ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MULTI INPUT MULTI OUTPUT INTELLIGENT MODELING TECHNIQUES AND APPLICATION TO HUMAN DRIVER

M.Sc. THESIS

Emre TEKİN

Department of Mechatronics Engineering

Mechatronics Engineering Programme

JUNE 2012

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MULTI INPUT MULTI OUTPUT INTELLIGENT MODELING TECHNIQUES AND APPLICATION TO HUMAN DRIVER

M.Sc. THESIS

Emre TEKİN
(518091045)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Prof. Dr. Şeniz ERTUĞRUL

JUNE 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK GİRİŞ ÇOK ÇIKIŞLI AKILLI MODELLEME TEKNİKLERİ VE İNSAN SÜRÜCÜYE UYGULANMASI

YÜKSEK LİSANS TEZİ

Emre TEKİN
(518091045)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şeniz ERTUĞRUL

JUNE 2012

Emre-TEKİN, a **M.Sc.** student of ITU **Graduate School of Science, Engineering and Technology** student ID 518091045, successfully defended the **thesis** entitled "**MULTI INPUT MULTI OUTPUT INTELLIGENT MODELING TECHNIQUES AND APPLICATION TO HUMAN DRIVER**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Şeniz ERTUĞRUL** ....
İstanbul Technical University

Jury Members : **Prof. Dr. Serhat ŞEKER** ....
İstanbul Technical University

Assist. Prof. Dr. Pınar Boyraz ....
İstanbul Technical University

Date of Submission : 4 May 2012
Date of Defense : 8 June 2012

To my mother and father,

FOREWORD

Firstly, I would like to express my deep appreciation and thanks for my advisorProf. Dr. Şeniz Ertuğrul who supported and trusted me during my work. Without her guidance and foresee, I would be lost in many details throughout this study due to my willingness to achive an advanced work.

Also, I would like to thank my family due to their patience to my madness in the last painful months of this study. I would like to thank my brothers due to their understanding for rejecting their offers to have a good time outdoors in this last months.

This study has been difficult for me because I have been working at the same time. I also thank to my boss Canan Yılmazel and my collegue / teammate Özgür Tunçer for their understanding and help at workduring my study.

Finally, I would like to thank IPG© company for their supportby supplying their CarMaker® software free of charge in the context of this study.

June 2012

Emre TEKİN
Mechanical Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

AI	**:** Artificial Intelligence
ANFIS	**:** Adaptive NeuroFuzzy Inference System
ANN	**:** Artificial Neural Network
FCM	**:** Fuzzy C-Means
FMLE	**:** Fuzzy Maximum Likelihood Estimate
GA	**:** Genetic Algorithm
GG	**:** Gath-Geva
LM	**:** Levenberg-Marquardt
MIMO	**:** Multi Input Multi Output
MISO	**:** Multi Input Single Output
MSE	**:**Mean Squared Error
MTLA	**:** Minimum Tracking Learning Algorithm
NARX	**:**Nonlinear Auto Regressive with eXogeneous inputs
NFIR	**:**Nonlinear Finite Impulse Response
PV	**:**Percentile Variance
TS	**:**Takagi Sugeno

LIST OF TABLES

Page

LIST OF FIGURES

MULTI INPUT MULTI OUTPUT INTELLIGENT MODELING TECHNIQUES AND APPLICATION TO HUMAN DRIVER

SUMMARY

With advances in technology in the last century, it has been possible to create numerous machines and tools that make life easier for human beings. Many of these technological devices interact with human, so consideration of human interaction to operation of these machines is an important design criterion for both safety requirements and highly capable machine development. At this point, it is important to imitate human reactions beforehand and include this behavior as a part of design process.

Interaction of human driver with vehicle is one of the subjects that is being worked actively by researchers. The scope of this study is to obtain driver model to achieve longitudinal and lateral control of vehicle simultaneously without indulging into complex mathematical modeling. To achieve this goal, intelligent modeling techniques have been reviewed and their application to human driver modeling has been studied. The intelligent modeling techniques that show satisfactory performance are selected for modeling task. The selected modeling algorithms are trained by using real human driver input-output samples. Results for application of ANFIS(Adaptive Neuro Fuzzy Inference) and ANN (Artificial Neural Network) intelligent modeling techniques to driver modeling are discussed.

In this study, driver model is obtained as two separate sub-models for longitudinal and lateral control. Longitudinal controller is responsible to decide gas and brake pedal positions while lateral controller is responsible to decide steering wheel angle position. These two controllers works in parallel simultaneously like a single model. CarMaker® software is used as simulation environment in this study together with Logitech© Driving Force Pro® gaming console.

