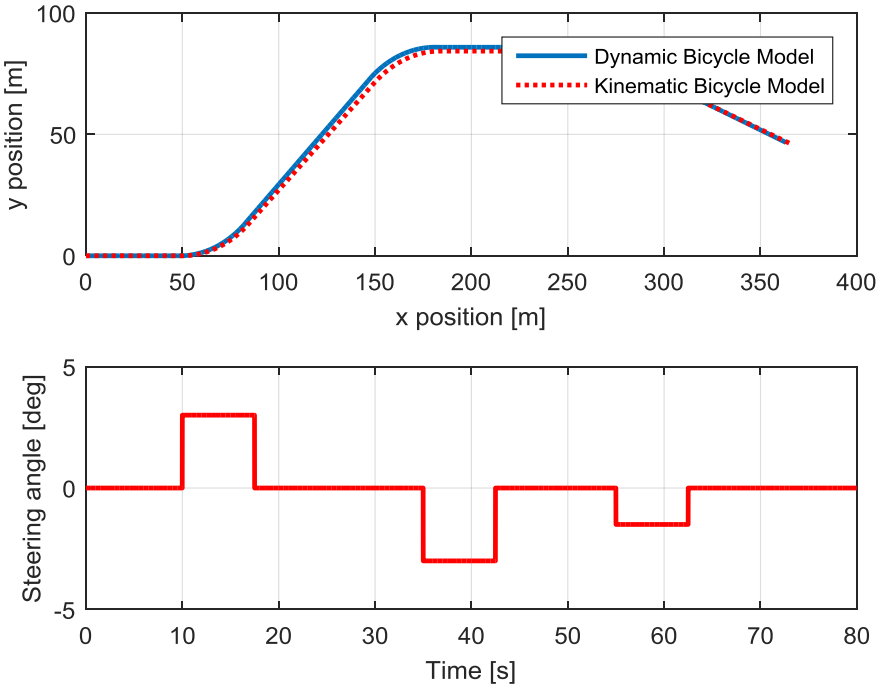

Figure 2.4 : Vehicle CG position change with 18 km/h.

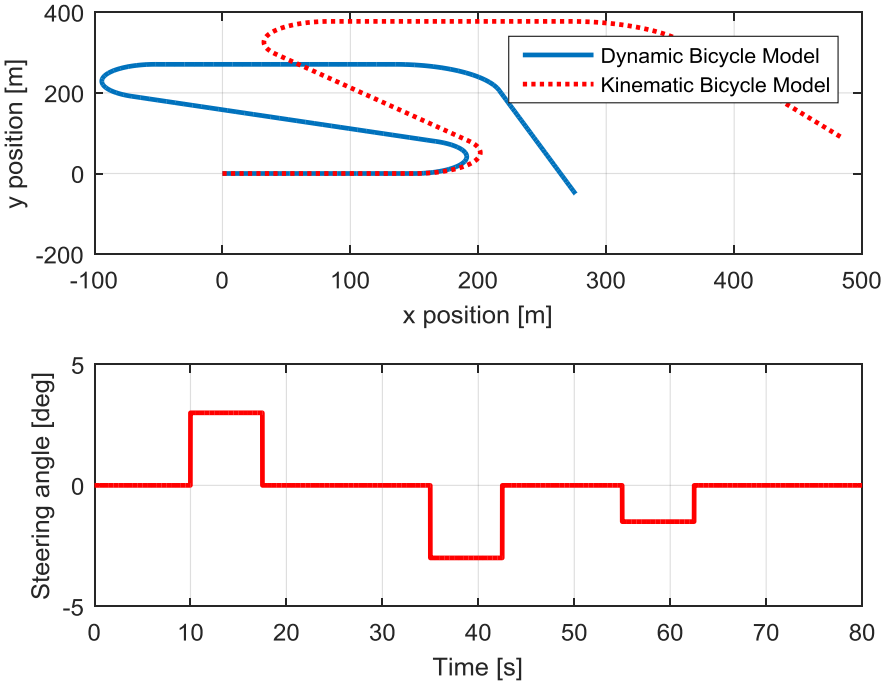

Figure 2.5 : Vehicle CG position changes with 54 km/h.

2.3 Linearized Dynamic Bicycle Model

For some applications or controllers, linearized vehicle model is rather suitable. Application of small angle assumptions to equations 2.25 and 2.26 gives [11]:

$$m\dot{v}_y = \frac{-C_f v_y - C_f L_f \dot{\psi}}{v_x} + C_f \delta + \frac{-C_r v_y + C_r L_r \dot{\psi}}{v_x} - m v_x \dot{\psi} \tag{2.35}$$

$$I_z \ddot{\psi} = \frac{-L_f C_f v_y - L_f^2 C_f \dot{\psi}}{v_x} + L_f C_f \delta + \frac{L_r C_r v_y - L_r^2 C_r \dot{\psi}}{v_x} \tag{2.36}$$

The linearized bicycle model can be used for simulations, but in order to get realistic results, the dynamic bicycled model is employed in non-linear form.

3. PATH TRACKING METHODS

In this part, three different path tracking algorithms are explained in detail. First two methods are based on the conception of an ordinary driver trying to track a given path. While the third approach is based on a simplified mathematical model of the vehicle that is trying to follow a given path. In that sense, it can be said that first two is more intuitive than the latter one, and it is obvious that each method has its own strengths and weaknesses.

3.1 Pure Pursuit Method

Pure pursuit is one of the most commonly used methods in a path tracking problem because of its simplicity and good performance. The main idea behind the pure pursuit method is to always 'look forward' for a target point on the trajectory and find the steering angle that makes the vehicle to pass through that target point. Hence, that translates the whole path tracking problem into a simple geometry problem. In this method tire slip is assumed to be zero and it is assumed that the tires fully comply with the non-holonomic constraints. Schematic view of the pure pursuit method is presented in Figure 3.1. Here, α is the angle between the vehicle heading and the line connecting vehicle's rear axle to the target point. l_d is the distance between vehicle's rear axle and the target point (Figure 3.1, yellow line), L is the wheelbase of the vehicle.

Applyig the law of sines to Figure 3.1 results in

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)} \tag{3.1}$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)} \tag{3.2}$$

$$\frac{l_d}{2\sin(\alpha)} = R \tag{3.3}$$

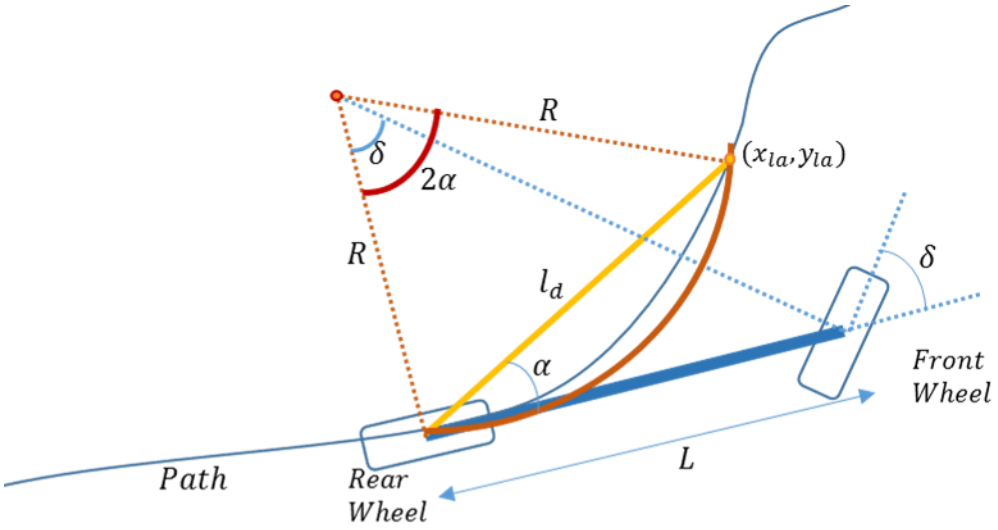

Figure 3.1 : Geometric explanation of Pure-Pursuit Method.

To simplify the calculations in geometric path tracking, four-wheeled vehicle converted into two-wheeled one, a rear and a front wheels, i.e. bycyle. Another assumption is that the vehicle moves only on a plane.

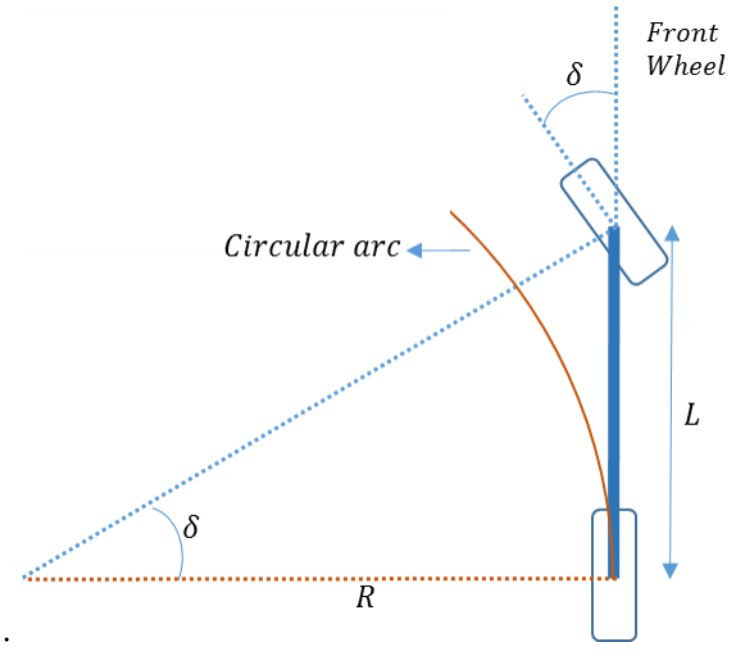

Figure 3.2 : Geometric relationship between steering angle and the vehicle motion.

As illustrated in Figure 3.2, this simple geometric relationship can be written as:

$$\tan(\delta) = \frac{L}{R} \tag{3.4}$$

where L is the distane between the front wheel and rear wheel, δ is the steering angle, and R is the turning radius of the circle. At low speeds, this model gives the closest model of the motion of a car. If equations 3.3 and 3.4 solves together:

$$\tan(\delta) = \frac{L}{\frac{l_d}{2\sin(\alpha)}} = \frac{2Lsin(\alpha)}{l_d} \tag{3.5}$$

With the help of this, the desired steering angle, $\delta_{pp}(t)$, which makes the vehicle achieve the target point can be found as:

$$\delta_{PP}(t) = \tan^{-1}\left(\frac{2Lsin\big(\alpha(t)\big)}{l_d}\right) \tag{3.6}$$

In addition, pure pursuit algorithm can be described with the following routine [21]

1. Find the current location and orientation of the vehicle in the global coordinate system
2. Find the point that is on the track and closest to the vehicle's rear axle
3. Search for the target point (x_{la}, y_{la}) that is on the track and at a distance l_d from the vehicle's rear axle.
4. Transform the target point coordinates defined in global coordinates to vehicle coordinates.
5. Calculate the angle $\alpha(t)$ and then determine the steering angle using equation 3.3.
6. Apply the steering angle found in step 5 and return to step 1 in next time step.

As it can be seen, pure pursuit method has only one parameter to be tuned, that is the look ahead distance, (l_d). In that sense, the look ahead distance behaves like a proportional gain [22]. If the look ahead distance is kept small, the vehicle tends to track the trajectory more precisely but control signal i.e., steering angle, changes rapidly and that can cause oscillations in the response [11]. If it is kept large, the response becomes smoother but large corner cuttings may be seen which decreases the

tracking quality and safety in some cases. These cases are illustrated in Figure 3.3. The look ahead distance can be tuned according to path geometry and velocity of the vehicle as studied in the previous works [10, 11, 22]. Decreasing the look ahead distance as path curvature increases and increasing it as path curvature decreases can be an adaptation to path geometry.

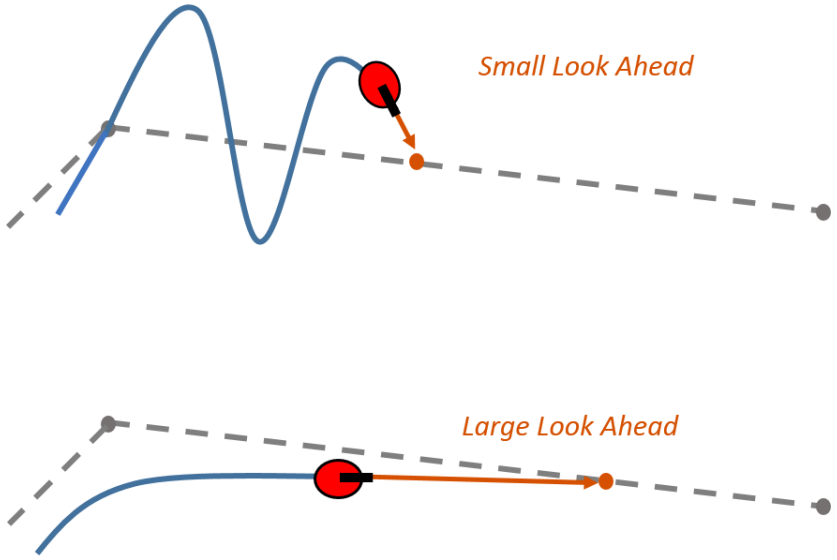

Figure 3.3 : Different Look ahead distance Results on tracking [23].

3.2 Stanley Method

Stanley method is first used in 2005 DARPA Grand Challenge by Stanford University in their autonomous vehicle called Stanley [24]. In this method, instead of "looking forward", a control strategy that is using the cross track error angular and orientation error with respect to the closest point on the trajectory is used. Geometric representation of Stanley is presented in Figure 3.4.

The steering angle due to Stanley method is given as in equation 3.7 [24].

$$\delta_{Sta}(t) = \psi(t) + \tan^{-1}\left(\frac{ke(t)}{v_x(t)}\right) \tag{3.7}$$

Here ψ is the orientation error, k is the track error gain, e is the track error and v_x is the vehicle longitudinal speed.

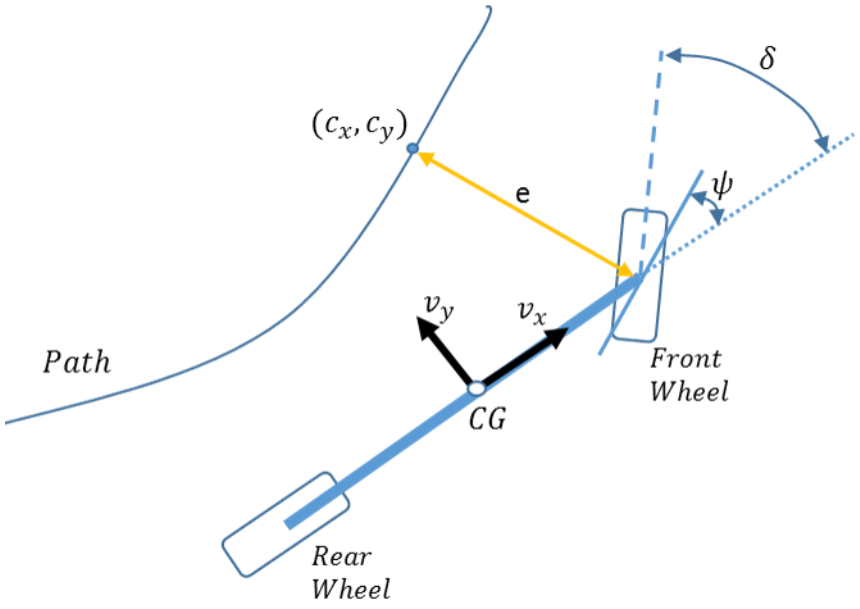

Figure 3.4 : Geometric explanation of Stanley Method.

Stanley method is one of the most powerful path tracking methods and it has several extensions that increases its effectiveness further. Due to the direct use of cross track error, it converges more rapidly to the desired path. In equation 3.8, the contribution of cross track error is scaled by the vehicle velocity. Its stability guaranteed robustness is another strength of the method [24]. However if the path is not smooth, in other words if the curvature of the path is not changing continuously, this method yields large tracking errors. Because Stanley method uses the closest point on the trajectory to the front wheel, there is no "look forward" behavior of the strategy. That means the vehicle does not start to steer until the vehicle arrives in to a sharp turning point and deviates largely from the planned trajectory afterwards.

3.3 Steady State Cornering Method

A different look ahead method that considers steady state cornering dynamics of the vehicle is proposed by K. Jalali et al. [14]. As can be seen in previous sections, Pure-Pursuit and Stanley methods depends on simple geometry; they do not consider the dynamics of the vehicle. On the other hand, in this method, a look ahead point that is not necessarily on the trajectory is selected. Then the target point is selected as the closest point on the planned trajectory to the look ahead point. Geometrical representation of the model is shown in Figure 3.5.