Inputs and outputs of the driver model is obtained with combination of statistical methods and expert knowledge. Correlation analysis is chosen among the statistical methods to identify simultaneously changing trends between selected inputs and outputs. Correlation analysis also helps to identify time delays of inputs and model order.

Validation of obtained driver models are performed in two different stages. In the first stage the outputs of training data and corresponding model outputs are compared with PV (Percentile Variance) technique and this comparison is repeated for validation data. Validation data is collected from real human driving in simulation environment. A different road profile is used to collect validation data. If the results for first stage are satisfactory, then the obtained models are embedded in CarMaker® software for second stage. The outputs of driver model are overwritten on CarMaker® driver block outputs. By IPG© CarMaker® movie animation the performance of driver model is checked to track the defined road track. It is seen that

the final model can successfully track the road profile that is different from the road profile where data to train intelligent models is collected.

In Section 6, the derived and conclusions and recommendations for future studies are presented. Also, two newly proposed MIMO hybrid intelligent modeling structures are discussed in this section with the further work needed for their practical application on driver model.

ÇOK GİRİŞ ÇOK ÇIKIŞLI AKILLI MODELLEME TEKNİKLERİ VE SÜRÜCÜYE UYGULANMASI

ÖZET

Günümüzde teknolojinin ilerlemesiyle birlikte insan yaşamını kolaylaştıran birçok makine ve araçlarının geliştirilmesi mümkün olmuştur. Bu ileri teknoloji araçların büyük bir bölümü insan ile etkileşimli çalışmaktadır. Bu nedenle yüksek güvenlik ve performansa sahip teknolojik araçların tasarımı ve geliştirilmesi aşamasında insan-makine etkileşimi önemli bir tasarım parametresi haline gelmiştir.

İnsan sürücünün taşıtla etkileşimi, insan-makine etkileşimi alanında aktif olarak çalışılan bir konudur. Motor ve taşıtın geliştirilmesi için yapılan ARGE çalışmalarında bilgisayar ortamında HIL(Hardware In the Loop) sistemleri ile uyumlu olarak kapalı çevrim çalışabilen ve insan sürücüyü taklit eden modellere ihtiyaç duyulmaktadır. Böylece, uzun veya fiziksel olarak denenmesi zor olan sürüş senaryoları robot sürücüler ile test edilebilinecektir.

Bu çalışmanın amacı, uğraştırıcı matematiksel modelleme çalışmalarına girilmeden taşıtın doğrusal ve yanal kontrolünü sağlayacak şekilde insan sürücü modelini elde edebilmektir. Bu amaca ulaşabilmek için akıllı modelleme yöntemleri araştırılmış ve bu yöntemlerin insan sürücü modelleme problemine uygulanması üzerinde çalışılmıştır. Bu yöntemler arasından, insan sürücü modellemesi için başarılı olanlar seçilmiş ve çıkarılan modellerin performansı test edilmiştir.

İnsan sürücünün doğrusal olmayan yapısı ve yanlılığından ötürü modelleme işi oldukça zorlaşmaktadır. Bu bağlamda, problem basitleştirmek için sürücü modeli doğrusal ve yanal olmak üzere iki ayrı kontrolör olarak çıkarılmıştır. Böylece çok giriş-çok çıkışlı olan insan sürücü modeli, çok giriş-tek çıkış olan iki alt modele ayrılmıştır. Ayrıca, sistemi bu şekilde basitleştirmek kullanılan akıllı modelleme teknikleri sayesinde insan sürücünün yanal ve doğrusal dinamiklerinin birbirinden bağımsız olarak fiziksel hassasiyetlerinin anlaşılmasında etkili olmuştur Bu iki modelin kapalı çevrim benzetimlerde birlikte çalışarak çok giriş-çok çıkışlı genel insan sürücü modelini temsil etmesi amaçlanmış ve amaca ne kadar ulaşıldığının anlaşılması için modellerin performansı test edilmiştir. Benzetimin trafiksiz ortamda, otomatik şanzımanlı taşıt ile ve eğimsiz yol profili ile yapılması çalışmadaki diğer basitleştirmelerdir.