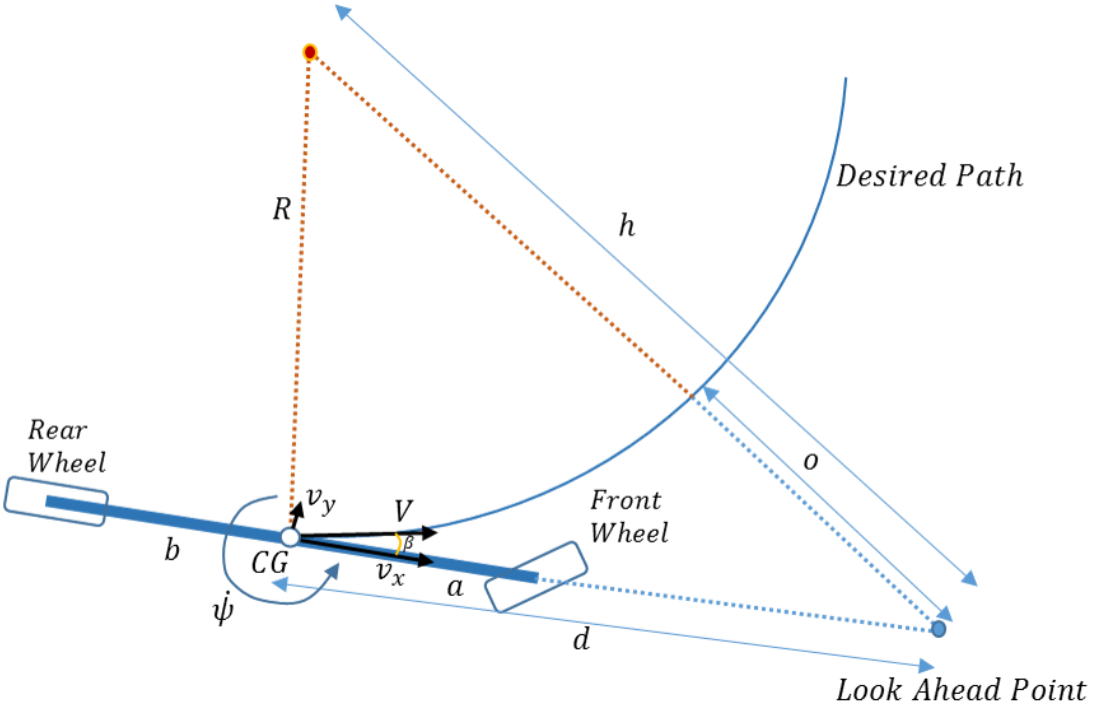

Figure 3.5 : Geometric explanation of Steady State Cornering Method.

At steady state $\dot{v}_y$ and $\ddot{\psi}$ terms become zero. That is why equations 2.35 and 2.36 become [14]:

$$0 = \frac{-C_f v_y - C_f L_f \dot{\psi}}{v_x} + C_f \delta + \frac{-C_r v_y + C_r L_r \dot{\psi}}{v_x} - m v_x \dot{\psi} \tag{3.8}$$

$$0 = \frac{-L_f C_f v_y - L_f^2 C_f \dot{\psi}}{v_x} + L_f C_f \delta + \frac{L_r C_r v_y - L_r^2 C_r \dot{\psi}}{v_x} \tag{3.9}$$

From this two equation, the steady state lateral velocity can be calculated as a function of the steady-state yaw rate as follows [25]:

$$v_y = T\dot{\psi} \tag{3.10}$$

where T is [25]:

$$T = L_r - \frac{L_f m v_x^2}{C_r(L_r + L_f)} \tag{3.11}$$

In general, the following statements can be made for a vehicle in steady-state circular motion [25]:

$$V = \sqrt{v_x^2 + v_y^2} \tag{3.12}$$

$$V = R\dot{\psi} \tag{3.13}$$

One can now obtain new expressions for $\dot{\psi}$ and δ from above Equations that are only in terms of the vehicle longitidunal velocity v_x, the radius of curvature R, and the vehicle parameters [25]:

$$\dot{\psi} = \frac{v_x}{\sqrt{R^2 - T^2}} \tag{3.14}$$

$$\delta = \frac{1}{\sqrt{R^2 - T^2}}\left(L_f + L_r - \frac{m v_x^2 (L_f C_f - L_r C_r)}{(L_f + L_r) C_f C_r}\right) \tag{3.15}$$

According to equation 3.11, the largest value for T is always less than L_r, which is the distance of the rear axle to the vehicle center of gravity. Since, in reality, a vehicle with front steering system can never have a radius of curvature less than its wheelbase, equations 3.14 and 3.15 will never encounter a singularity problem.

In order to calculate an appropriate expression for the steady-state look ahead offset o, where $o = h - R$, and expression for h is first defined as follows [25]:

$$h = \sqrt{d^2 + R^2 - 2Rd\cos(\frac{\pi}{2} + \beta)} = \sqrt{d^2 + R^2 - 2Rd\frac{v_y}{V}} \tag{3.16}$$

Using Equations 3.11, 3.14 and 3.16, the final expressions for h and o are obtained as follows [25]:

$$h = \sqrt{d^2 + R^2 + 2dT} \tag{3.17}$$

$$o = \sqrt{d^2 + R^2 + 2dT} - R \tag{3.18}$$

Finally, from Equations 3.16 and 3.19 the ratio between the desired steering input δ and the look ahead offset o is calculated as follows [25]:

$$\frac{\delta}{o} = \frac{\frac{1}{\sqrt{R^2 - T^2}}\left(L_f + L_r - \frac{mv_x^2(L_f C_f - L_r C_r)}{(L_f + L_r)C_f C_r}\right)}{\sqrt{d^2 + R^2 + 2dT} - R} \tag{3.19}$$

At this point, two important assumptions are made by the authors of paper in order to simplify equation 3.20 [25]. First, using Taylor's expansion [14]:

$$\forall x, \varepsilon \in \mathbb{R}, x > 0 : if \frac{|\varepsilon|}{x} \ll 1 \Rightarrow \sqrt{x + \varepsilon} = \sqrt{x} + \frac{\varepsilon}{2\sqrt{x}} \tag{3.20}$$

And assuming that $\frac{|d(d+2T)|}{R} \ll 1$, equation 3.19 can be rewritten as follows [25]:

$$\frac{\delta}{o} = \frac{2\left(L_f + L_r - \frac{mv_x^2(L_f C_f - L_r C_r)}{(L_f + L_r)C_f C_r}\right)}{d(d + 2T)\sqrt{1 - \frac{T^2}{R^2}}} \tag{3.21}$$

Next, by assuming that $\frac{|T|}{R} \ll 1$ and, thus $\sqrt{1 - \frac{T^2}{R^2}} \approx 1$, equation 3.21 can be further simplified as follows and The steering angle due to Steady State Cornering method is [25]:

$$\delta_{SSC} = \frac{2\left(L_f + L_r - \frac{mv_x^2(L_f C_f - L_r C_r)}{(L_f + L_r)C_f C_r}\right)}{d(d + 2T)} o \tag{3.22}$$

The required steering angle is calculated as a function of the look-ahead offset *o*, the vehicle longitudinal velocity $v_X$and some other vehicle parameters. The calculation is done using a linearized dynamic bicycle model at the steady state cornering conditions, i.e., constant lateral velocity and yaw rate.

As can be seen, the control strategy includes vehicle longitudinal velocity v_x. This makes the method to be adaptive for velocity changes. In addition, the look ahead distance can be configured to adjust the stiffness of the control law. Speed adaptation and model based control can be counted as other advantages of the method. However modelling errors and parameter changes can be problematic for this method.

In addition, the look ahead distance optimization is required with respect to velocity as in the case of Pure-Pursuit. A dynamic look ahead distance is proposed in [14] as follows

$$d_{look-ahead}(t) = d_{const} + t_{driver} v_x(t) \tag{3.23}$$

where d_{const} is a constant distance which is an ordinary driver look ahead at low velocities, t_{driver} is the steering reaction time of an ordinary driver and $u(t)$ is the vehicle's longitudinal velocity. As one can see, similar to a real driving case, look ahead distance increases as speed increases.

3.4 Some Key Algorithms for Implementations of the Methods

In the literature, most of the methods need the closest point calculation. Pure-Pursuit method uses closest point calculation to search for look ahead distance, Stanley method needs it to calculate cross track error and curvature of this point and Steady-State Cornering method also needs it for calculation of steady-state look ahead offset, *o*. However, there is a lack of information about calculation of closest point on path. In this section, implementation of path and some key algorithm are shared..

3.4.1 Implimentation of paths

The path is created by points of arrays composed of x and y coordinates respect to the global coordinate system. Also non-curvature changes are permitted. An example of path coordinates and the path itself can be seen below in Table 3.1 and Figure 3.6 respectively.

Table 3.1 : Example path coordinate points

X points	Y points
2	2
4	3
5	3
12	9
12	12

Table 3.1 : Example path coordinate points (cont'd)

X points	Y points
0	12
-0.5	0
1	-2

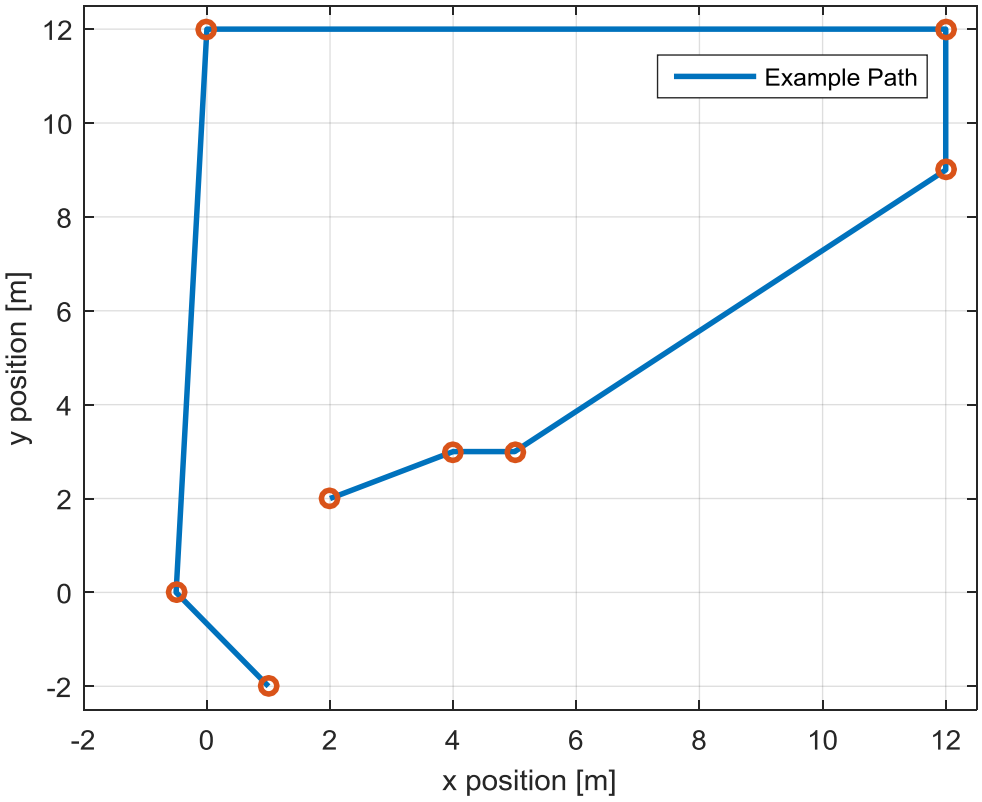

Figure 3.6 : An example path.

3.4.2 Closest point algorithm

As mentioned in the beginning of this part, all methods need to calculate the closest point. In order to find the closest point, path should be investigated partially. In this thesis, a line between two points is called 'segment'. An illustration of segments on an example path is presented in Figure 3.7. Here *i* indicates the index of each segment.

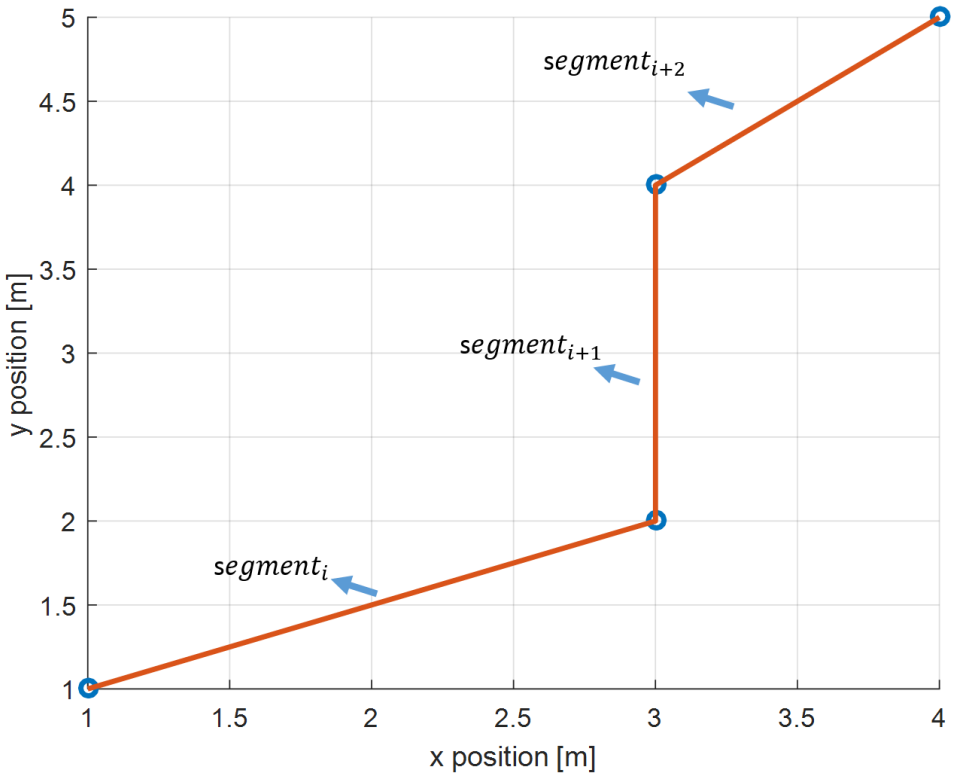

Figure 3.7 : Segments on an example path.

3.4.2.1 Finding closest point on a segment

In order to find the closest point, vector norms can be used. In Mörtberg's thesis, there is some information on how to find the closest point on a segment [21]. Let S_1 be the starting point of the segment; S_2 be the end point of the segment and P be the search point which are needed to calculate the closest point. These points are illustrated in Figure 3.8.

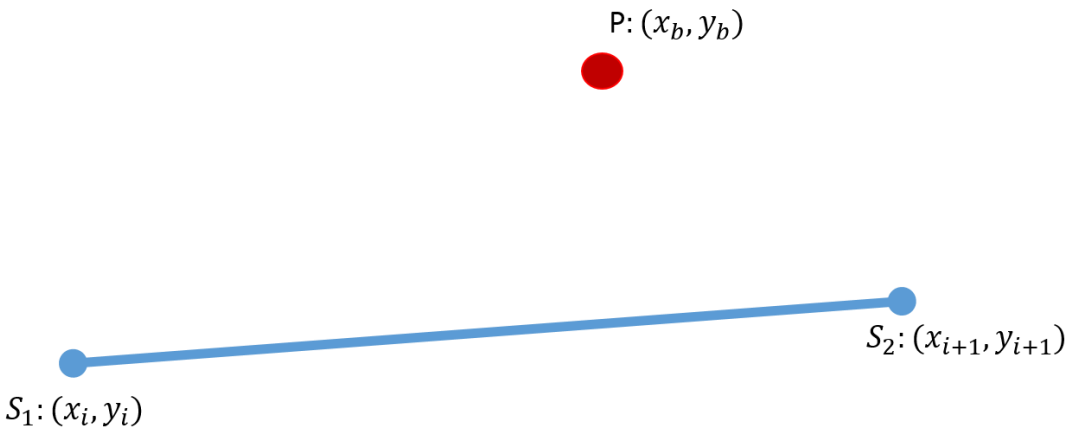

Figure 3.8 : Segment, Segment Points and Search point.

In Figure 3.8, the closest point should be on the segment. In order to find this point, let $\vec{V}$ and $\vec{W}$ be vectors described as follows:

$$\vec{V} = S_2 - S_1 \tag{3.24}$$

$$\vec{W} = P - S_1 \tag{3.25}$$

As it is seen in Figure 3.9, closest point is the orthogonal projection point of $\vec{W}$ on $\vec{V}$.

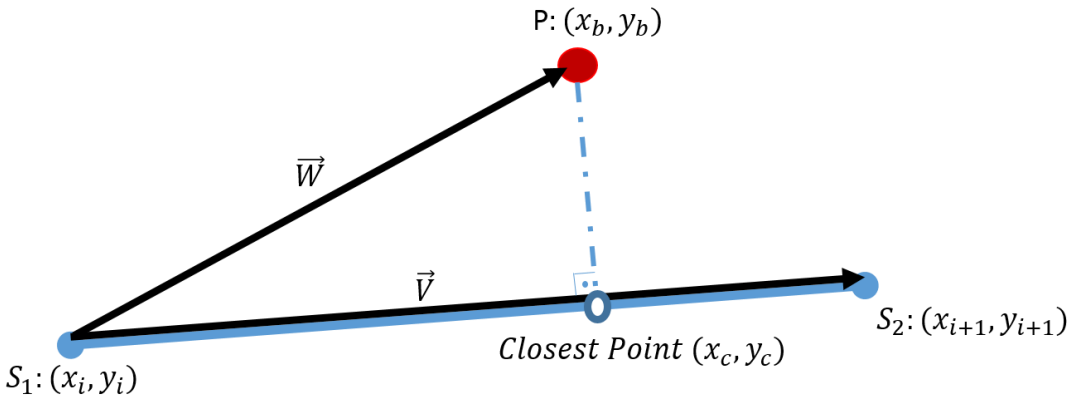

Figure 3.9 : Closest point inside segment.

The closest point $(x_c,\ y_c)$ can be found from equation 3.26:

$$Closest\ Point = S_1 + \left(\frac{\vec{W} \cdot \vec{V}}{\vec{V} \cdot \vec{V}}\right)\vec{V} \tag{3.26}$$

However, there are other cases where the closest point is not on the segment. These cases are shown in Figure 3.10 and 3.11.

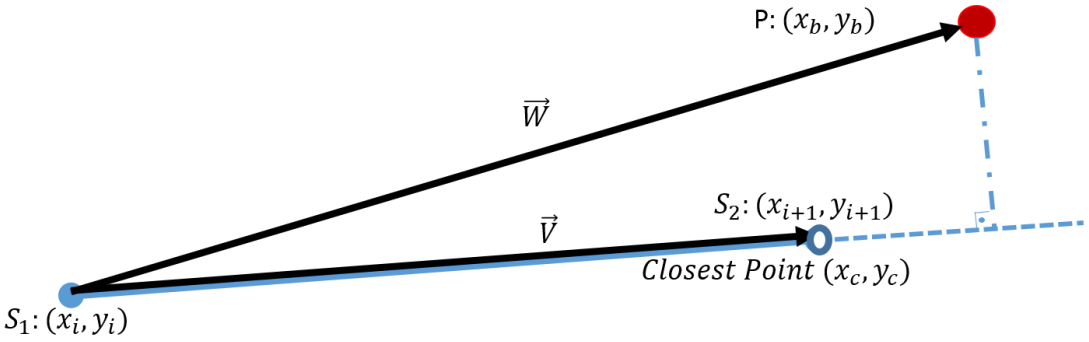

Figure 3.10 : Closest point in front of the segment.

In Figure 3.10, the orthogonal projection is in front of the segment's endpoint. Therefore, the closest point is the end of the segment point for this case. Dot product of the vectors is used for mathematical representation of this case in equation 3.27.

$$if\left(\vec{V}\cdot\vec{V} < \vec{W}\cdot\vec{V}\right)\ Then\ \ Closest\ Point = S_2 \tag{3.27}$$

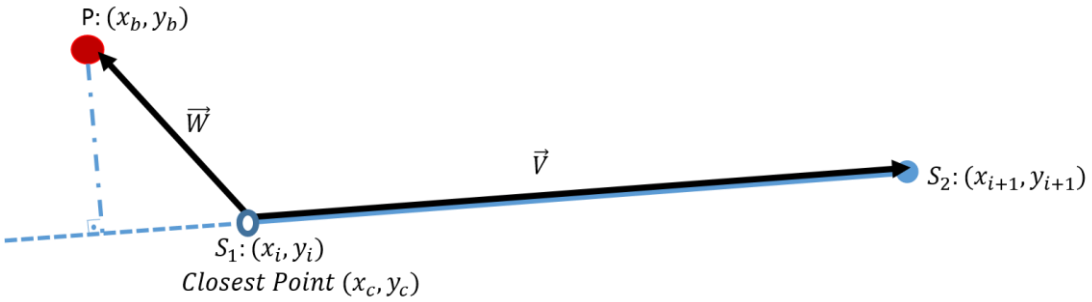

Figure 3.11 : Closest point behind the segment.

In Figure 3.11, the orthogonal projection is behind the segment's endpoint. Therefore, the closest point is the starting point of the segment in this case. dot product of the vectors is used for mathematical representation of this case in equation 3.28.

$$if\left(\vec{W}\cdot\vec{V} < 0\right)\ Then\ \ Closest\ Point = S_1 \tag{3.28}$$

If the angle between two vectors is higher than 90 degrees, vector dot product gives negative value. To sum up, if we conclude our calculations, the closest point function on a segment is given as follows:

$$Closest\ Point = \begin{cases} \vec{W}\cdot\vec{V} < 0 & S_1 \\ \vec{V}\cdot\vec{V} < \vec{W}\cdot\vec{V} & S_2 \\ else & S_1 + \left(\dfrac{\vec{W}\cdot\vec{V}}{\vec{V}\cdot\vec{V}}\right)\vec{V} \end{cases} \tag{3.29}$$

3.4.2.2 Finding closest point on a path

In the previous section, the closest point calculation on a segment is done. Nevertheless, the path is composed of segments. In order to find the closest point regarding to search point, *P*, distances between search point and the closest points of each segments must be compared. For example, in Figure 3.12, d_3 is the minimum distance comparing to d_1 and d_2.

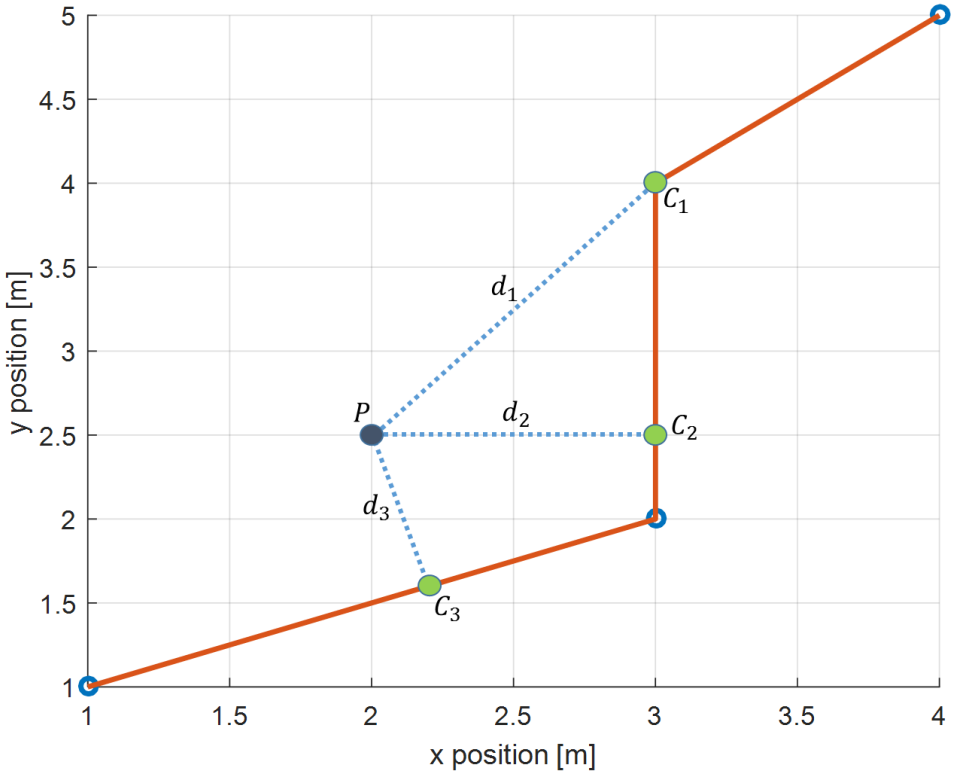

Figure 3.12 : Closest points on all segments.

Now the closest point to the search point on the path can be found, however, if all closest points of segments and their distances are calculated at each step, it will cause two major problems which are work load and skip path problems. Firstly, the path can be formed by thousands of points and the calculation of closest point could take too much time. Secondly, if the path has cross-sections or intersections somewhere in the path, algorithm can skip some segments of the path. In order to prevent this problem, index system is used. At each step, search segments are limited with an iteration number. Yet, this number should be chosen wisely, it should be enough for finding the closest point and not too large to increase the work load. After finding closest point by the help of iteration, index parameter will be updated, then the new closest point is searched from the updated index for the next step. To sum up, flow chart of the closest point search algorithm is demonstrated below. At first, a d_s value is assign to start the algorithm. If there was not an initial assignment, the smallest distance calculation might be stuck and not be updated with other segments. If d_s is too small, it may result in malfunctioning or not functioning of the algorithm. Due to this reason, d_s is given a high value, d_{max} . After that, $i_{updated\ \text{index}}$, which is calculated at previous step time, is used.

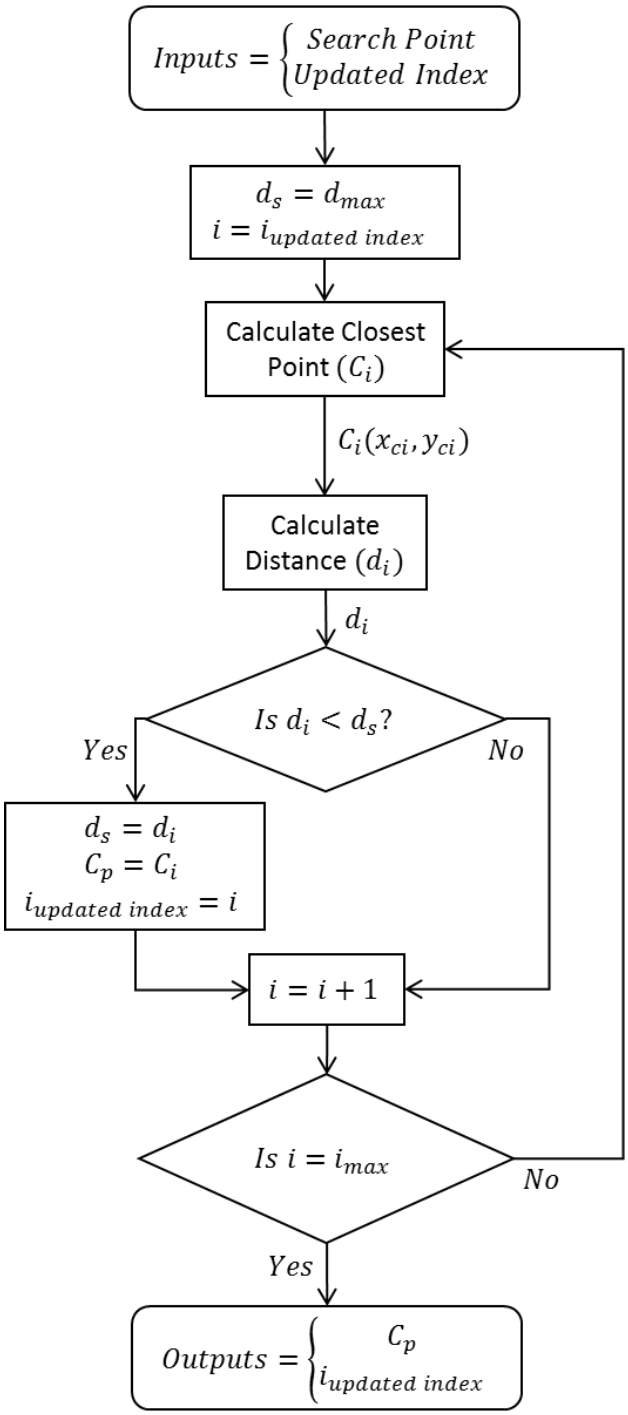

Figure 3.13 : Flow chart of Closest Point Search Algorithm.

Then, distances between search point and the closest points of segments are compared and if that segment has smallest distance than others, C_p and index are updated until the end of iteration number i_{max}.

3.4.3 Pure Pursuit goal point algorithm

Pure Pursuit look ahead point is described as certain distance ahead closest point on path. Therefore, it can be described as a circle and a line intersection problem. Circle center is the search point, *P*, circle radius is look ahead distance, l_d. Then the segment, which intersects with l_d, shall be found. After, intersection point should be calculated.

In order to find intersecting segment, distances from the search point to endpoints of the segments are compared with look ahead distance. If the distance between the endpoint of a segment and the search point is greater than look ahead distance, this segment has intersection with look ahead distance. The demonstration is given in Figure 3.14.

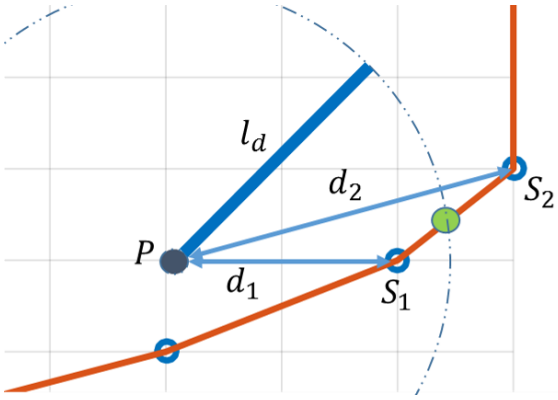

Figure 3.14 : Pure Pursuit Goal Point.

Second step is to find intersection point. First segment is written in line equation form if the segment's starting and end points described as $S_1(S_{1x}, S_{1y})$ and $S_2(S_{2x}, S_{2y})$ and search point $P(P_x, P_y)$ as in equation 3.30.

$$y = \left(\frac{S_{2y} - S_{1y}}{S_{2x} - S_{1x}}\right) x + S_{1y} \tag{3.30}$$

Note that it is not segment equation, it is the line equation of the segment. Therefore, there are three intersection cases between a segment and a circle except no intersection. Note that, due to the algorithm, no intersection case never occurs. These three cases are shown in Figure 3.15.

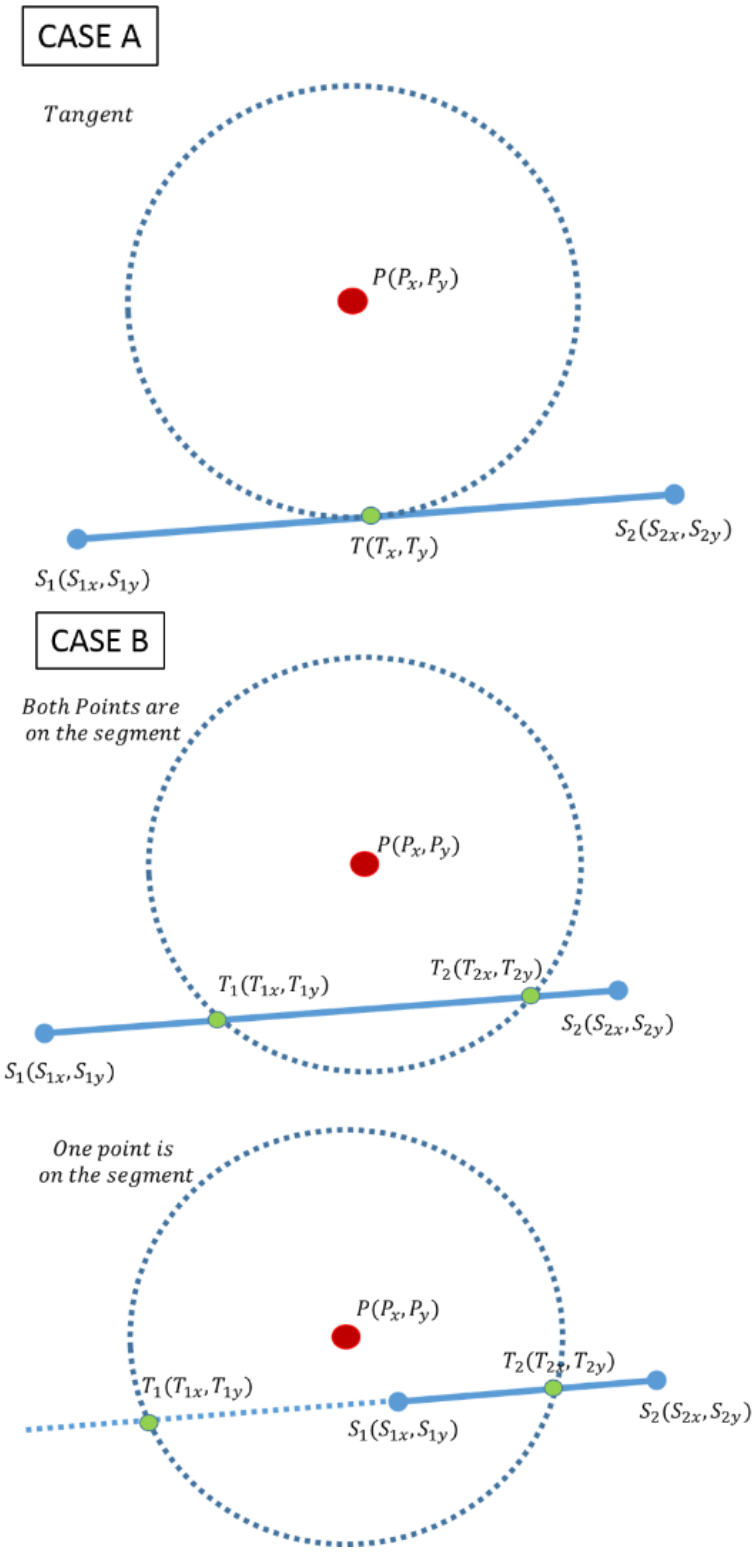

Figure 3.15 : Goal Point cases.

If the circle equation is described as follows:

$$\left(y - P_y\right)^2 + (x - P_x)^2 = l_d^2 \tag{3.31}$$

If equation 3.30 is summarized with equation 3.31 and terms are rearranged, we will get second order function as follows:

$$(m^2+1)x^2+\left[\left(\left(S_{1y}-mS_{1x}\right)-P_y\right)2m-2P_x\right]x + \left[\left(\left(S_{1y}-mS_{1x}\right)-P_y\right)\right]^2+P_x-l_d^2=0 \tag{3.32}$$

where $m=\left(\frac{S_{2y}-S_{1y}}{S_{2x}-S_{1x}}\right)$, if the equation is written in the form of '$Ax^2+Bx+C=0$', constants are described as follows:

$$A=(m^2+1) \tag{3.33}$$

$$B=\left[\left(\left(S_{1y}-mS_{1x}\right)-P_y\right)2m-2P_x\right] \tag{3.34}$$

$$C=\left[\left(\left(S_{1y}-mS_{1x}\right)-P_y\right)\right]^2+P_x-l_d^2 \tag{3.35}$$

As $\Delta=\sqrt{B^2-4AC}$, second order function solution is given as follows:

$$T_{1,2}=\begin{cases}\Delta=0 & \dfrac{-B+\sqrt{\Delta}}{2A}\\ \Delta>0 & \dfrac{-B\mp\sqrt{\Delta}}{2A}\\ \Delta<0 & No\ solution\end{cases} \tag{3.36}$$

In case A shown in Figure 3.15, there is only one intersection point due to the fact that the circle is tangent to the segment. For this case $\Delta=0$ and there is only one condition. However, in case B shown in Figure 3.15, there are two conditions, due to the fact that Δ is greater than zero. Firstly, we need to identify the amount of intersection points on a segment. If the look ahead distance intersects with the segment at two different points, the intersection point close to the endpoint of the segment is chosen. If there is only one intersection on the segment, the other intersection will be outside the segment, then the algorithm automatically picks the point on the segment. To sum up, flow chart of Pure Pursuit Goal Point Find Algoritm is demonstrated below.

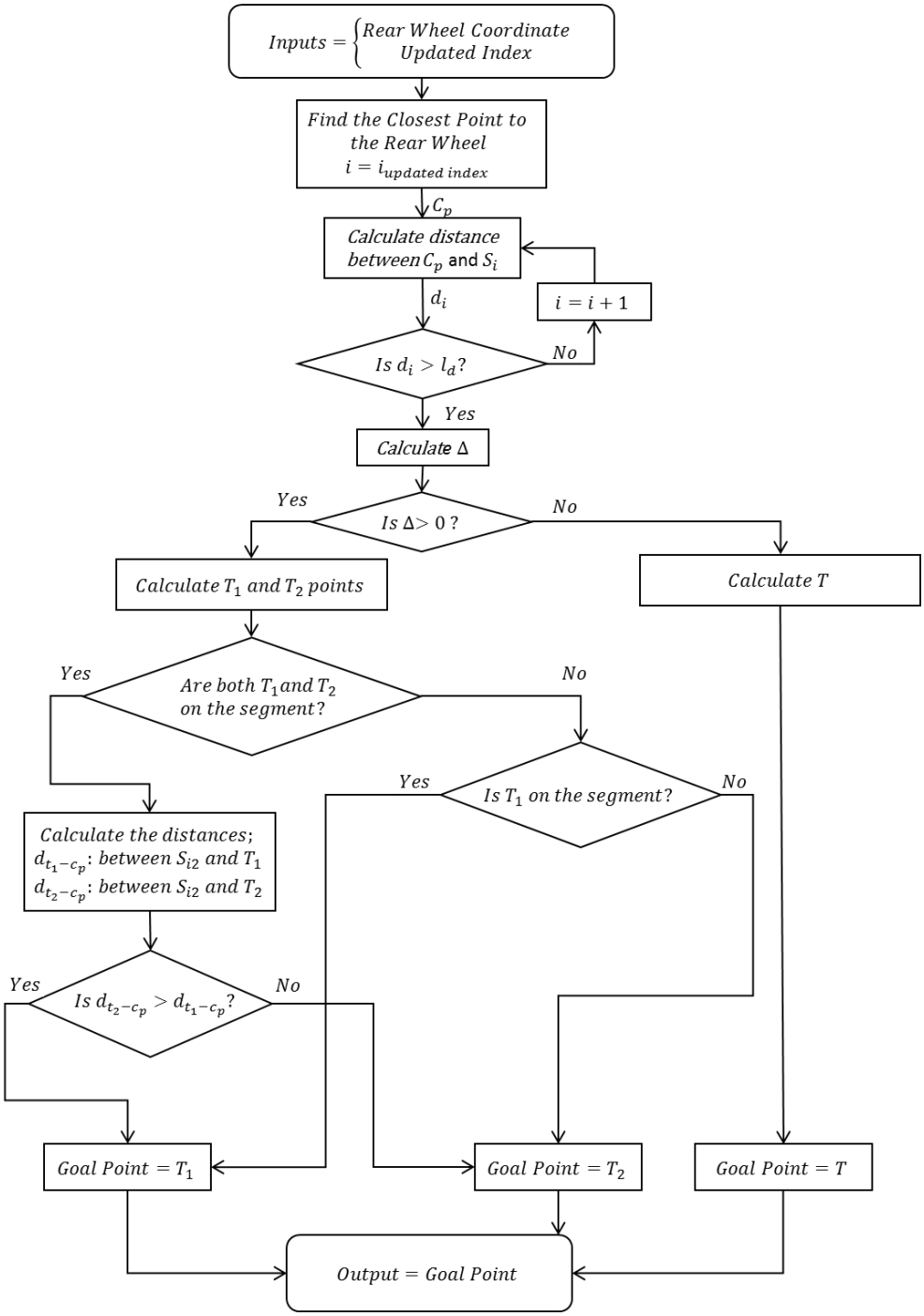

Figure 3.16 : Flow Chart of Pure Pursuit Goal Point Find Algoritm.

3.4.4 Cross Track error algorithm

One of the other unique algorithm was calculation of the cross track error. Stanley and Steady State cornering method needs to find the distance between closest point and their search point. Moreover, this distance should have sign value regarding to being

on the left- or right-hand side of the path. In this thesis, this problem is solved with changing coordinate systems. If closest point and search point coordinates transform regarding to vehicle, the difference between x coordinates will have sign value. For example in order to find Stanley Method cross track error coordinate transformation from global to vehicle is given as follows:

$$x_f' = x_f \cos\left(\psi - \frac{\pi}{2}\right) + y_f \sin\left(\psi - \frac{\pi}{2}\right) \tag{3.37}$$

$$x_c' = x_c \cos\left(\psi - \frac{\pi}{2}\right) + y_c \sin\left(\psi - \frac{\pi}{2}\right) \tag{3.38}$$

The difference between $x_f{}'$ and $x_c{}'$ gives the sign of the error. As it is seen from Figure 3.17 and 3.18, after the transformation of coordinates, y coordinates have different values. That is why the difference between new x coordinates are only used for sign product as seen below.

$$e = sign(x_f' - x_c')\sqrt{(x_f - x_c)^2 + (y_f - y_c)^2} \tag{3.39}$$

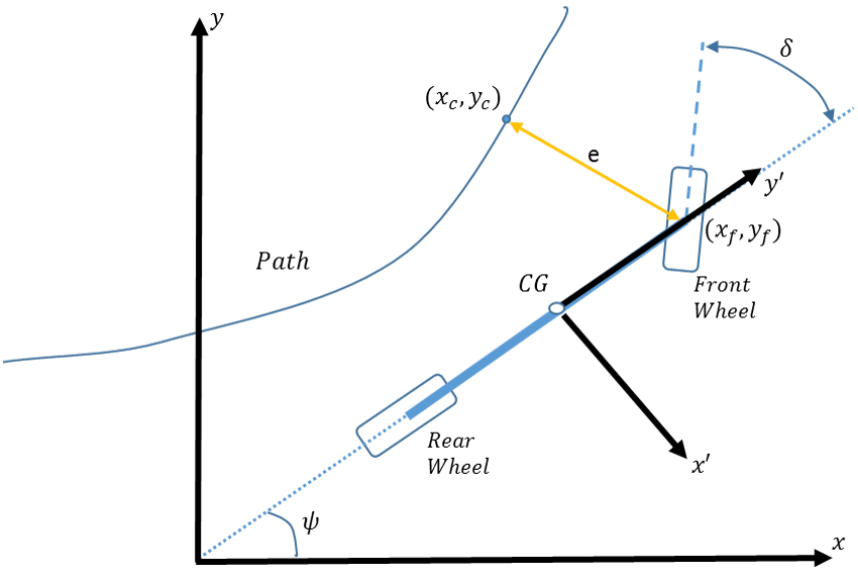

Figure 3.17 : Points and crosstrack error in global coordinate system.

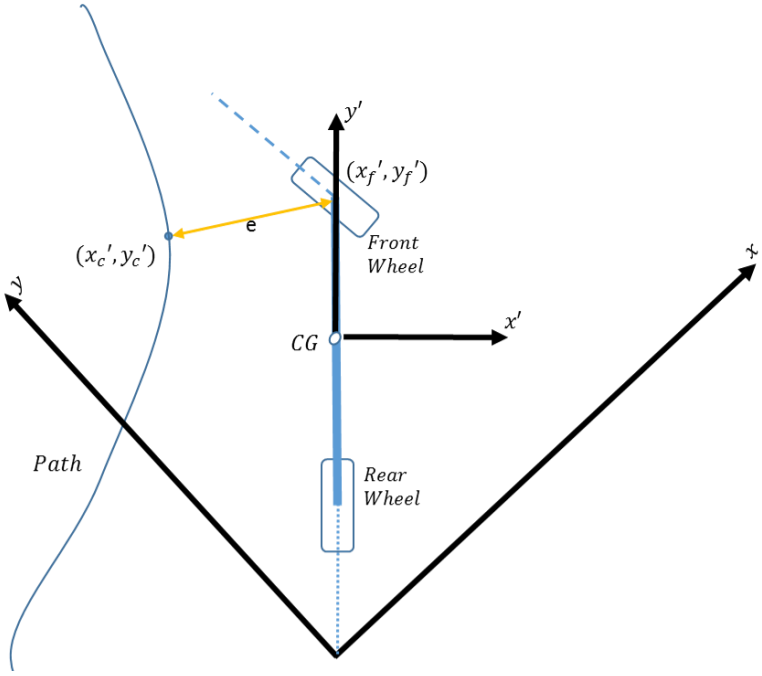

Figure 3.18 : Points and crosstrack error in vehicle coordinate system.

4. HYBRID CONTROLLER

In section 3, three different methods for path tracing are defined, and it is clear that each method has its own advantages and disadvantages. It is known that Stanley method has better performance on smooth trajectories with continuous curvature. However, since the method does not look forward, sharp changes on path points cause large deviations from the target. In contrast, Pure-Pursuit and Steady State Cornering methods look forward in order to maneuver. For that reason, the latter two methods can preview sudden changes on the path beforehand. But on a smooth path, these two methods cut corners and their performances are not as good as Stanley. For this curvature problem of the path, two solutions can be proposed. The first one is to recalculate the path points in order to make the path smoother. In this method, sharp parts of the path are converted to higher order polynomials and the path can be divided into pieces at sharp parts to generate more points. The second solution is to propose a control structure that deals with the sharp changes in the trajectory without changing the original path structure. In this paper, the latter proposition is explained in detail. A hybrid controller that puts together the strong capabilities of different methods that are previously discussed will be presented.

As mentioned before the look ahead methods are good at sharp changes in the trajectory and are bad in cutting corner behavior, and Stanley is vice versa. The proposed hybrid controller is using Pure-Pursuit and Stanley Method at the same time. A weight factor is adjusted depending on the smoothness of the path ahead. As the path gets smoother, the weight of the Stanley method is increased, if a sharp change is ahead the weight of the pure pursuit method is increased. To decide if the path is smooth or not, the look ahead strategy used in steady state method is implemented.

A schematic representation of the problem is shown in Figure 4.1. As in the steady state cornering method, the target point on the path that is the closest to the look ahead point is found.

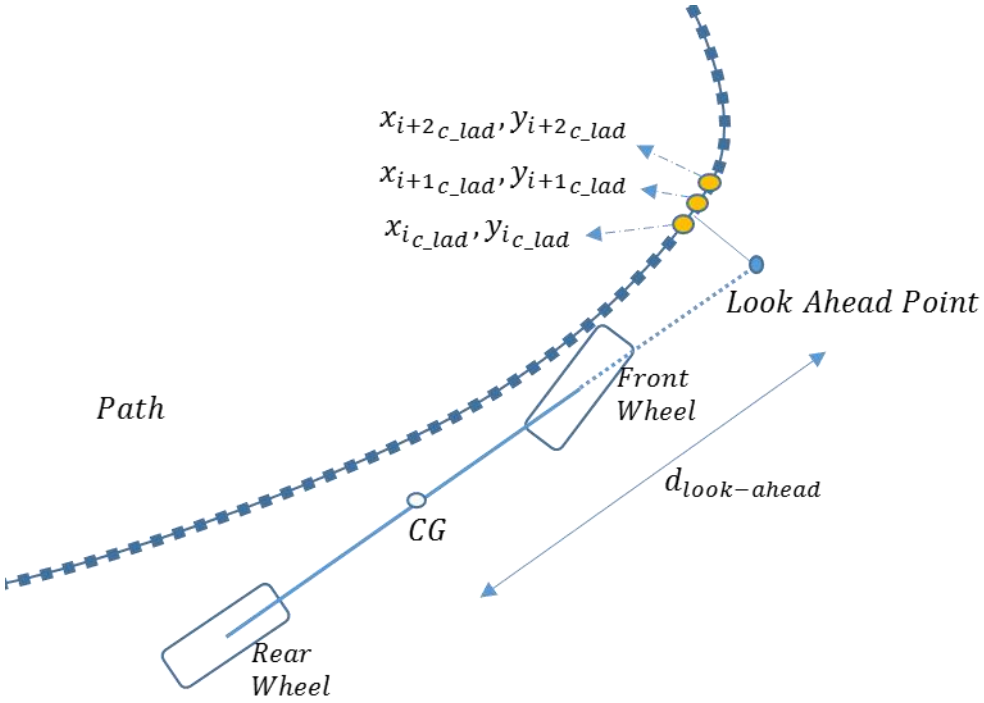

Figure 4.1 : Smoothness Calculation.

Then the angle difference (θ_{dif}) of the path segment that the target point lies on and the succeeding segment is calculated as:

$$\theta_{dif} = \tan^{-1}\left(\frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}}\right) - \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \tag{4.1}$$

In some applications 15 degrees and above is defined as sharp turn, that is why 15-degree angle is selected threshold [26]. Then a weight rule is defined as:

$$\delta_{Hybrid}(t) = \begin{cases} |\theta_{dif}| \leq 15^o & 0.1\delta_{PP}(t) + 0.9\delta_{Sta}(t) \\ |\theta_{dif}| > 15^o & 0.9\delta_{PP}(t) + 0.1\delta_{Sta}(t) \end{cases} \tag{4.2}$$

As given above, if the angle between two segments are larger than 15 degrees a desired steering is weighted on Pure-Pursuit method steering command δ_{PP}, otherwise Stanley method steering command δ_{Sta} is enforced. Two methods are not fully disabled in any case in order not to lose the strong side of each method. The design of hybrid controller structure can be seen in Figure 4.2

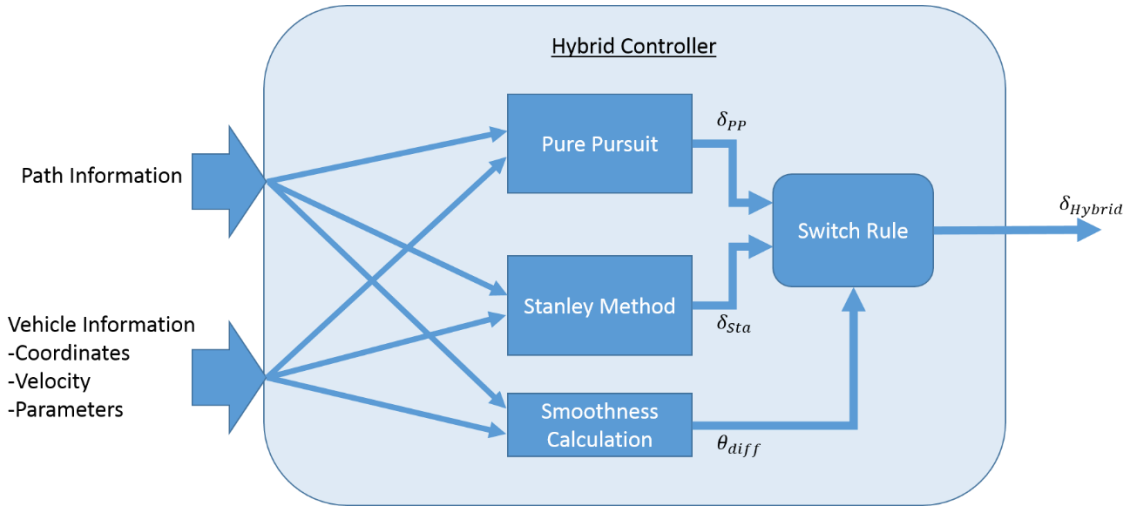

Figure 4.2 : Hybrid controller structure.

However, if there is a very sharp change, such as a perpendicular turn, θ_{dif} term will get larger than 15 degrees just at one time. Hence, the switching will not have any effect on the vehicle's behavior. So as to overcome this, a timer is used to keep the rule valid for a while when a rule change occurs. In order to find this time, 90 degree turn and 50 km/h speed is tested with 0.2 sec increments. As it is seen from Figure 4.3, 1 second timer value gave better results in comparison, that is why timer value is selected as 1 second and hybrid controller finalized as follows:

$$t_h = \begin{cases} |\theta_{dif}| > 15^o & timer: ON \\ t_h > 1 \text{ sec} & timer: OFF \end{cases} \tag{4.3}$$

$$\delta_{Hybrid}(t) = \begin{cases} 0 < t_h < 1 & 0.9\delta_{PP}(t) + 0.1\delta_{Sta}(t) \\ else & 0.1\delta_{PP}(t) + 0.9\delta_{Sta}(t) \end{cases} \tag{4.4}$$

where t_h is the timer value. In Figure 4.4, hybrid controller switching between desired steering angles is shown with blue area. Pure Pursuit and Stanley method desired angles are internal controllers of hybrid controller. They work simultaneously inside the hybrid controller. Pure Pursuit weighted controller is activated at 0.45 sec and then it switched back to Stanley weighted controller at 1.45 sec. One of the most interesting point is that Pure Pursuit method starts to maneuver the vehicle before reaching the sharp turn. When turning starts, Stanley method still wants to follow the front wheel closest path. Due to its nature, it attempts to keep steering in line. Then at 1.18 second, closest point regarding to front wheel reaches to 90 degree turn, Stanley method desires an immediate change in the angle from negative to positive as well as relatively higher degree increase in the steering angle comparing to Pure Pursuit method can be

recognized. In the next section, hybrid controller method performance, comparison between hybrid controller and other methods will be discussed.

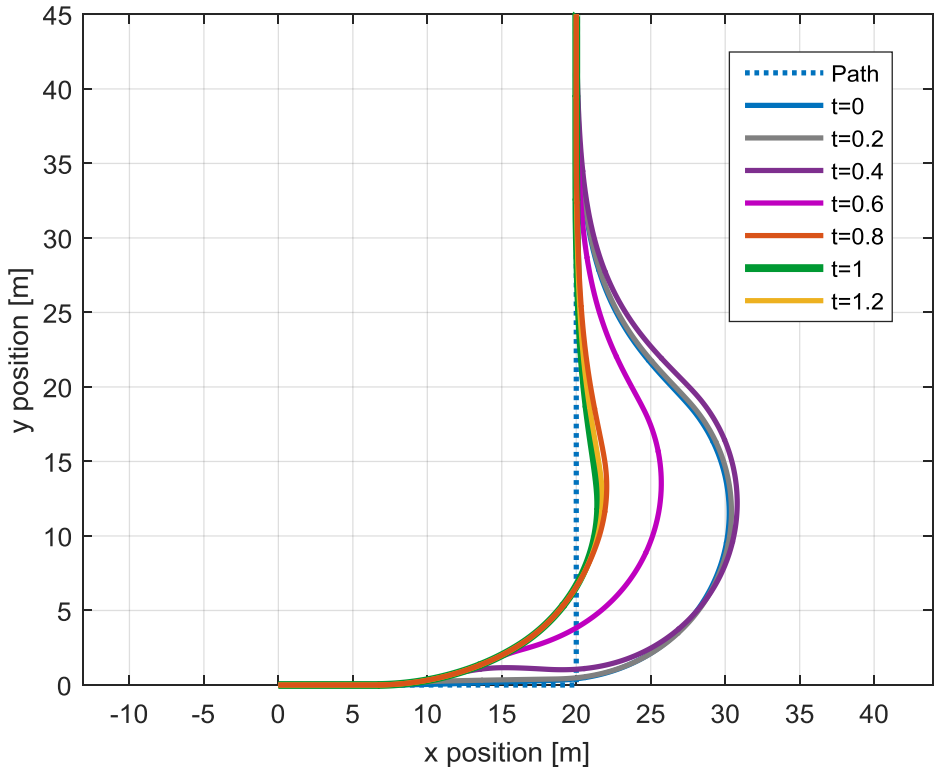

Figure 4.3 : Hybrid Controller performance changes with different timer values.

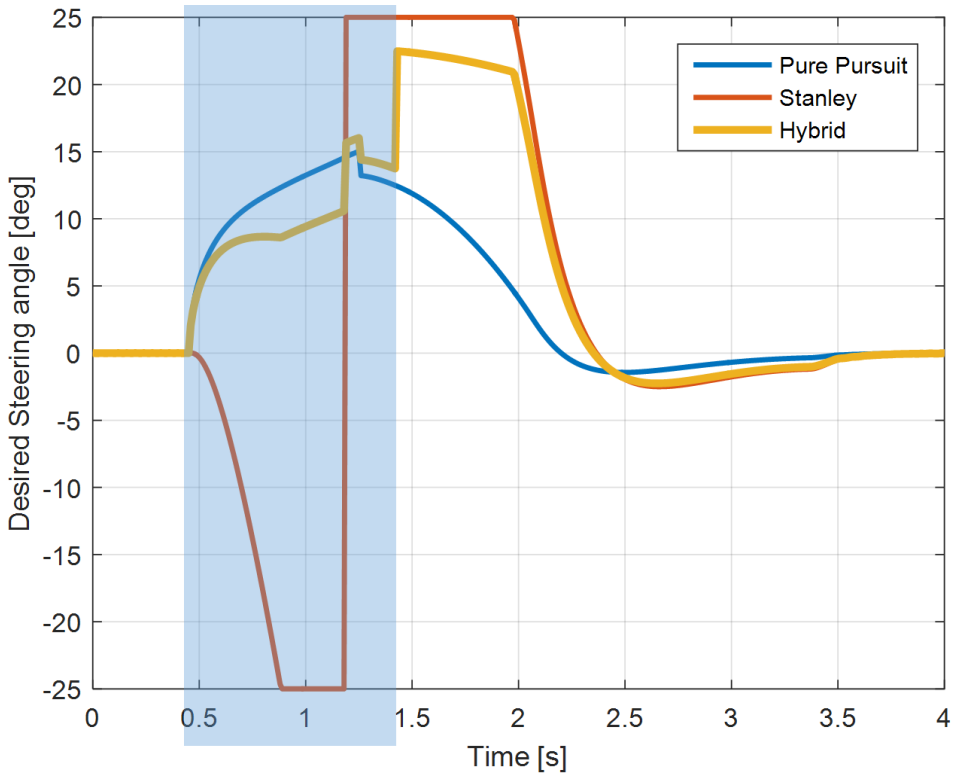

Figure 4.4 : Hybrid controller internal desired angles and final calculated desired angle.

5. SIMULATION RESULTS

In this section, simulation paths, error criterion and simulation results are given. Several paths are defined in order to test and compare the described path tracking algorithms. The methods are tested on three different tracks.

5.1 Paths

In order to make a comparison, three different track generated as Circle, Rectangle and mixed path. In all paths, black triangle is shows starting point of the path (0,0) and indicates direction of the road.

5.1.1 R50 Circle track

As it mention from Section 3, Stanley method is the most powerful method on continuous curves, howerver other methods are not performing as Stanley method. In order to test that, simultaneous curvetaure path generated which named as R50 circle track. This track is a circle with 50m radius and with path points at 0.01 radian increment.

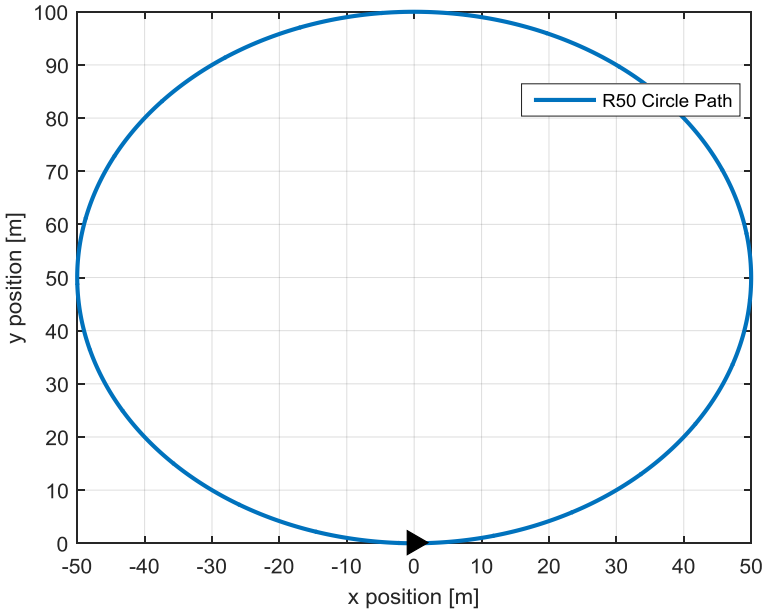

Figure 5.1 : R50 Cricle Track.

5.1.2 150x120 Rectangular track

The second is a rectangular track with 150m width and 120m length. The path points are generated with 1m increments along the path. This track is made for testing sharp turn behavior among the methods. We know that Stanley has problem with sudden changes on curvetures, so 90 degree changes is perfect to test methods performance.

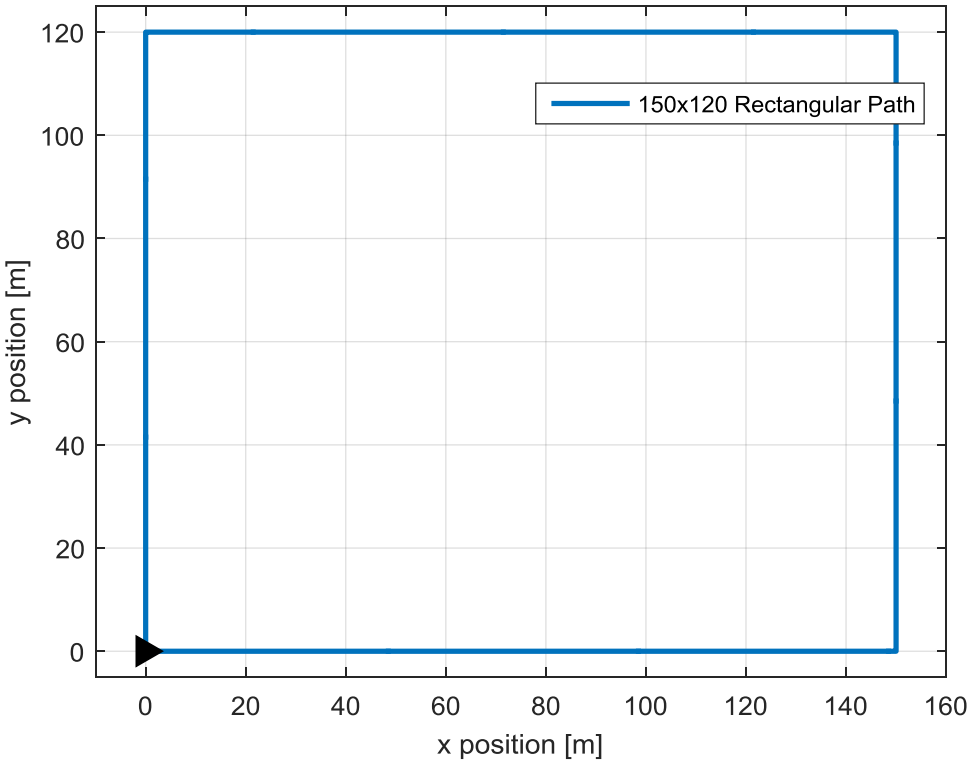

Figure 5.2 : 150x120 Rectangular Track.

5.1.3 Mixed track

For the third track, in order to test smooth and sharp edged path behaviors together, a complex path (called as mixed path) is generated as in Figure 5.3. The mixed path has 50m radius smooth turns, 90 degree turns, 20 degree bumped turns, 45 degree turns and some other smooth and sharp turns. Notice that in mixed path, the distances between path points are not constant. For straight lines 1m increments is used but for circular arcs there is different values are used for increment

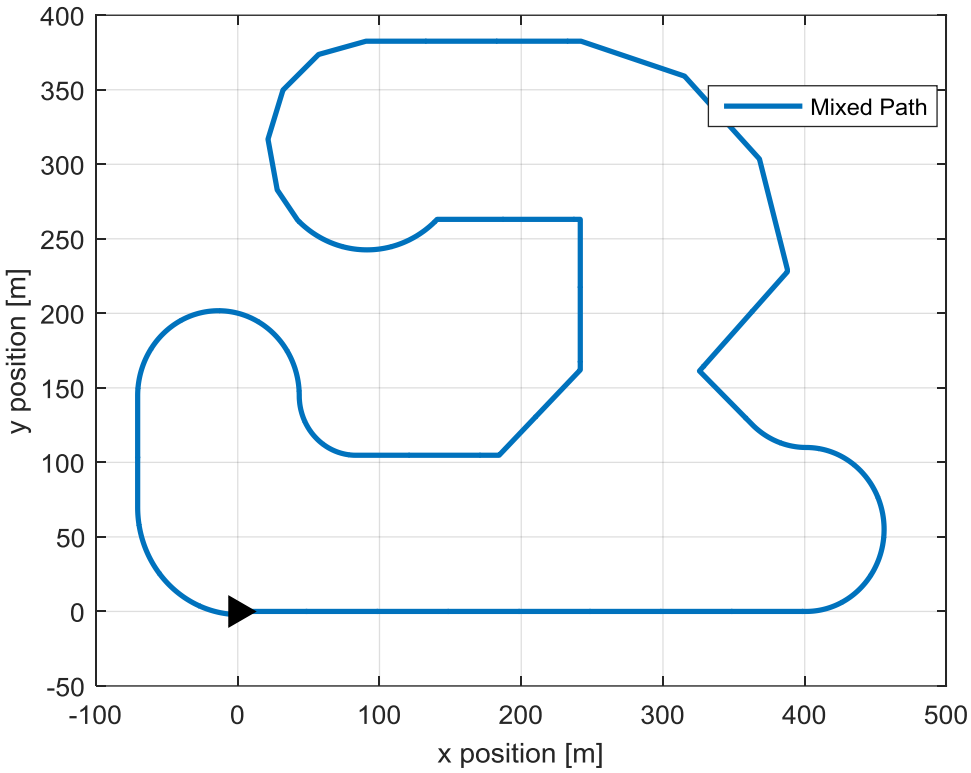

Figure 5.3 : Mixed Path.

5.2 Error Criterion

Cross track error which is shown in Fig. 7 is selected as the error criterion. In figure e_{CG} is the distance between the vehicle CG and the nearest point to the vehicle CG. And also e has the signed value depend on vehicle is on the right and loeft side of the path.

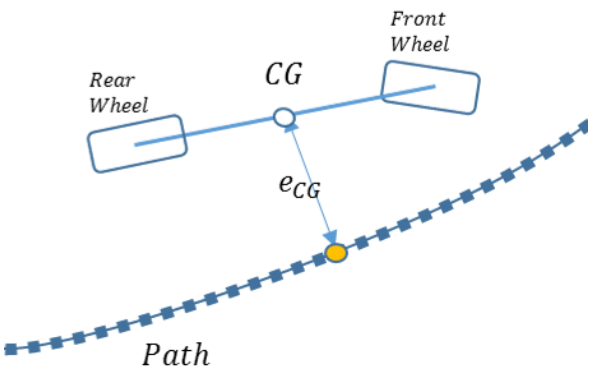

Figure 5.4 : Cross track error of vehicle CG.

The resultant error of each track and method is calculated in two different ways. The first one is the mean absolute error, which depends on L_1 norm:

$$E_1 = \frac{\sum |e_{CG_i}|}{k} \tag{4.1}$$

Second way is L_2 norm or so called as the Euclidean norm which is the square root of the sum of square errors:

$$E_2 = \sqrt{\sum e_{CG_i}^2} \tag{4.1}$$

where *k* represents the time steps.

5.3 Simulation Results

In the simulations the look ahead distance, d_{look_ahead} is selected as described in Steady State Corner Method section for all methods to make a fair comparison. Constant distance and reaction times are selected as 4 meters and 0.7 seconds respectively as proposed in the previous work [14]. For Stanley and Stanley part of the proposed hybrid method controller, *k* is selected as 2.5. For all methods, the desired steering angle is limited with 15 degrees. Three constant vehicle test speeds are selected: slow, normal and fast which are 20 km/h, 50 km/h and 80km/h respectively. All methods run as separately until reaching determined final point in order to make fair comparison. Due to some methods overshooting, their traveling is taking more time than others, that is why reaching final point selected as stop simulation criteria.

5.3.1 R50 Circle track results

Circle track is very smooth, in other words angle changes between path segments are very small. As expected, Errors for the Stanley method are small in comparison to the Pure-Pursuit and the Steady-State methods as shown in Table 5.1. Since the path is very smooth, there is no switching in the hybrid method, so it operates in Stanley weighted mode. One can see in Figures 5.6, 5.8 and 5.10 that Stanley and Hybrid methods are very close to the path. It can be said that small contribution of look forward behavior at low speeds increases the performance of Stanley method and this is because of the added cutting corner behavior. For high speeds, Stanley method gives a better result since, in a sense, it directly senses the cross-track error and tries to compensate it.

Table 5.1 : Error values for R50 Circle Track.

R50 Circle Track		Pure Pursuit	Stanley	Steady State	Hybrid
20 km/h	E_1	0.0465	0.0429	0.2904	**0.0428**
	E_2	3.5343	3.2266	22.1709	**3.2210**
50 km/h	E_1	0.3050	0.0153	0.6948	**0.0051**
	E_2	14.8979	0.7295	33.8730	**0.2577**
80 km/h	E_1	1.2680	**0.2399**	1.6723	0.2697
	E_2	49.4525	**9.1358**	65.3424	10.2756

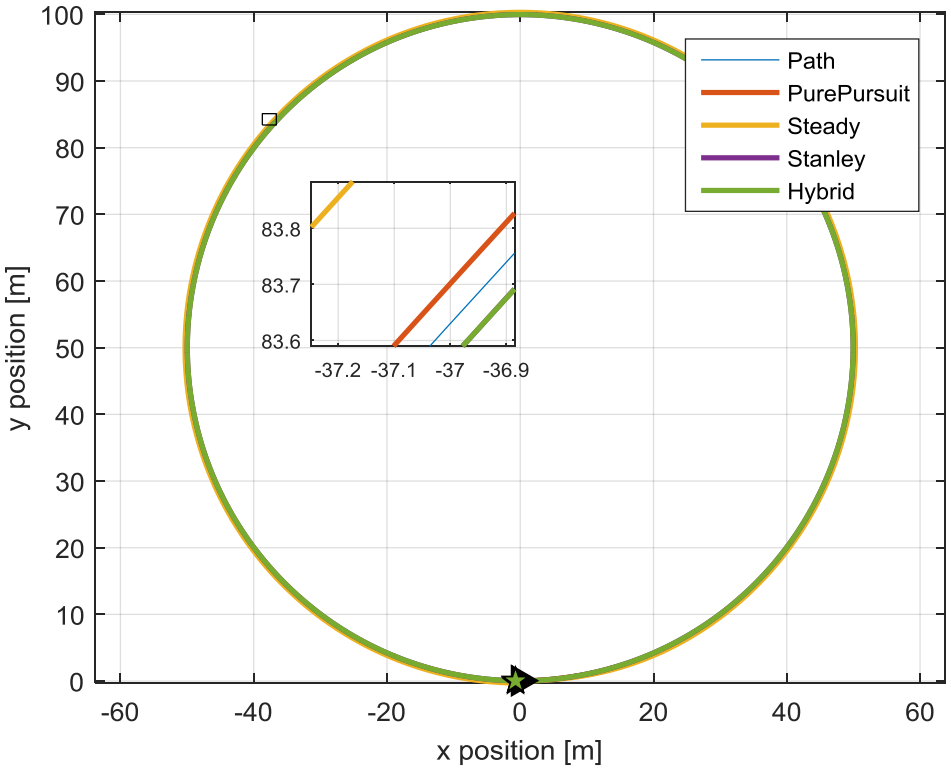

Figure 5.5 : Vehicle CG Position Changes with 20 km/h speed on R50 Circle Track.

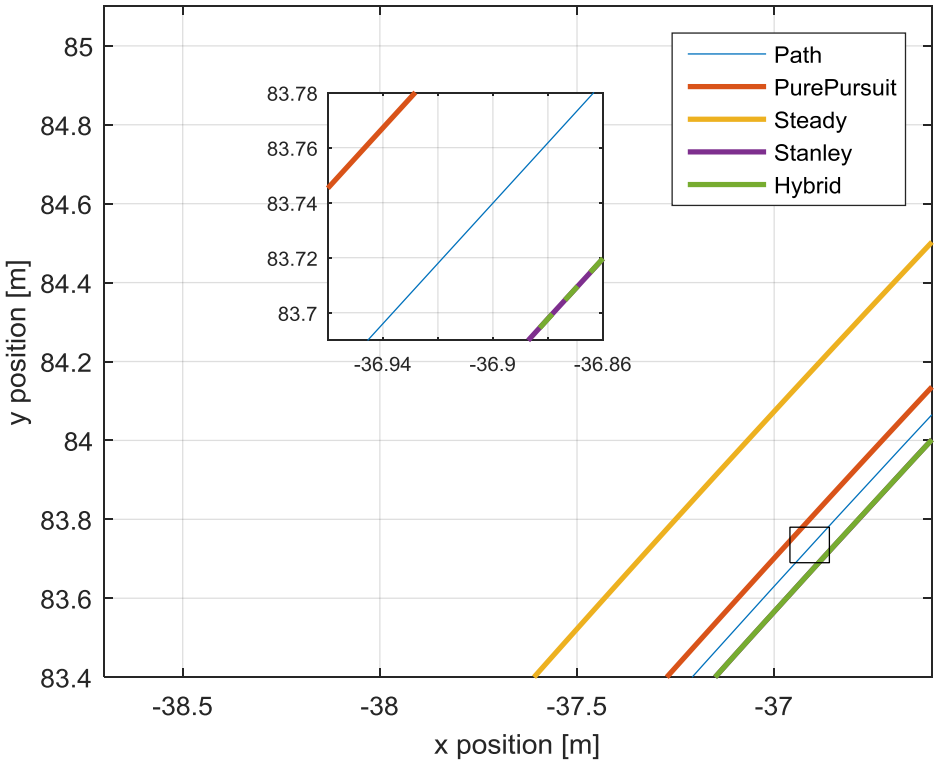

Figure 5.6 : Zoomed view of given sample of previous graph for Vehicle CG Position Changes with 20 km/h speed on R50 Circle Path.

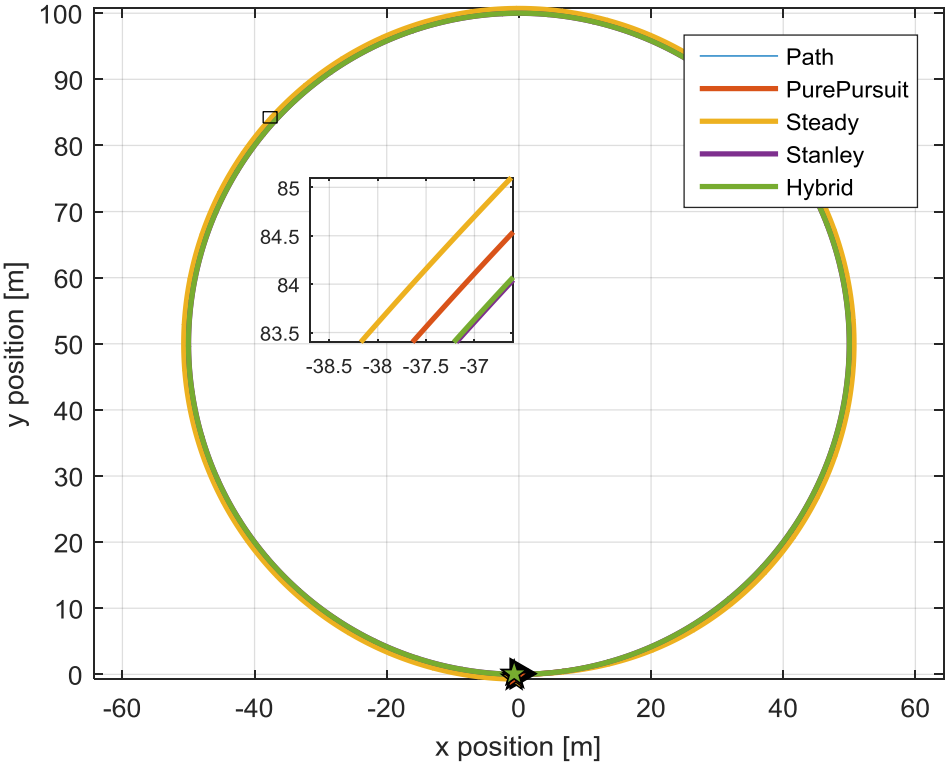

Figure 5.7 : Vehicle CG Position Changes with 50 km/h speed on R50 Circle Track.

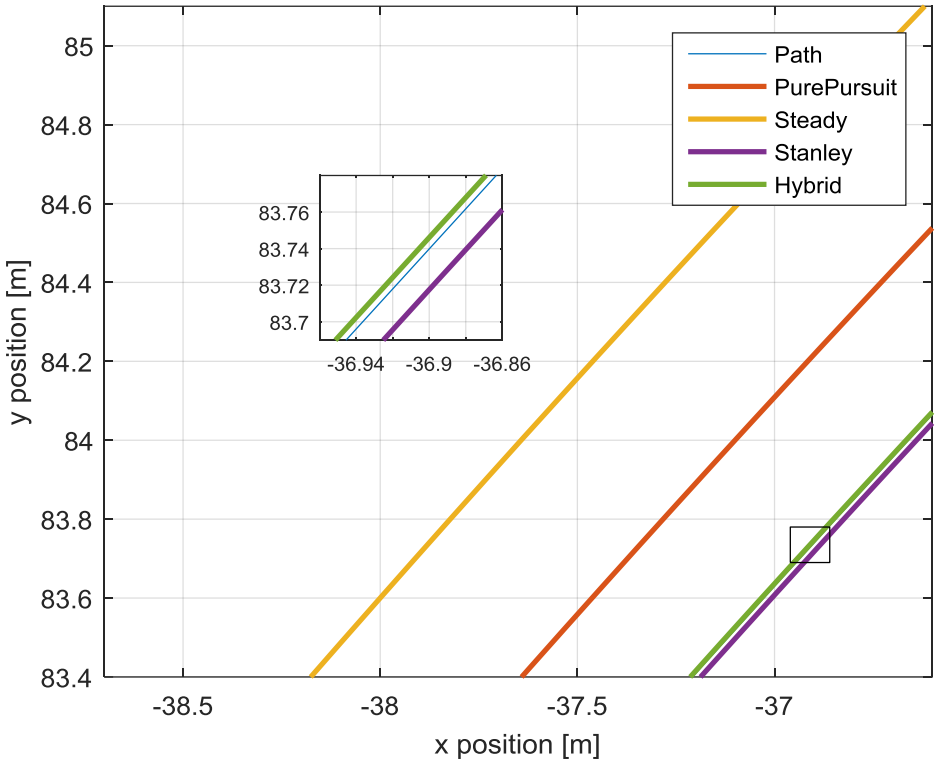

Figure 5.8 : Zoomed view of given sample of previous graph for Vehicle CG Position Changes with 50 km/h speed on R50 Circle Path.

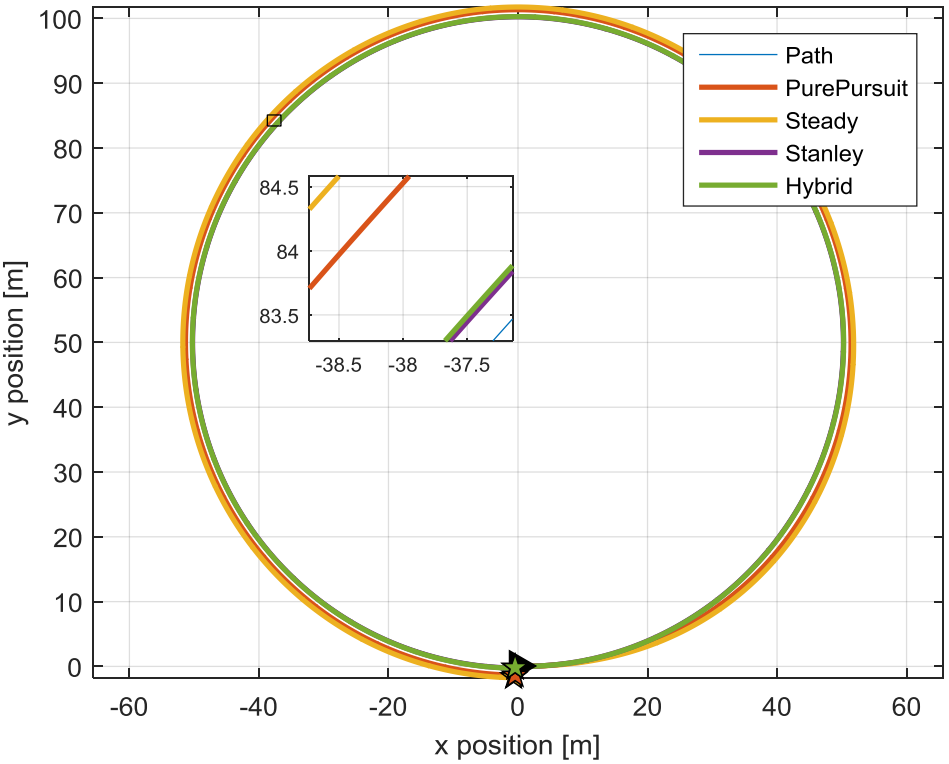

Figure 5.9 : Vehicle CG Position Changes with 80 km/h speed on R50 Circle Track.

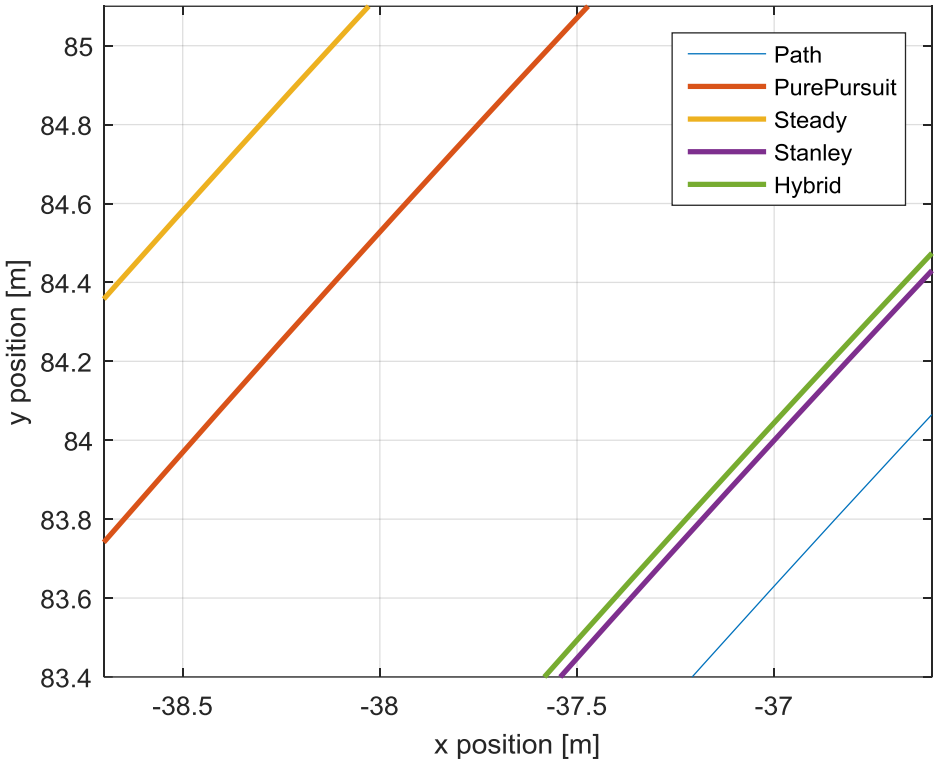

Figure 5.10 : Zoomed view of given sample of previous graph for Vehicle CG Position Changes with 80 km/h speed on R50 Circle Path .

5.3.2 150x120 Rectangular track results

In the second track, sharp turns are tested. As it is seen from Figure5.12, 5.14 and 5.16, Stanley is not starting to steer until vehicle comes very close to the turning point. In other methods, because of the look ahead behavior steering is initiated earlier. Zoomed figure shows that proposed hybrid controller gives the best result in comparison to other methods. It starts turning before the sharp turn, after that due to Stanley method taking action, cross track error is compensated rapidly.

As it is seen from Table 5.2, Hybrid controller performs very well in comparison to other methods. At high speeds because the pure pursuit method makes less overshoot from the path, maximum deviation is the smallest among others. That makes it the best with respect to E_2 error criterion.

Table 5.2 : Error values for 150x120 Rectangular Track.

150x120 Rectangular Track		Pure Pursuit	Stanley	Steady State	Hybrid
20 km/h	E_1	0.1735	0.2348	0.2886	**0.0533**
	E_2	41.1427	80.0260	72.9877	**24.5357**
50 km/h	E_1	0.5241	1.4531	0.7912	**0.2036**
	E_2	59.7504	217.2981	98.6778	**33.3956**
80 km/h	E_1	2.2113	8.3510	5.6096	**2.0596**
	E_2	**171.4176**	703.7385	482.8719	183.7230

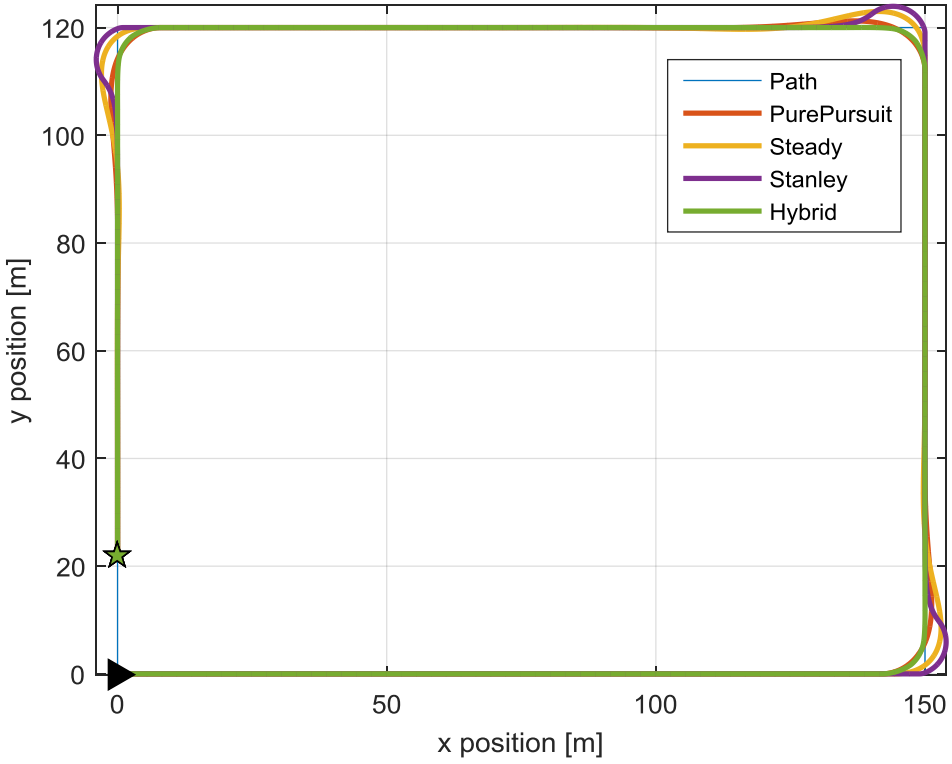

Figure 5.11 : Vehicle CG Position Changes with 20 km/h speed on 150x120 Rectangular Path.

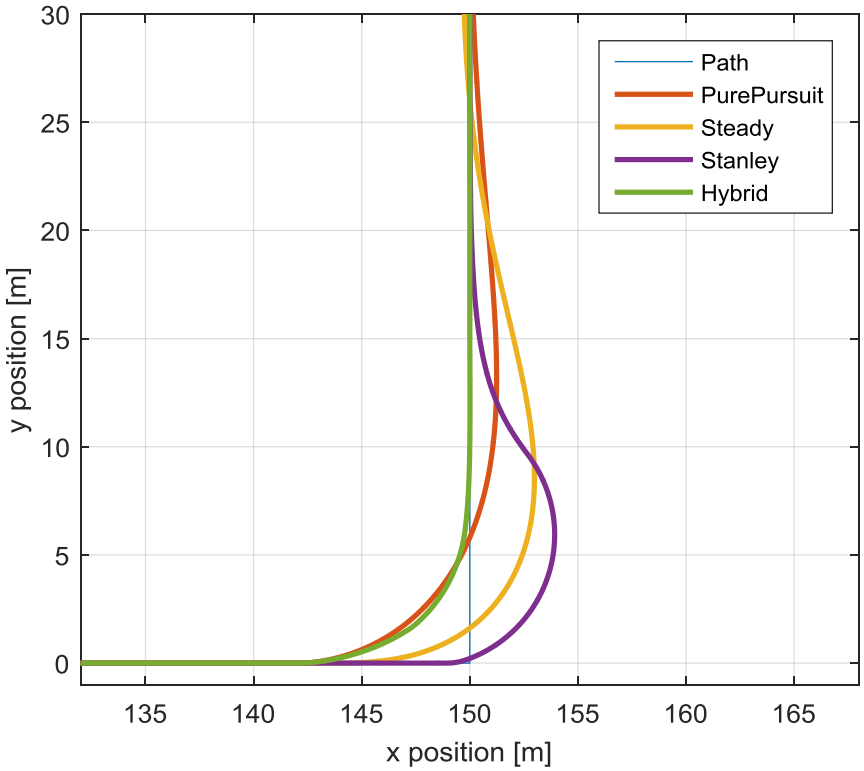

Figure 5.12 : Zoomed right bottom corner of Vehicle CG Position Changes with 20 km/h speed on 150x120 Rectangular Path.

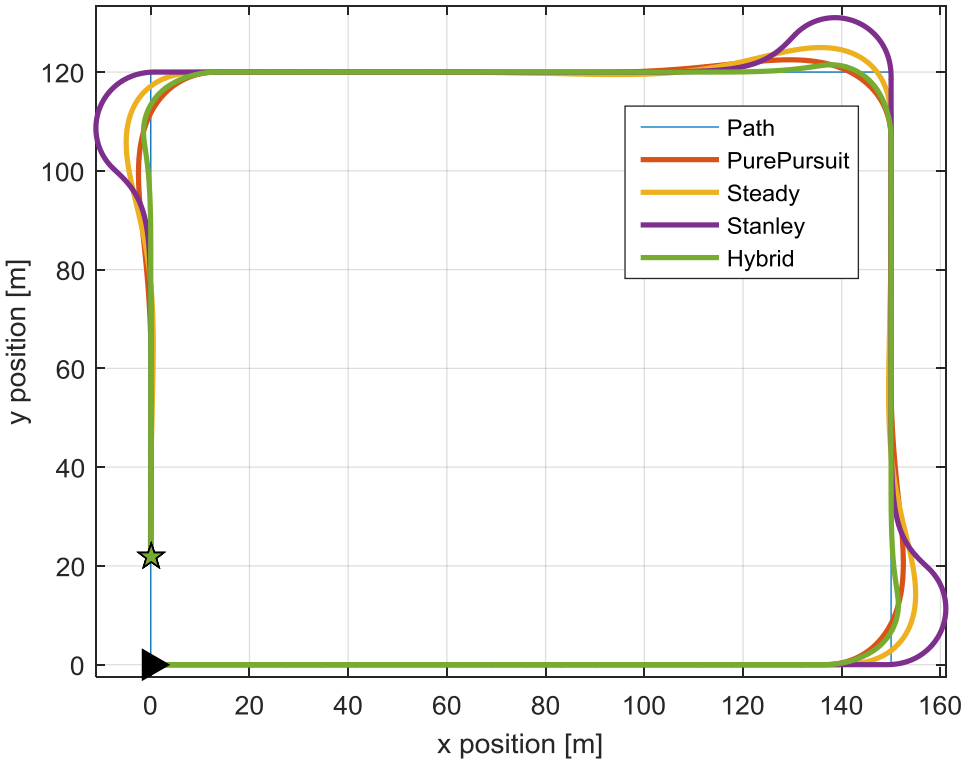

Figure 5.13 : Vehicle CG Position Changes with 50 km/h speed on 150x120 Rectangular Path.

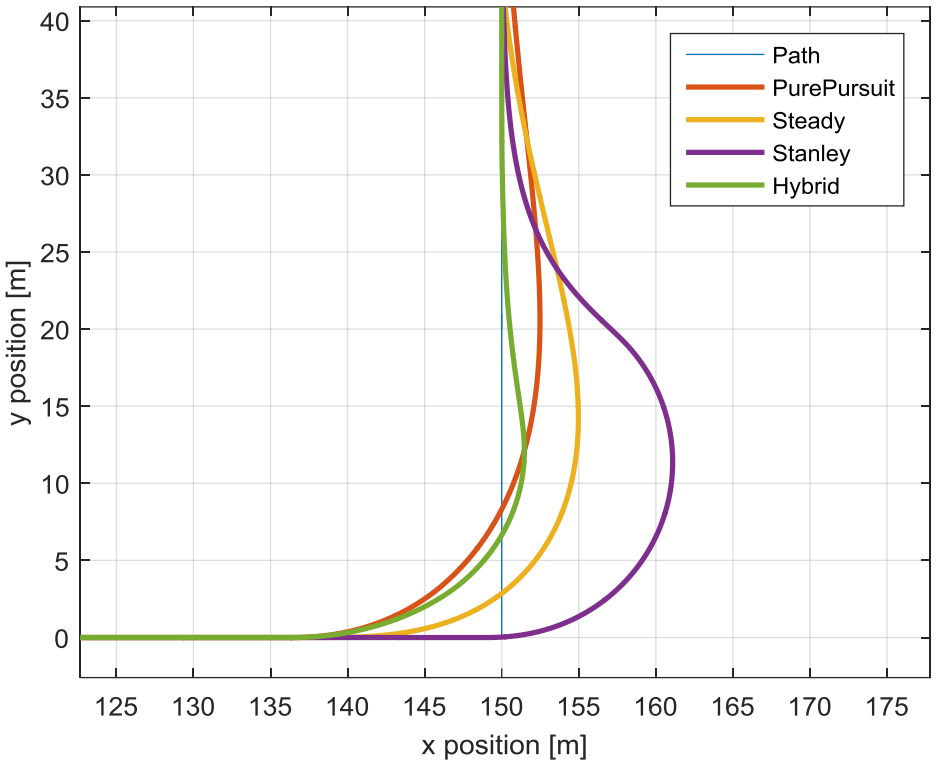

Figure 5.14 : Zoomed right bottom corner of Vehicle CG Position Changes with 50 km/h speed on 150x120 Rectangular Path.

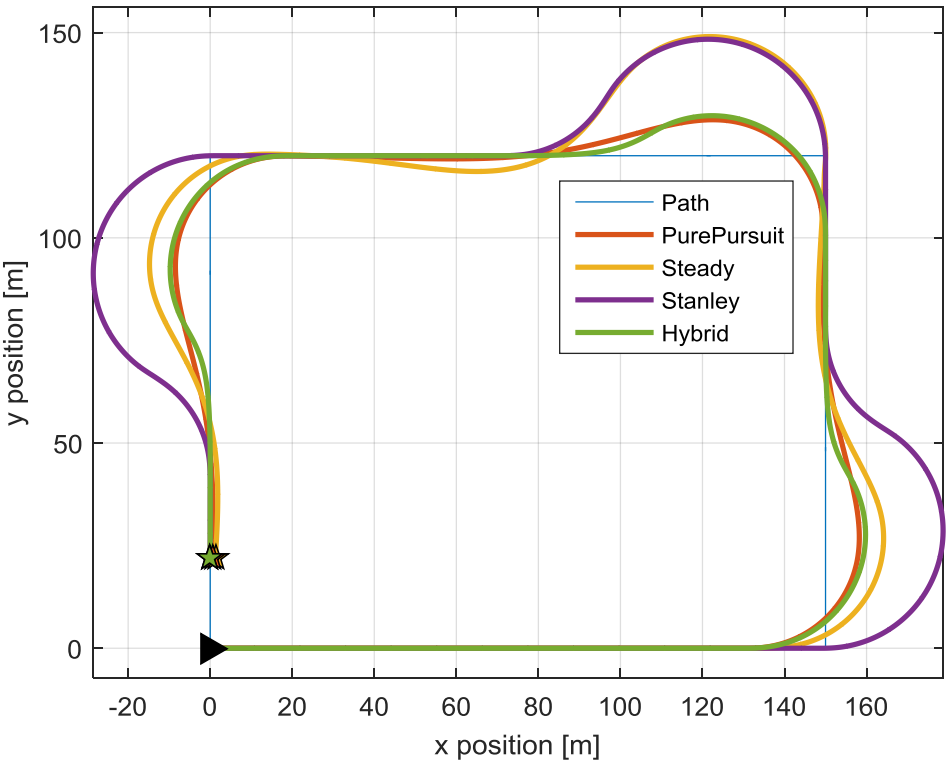

Figure 5.15 : Vehicle CG Position Changes with 80 km/h speed on 150x120 Rectangular Path.

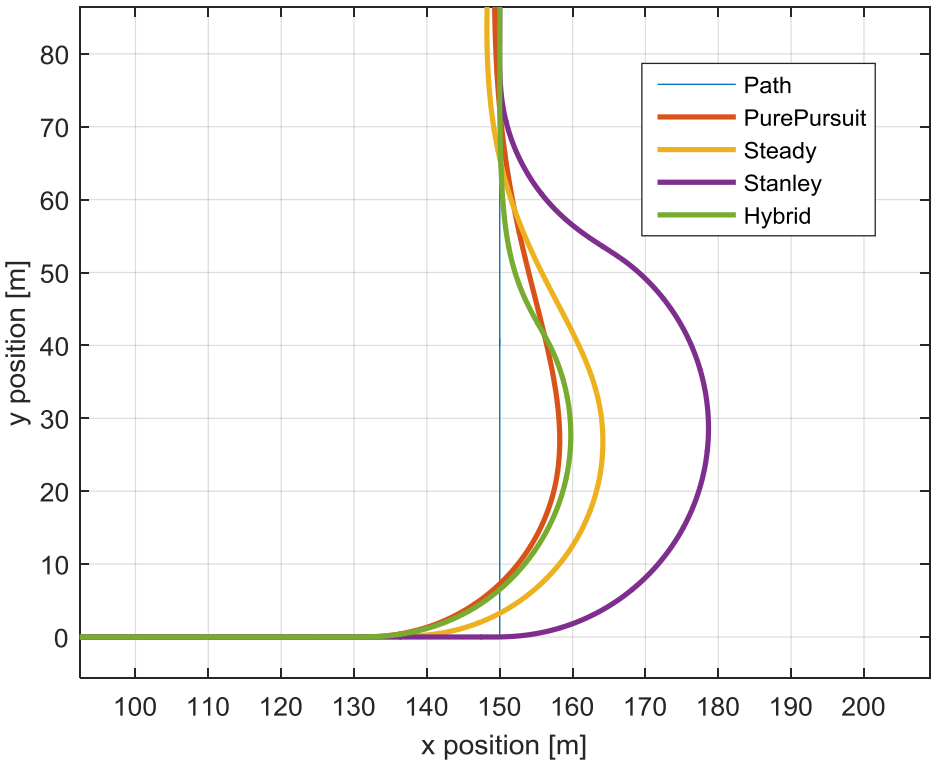

Figure 5.16 : Zoomed right bottom corner of Vehicle CG Position Changes with 80 km/h speed on 150x120 Rectangular Path.

5.3.3 Mixed path results

In order to see the effectiveness of the proposed hybrid controller, a mixed path is used for simulations as explained in the previous section. This path has sharp edges, smooth turns and sharp turns. In Figures 5.17, 5.18 and 5.19, turns with different characteristics can be seen with closer looks. On a smooth turn as in Area A1 and Area B, it can be seen that Stanley and Hybrid controller are the closest ones to the path. On sharp turns as in Area C and Area D, pure pursuit and the hybrid methods perform better. As it is seen from Table 5.3, in mixed path, the hybrid controller outperforms other methods regard to E_1 and E_2 error criteria. Nevertheless, it can be seen that Pure-Pursuit gets better in L_2 norm error criteria because of its cutting corner behavior as speed increases. The amount of error differences between the proposed hybrid method and the other methods increases especially at higher speeds.

Table 5.3 : Error values for Mixed Track.

150x120 Rectangular Track		Pure Pursuit	Stanley	Steady State	Hybrid
20 km/h	E_1	0.1051	0.0542	0.2024	**0.0424**
	E_2	47.9633	65.3639	80.0686	**27.8387**
50 km/h	E_1	0.3357	0.3644	0.5048	**0.1071**
	E_2	71.0077	187.4155	108.2300	**39.3902**
80 km/h	E_1	0.9917	2.2331	1.3696	**0.5954**
	E_2	**156.9142**	611.6856	250.7458	158.4374

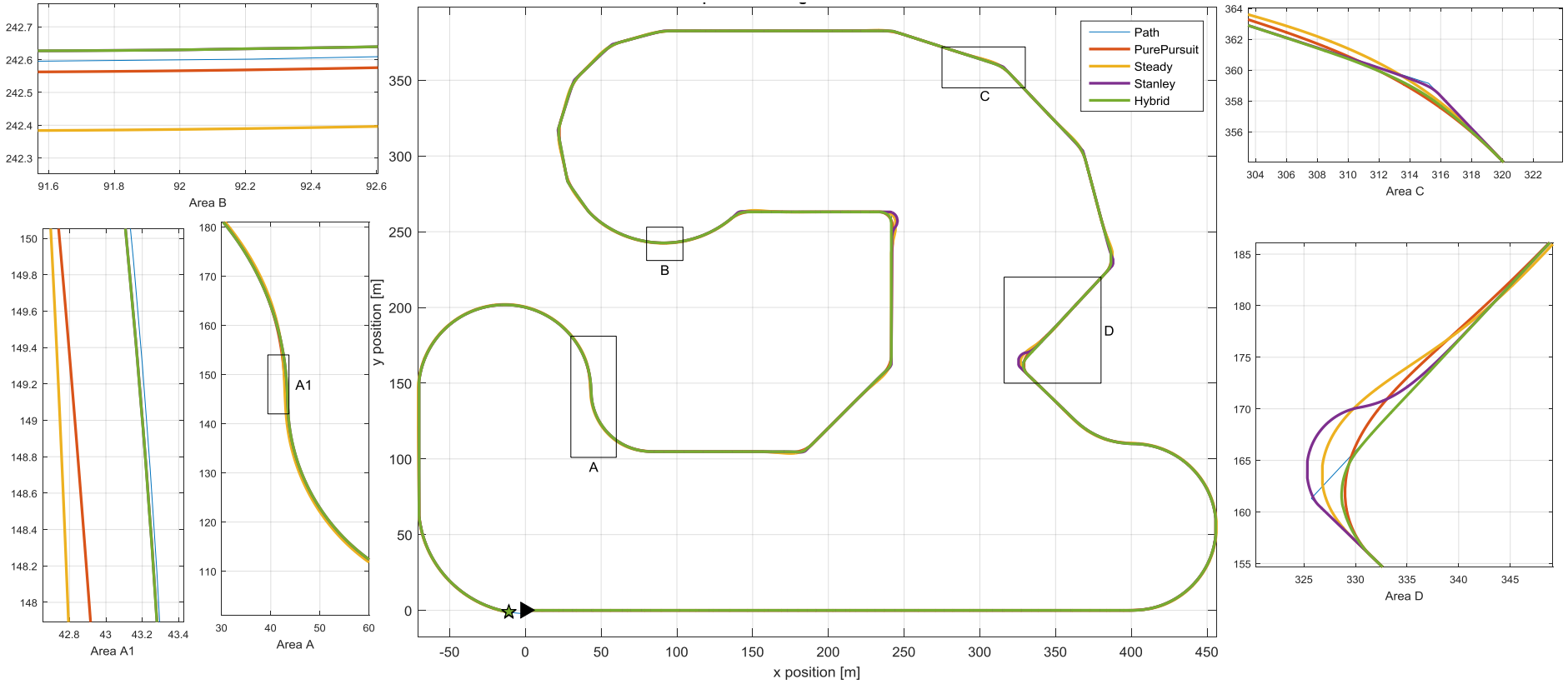

Figure 5.17 : Vehicle CG position changes with 20 km/h speed on mixed path and zoomed sections.

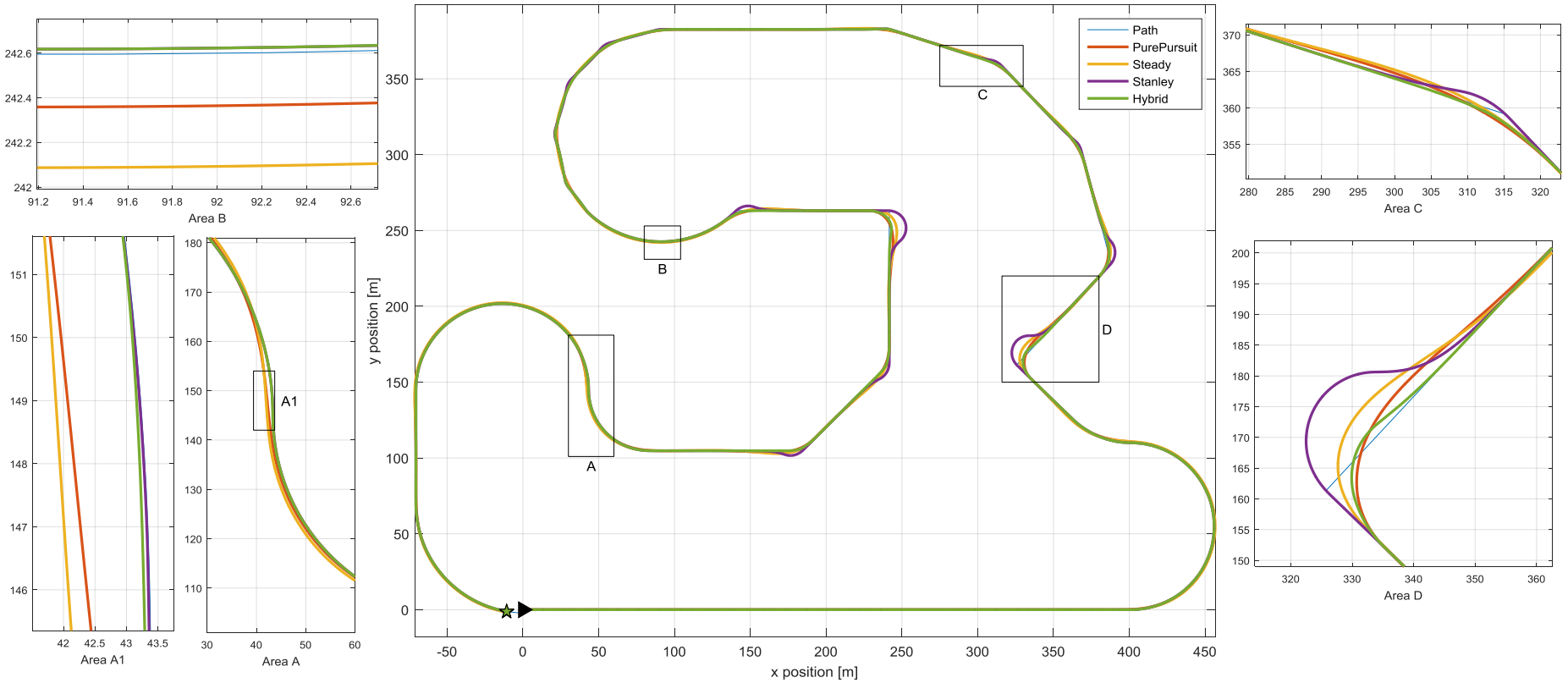

Figure 5.18 : Vehicle CG position changes with 50 km/h speed on mixed path and zoomed sections.

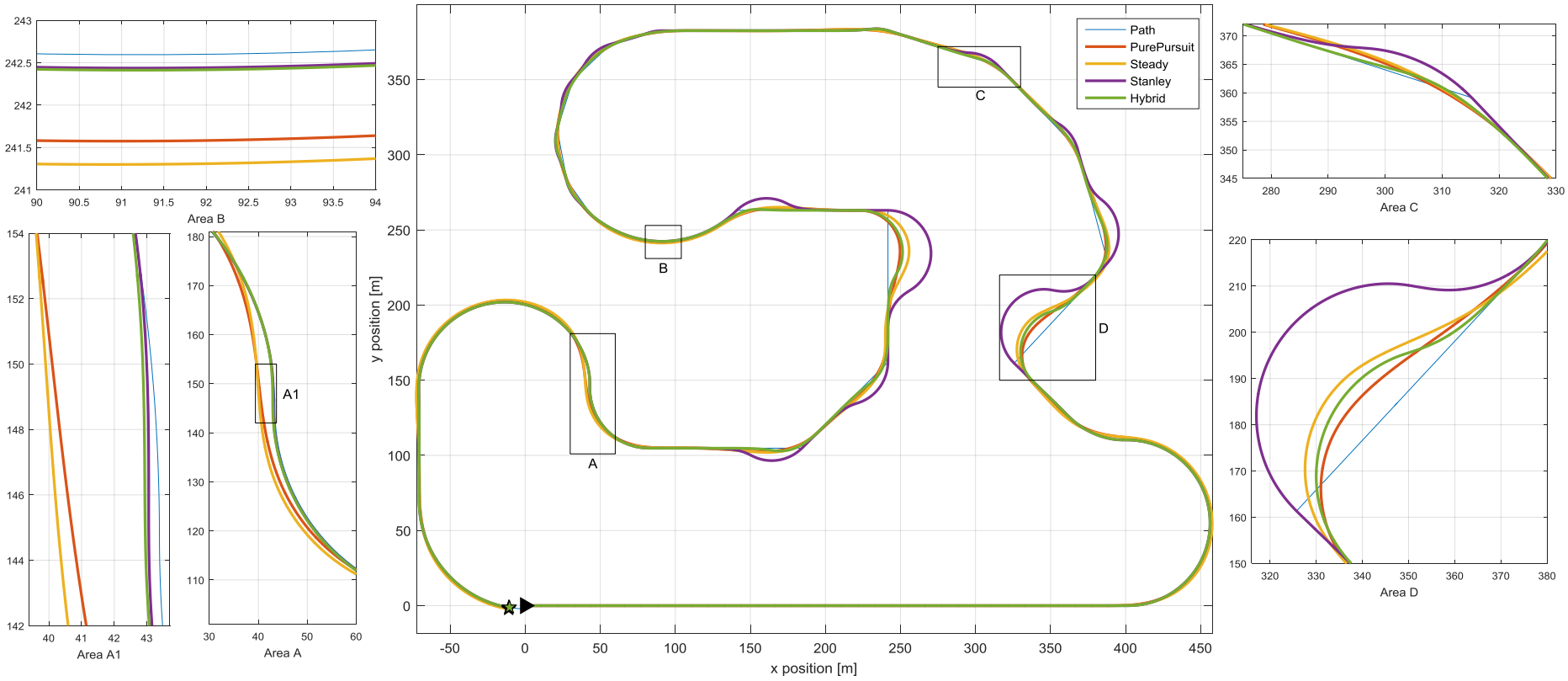

Figure 5.19 : Vehicle CG position changes with 80 km/h speed on mixed path and zoomed sections.

6. CONCLUSION AND DISCUSSION

In order to increase the performance of the controller on sharp edged path, two different path tracking method combined with a rule named *'hybrid controller'* is proposed. For this purpose, realistic vehicle models are given in detail, then some widely-used path tracking methods are investigated, then the hybrid controller is examined in detail, and finally, the simulations are performed in order to show the effectiveness of the proposed controller.

Due to Ackerman steering system, vehicle model can be presented by bcycle model. These realistic vehicle models are namely; kinematic bcycle, dynamic bcycle and linearized dynamic model. Dynamic bcycle model is chosen because it is the closest modelling amongst the others.

In this part, three different path tracking algorithms are explained in detail. First two methods are based on the conception of an ordinary driver trying to track a given path. While the third approach is based on a simplified mathematical model of the vehicle that is trying to follow a given path. In that sense, it can be said that first two are more intuitive than the latter one, and each method has its own strengths and weaknesses.

For this curvature problem of the path, two solutions can be proposed. The first one is to recalculate the path points for a smoother path. In this method, sharp parts of the path are converted to higher order polynomials and the path can be divided into pieces at sharp parts to generate more points. The second solution is to propose a control structure that deals with the sharp changes in the trajectory without changing the original path structure.

As mentioned before the look ahead methods are good at sharp changes in the trajectory and are bad at cutting corner behavior, and Stanley is vice versa. The proposed hybrid controller instantaneously uses Pure-Pursuit and Stanley Method. The weights of these methods change depending on the smoothness of the path ahead. As the path gets smoother, the weight of the Stanley method increases, if a sharp change is ahead the weight of the Pure Pursuit method increases.

Three different paths are created; a circular, a rectangular; and a mixed path. Each path is simulated with three different velocities. Under the assumption of constant longitudinal velocity, these speeds selected as20 km/h, 50km/h and 80 km/h.

Circle track is very smooth, in other words angle changes between path segments are very small. As expected, Errors for the Stanley method are small in comparison to the Pure-Pursuit and the Steady-State methods.

In the second track, sharp turns are tested with 90-degree turns. As it is expected Stanley does not start to steer until the vehicle gets very close to the intersection point. In other methods, because of the look ahead behavior steering is initiated earlier. When the results are evaluated, hybrid method responses faster than the others to minimize its errors. Pure Pursuit method's L2 norm gives the best result at the speed of 80km/h because the rise in the velocity causes more overshoot in Stanley and although the weight of Stanley in the hybrid controller is few, which is included in the overall calculation, it is observed that Stanley has more overshoot comparing to Pure Pursuit.

Lastly, the mixed-path simulations are done. Owing to the fact that this path has both smooth curvatures and sharp turns, and hybrid controller uses the strengths of both methods, it is observed that hybrdid method gives the best performance.

This thesis is summarized in conference paper and will present at the 3rd SICE International Symposium on Control Systems (ISCS) in JAPAN which is sponsored by IEEE and CSS.

In the future, in order to increase the performance of hybrid controller, weights and the timer value can be optimized. There might be further research on whether the change in the weight regarding to the smoothness of the path has an impact on the performance. In addition, hybrization can be done with other methods such as, vector pursuing, MPC and other controllers.

REFERENCES

[1] **Parissien, S. (2014).** The Life of the Automobile: The Complete History of the Motor Car. Macmillan.

[2] **Le Vine, S., Zolfaghari, A., & Polak, J. (2015).** Autonomous cars: the tension between occupant experience and intersection capacity. *Transportation Research Part C: Emerging Technologies,* 52, 1-14.

[3] **Pomerleau, D. A. (1989).** Alvinn: An autonomous land vehicle in a neural network (No. AIP-77). Carnegie-Mellon Univ. artificial intelligence and psychology project.

[4] **Xie, M., Trassoudaine, L., Alizon, J., Thonnat, M., & Gallice, J. (1993, May)**. Active and intelligent sensing of road obstacles: Application to the European Eureka-PROMETHEUS Project. In Computer Vision, 1993. Proceedings., Fourth International Conference on (pp. 616-623). IEEE.

[5] **Gibbs, S. (2016).** Tesla releases video of fully autonomous Model X electric car. the Guardian. Retrieved 23 November 2016, from https://www.theguardian.com/technology /2016/oct/20/tesla-releases-video-of-fully-autonomous-model-x-electric-car

[6] **Gibbs, S. (2016).** Google's self-driving car in broadside collision after other car jumps red light. the Guardian. Retrieved 23 November 2016, from https://www.theguardian.com/technology/2016/sep/26/google-self-driving-car-in-broadside-collision-after-other-car-jumps-red-light-lexus-suv

[7] **Davies, A. (2016).** Uber's Self-Driving Truck Makes Its First Delivery: 50,000 Budweisers. WIRED. Retrieved 23 November 2016, from https://www.wired.com/2016/10/ubers-self-driving-truck-makes-first-delivery-50000-beers/

[8] **Davies, A. (2016).** Everyone Wants a Level 5 Self-Driving Car—Here's What That Means. WIRED. Retrieved 23 November 2016, from https://www.wired.com/2016/08/self-driving-car-levels-sae-nhtsa/

[9] **Scopus (2016).** Scopus.com. Retrieved 23 November 2016, from https://www.scopus.com/

[10] **Park, M. W., Lee, S. W., & Han, W. Y.** (2014, October). Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In *Control, Automation and Systems (ICCAS), 2014 14th International Conference onn* (pp. 1443-1447). IEEE.

[11] **Snider, J. M.** (2009). Automatic Steering Methods for Autonmous Automobile Path Tracking (Rapor No. CMU-RI-TR-09-08) Technical Report Carnegie Mellon University

[12] **Hellstrom, T., & Ringdahl, O.** (2006). Follow the Past: a path-tracking algorithm for autonomous vehicles. *International journal of vehicle autonomous systems, 4(2-4), 216-224*

[13] **Wit. J. S.** (2000). *ector pursuit path tracking for autonomous ground vehicles* (Doctoral dissertation). Retrieved from http://www.dtic.mil/

[14] **Jalali, K., Lambert, S., & McPhee, J.** (2012). Development of a path-following and a speed control driver model for an electric vehicle. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems, 5(2012-01-0250), 100-113*

[15] **Ding, N., Zhang, Y., Gao, F., & Xu, G** (2013). A Gain-Scheduled PID Controller for Automatic Path Following of a Tractor Semi-Trailer.*SAE International Journal of of Commercial Vehicles, 6 (2013-01-0687), 110-117*

[16] **Ping, E. P., & Swee, S. K.** (2012). Simulation and experiment of automatic steering control for lane keeping manoeuvre.. *4th International Conference on Intelligent and Advanced Systems (ICIAS).* IEEE

[17] **Wu, S. J., Chiang, H. H., Perng, J. W., Lee, T. T., & Chen, C. J.** (2005). The automated lane-keeping design for an intelligent vehicle. *Intelligent Vehicles Symposium in IEEE Proceedings,508-513*

[18] **Song, P., Zong, C., & Tomizuka, M** (2015). Combined longitudinal and lateral control for automated lane guidance of full drive-by-wire vehicles. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems s, 8 (2015-01-0321), 419-424*

[19] **Rosolia, U., Braghin, F., Alleyne, A., & Sabbioni, E.** (2015). NLMPC for Real Time Path Following and Collision Avoidance. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems s, 8 (2015-01-0313), 401-405*

[20] **Ding, J., Li, K., & Hedrick, K..** (2015). A Robust Lane-Keeping 'Co-Pilot'System Using LBMPC Method. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems s, 8 (2015-01-0322), 219-228*

[21] **Mörtberg, H.** (2006). *Control and dynamic modeling of an Autonomous Ground Vehicle* (Master of Science Thesis). Swedish Defence Research Agency.SWEDEN

[22] **Rajamani, R.** (2011). *Vehicle dynamics and control*. Springer Science & Business Media..

[23] **Mathworks (2016).** Pure Pursuit Controller. Retrieved 23 November 2016, from https://www.mathworks.com/help/robotics/ug/pure-pursuit-controller.html

[24] **Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S.** (2007). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. *American Control Conference,* 2296-2301.

[25] **Ozguner, U., Unyelioglu, K. A., & Hatipoglu, C.** (1995). An analytical study of vehicle steering control.. *Proceedings of the 4th IEEE Conference on Control Applications ,125-130*

[26] **Waze (2016)** Interactive junction instruction algorithm Retrieved 23 November 2016, from https://wiki.waze.com/wiki/Interactive_junction_instruction_algorithm#roundabout_angle

CURRICULUM VITAE

Name Surname **:** Mertcan Cibooğlu

Place and Date of Birth **:** Beyoğlu/İstanbul, 12.12.1991

E-Mail **:** Mertcan.cibooglu@gmail.com

EDUCATION

- **B.Sc.** **:** 2014, Istanbul Technical Engineering, Control and Automation Engineering
- **B.Sc. (Second Major)** **:** 2015, Istanbul Technical Engineering, Aeronautical Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2015-Present: AVL Research and Development Turkey

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **M. Cibooğlu,** U. Karapınar, M.T. Söylemez 2017: Hybrid Controller Approach for an Autonomous Ground Vehicle Path Tracking Problem, In SICE ISCS 2017 - International Symposium on Control Systems, March 6-9, 2017, Okayama, Japan.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **M. Cibooğlu,** F.Çalışkan, 2015: Unknown Input Observer Based on Sensor Fault Tolerant Control for Unmanned Aerial Vehicle," TOK 2015 – Turkish National Meeting on Automatic Control, September, 2015, Denizli, Turkey, (in Turkish).