Çalışmada benzetim ortamı olarak IPG© CarMaker® yazılımı ve Logitech© Driving Force Pro® oyun konsolu kullanılmıştır. Böylece insan sürücünün bu sanal benzetim ortamında direksiyon simidi, gaz ve fren pedallarını kontrol ederek ve görsel video geri beslemesi ile taşıtı kullanabilmesi sağlanmıştır. Bu kurulum ile insan sürücü eylemleri ve taşıt-motor dinamikleri veri olarak kaydedilmiştir. Alınan kayıtlar daha sonra veri tabanlı olan akıllı modelleme ve sistem tanılama yöntemleri ile insan sürücü modelinin elde edilmesinde kullanılmıştır. Sürücünün taşıtı kumanda edebilmesi Logitech© Driving Force Pro® oyun konsolu aracılığıyla

başarılabilmiştir. Simulink® üzerinde geliştirilen bir ara yüz model ile oyun konsolunun CarMaker® yazılımına tanıtılması sağlanmıştır. Daha sonra ara yüz modelinin çıkışları CarMaker® sürücü bloğunun girişleri üzerine yazılarak sürüş ortamının yazılım ve donanım kısmını kapsayacak şekilde oluşturulması tamamlanmıştır.

Çalışmada insan sürücü modelleme problemine uygun olacağı düşünülen akıllı modelleme yöntemleri incelenmiş ve bu yöntemlerin sürücü modeline uygunluğunu sınamak amacıyla toplanan sürücü verileri üzerinde uygulaması yapılmıştır. İncelenen akıllı modelleme teknikleri arasında ANFIS (Adaptive Neuro Fuzzy Inference) ve ANN (Artificial Neural Network) algoritmaları uygulamadaki basitliği ve tatmin edici performansı nedeniyle sürünün yanal ve doğrusal kontrolör olarak modellenmesinde tercih edilmiştir. ANFIS, TS(Takagi Sugeno) bulanık mantık yapısının parametrelerinin toplanan veriler ile otomatik olarak en iyileştirilmesini sağlamaktadır. ANFIS yapısındaki bu en iyileştirmede maliyet fonksiyonundaki doğrusal parametreler için en küçük kareler metodu kullanılırken, doğrusal olmayan üyelik fonksiyonu parametrelerinin en iyileştirilmesi için hataların geriye yayılımı algoritması kullanılarak hibrit bir algoritma oluşturulmuştur. ANN yapısında ise Levenberg-Marquardt en iyileştirme algoritması, ANN yapısında nöronlar arası ağırlık faktörlerinin eğitilmesi ile yapılmaktadır. Ayrıca, bu çalışma kapsamında geliştirilen iki yeni hibrit akıllı modelleme yöntemlerine ait yapıların detayları Bölüm 6'da anlatılmaktadır.

Bu çalışmada çıkarılan yanal ve doğrusal sürücü modellerinin performansının yüksek olması için en fazla dikkat edilmesi gereken hususlardan biri sürücünün dinamik karakteristiklerini yansıtan doğru giriş-çıkış bileşenlerinin seçilmesidir. Seçilen giriş-çıkış bileşenlerinin sürücü karakteristiğini yansıtmaması durumunda kullanılan akıllı modelleme yönteminin başarısının ne kadar yüksek olduğunun bir önemi kalmamaktadır. Doğru giriş-çıkış bileşenlerinin belirlenmesi için ilk aşama zengin verinin elde edilmesidir. Zengin veri sürekli uyaranlık ve mümkün olan en geniş frekans bandında bileşenlere sahip olma özelliklerini taşımalıdır. Çalışmada insan sürücünün kapalı çevrimde olduğu sanal sürüşler benzetim ortamında gerçeklenmiş ve toplanan verilerin zengin veri olma özelliği analiz edilmiştir. Yapılan analizler sonucunda seçilmeye aday her giriş verisinin en az 50. mertebeye kadar sürekli uyaran olduğu anlaşılmış ve bu mertebe seçilen model mertebeleri göz önüne alınarak yeterli bulunmuştur. Ayrıca toplanan verilerin frekans bileşenleri güç spektrumunda analiz edilmiş ve toplanan verilerin frekans bandının sürücü dinamiğinin frekans bandının tamamını uyaracak kadar geniş olmadığı görülmüştür. Bununla birlikte, sürücü modelinin çıkarılmasında fiziksel olarak gerçek olmayacak kadar çeşitli varyasyon içeren senaryolar kullanılmasından dolayı çıkarılan sürücü modelinin gerçek koşullara yakın senaryolarla benzetimi yapılması durumunda daha geniş frekans bandında uyarılmayacağı ve bu nedenle başarılı olacağı düşünülmüştür. Nitekim gerçek koşullara yakın şartlarda yapılan sınama çalışmalarında sürücü modelinin yüksek performans göstermesi bu öngörüyü desteklemektedir.

Sürücü modeli için uygun giriş-çıkış bileşenlerinin belirlenmesi için belirli bir sistematik bulunmamaktadır. Literatürdeki mevcut istatistikî yöntemlerin birçoğu tek giriş-çıkışlı sistemler için uygun giriş-çıkış grubunun belirlemeye yöneliktir. Bununla birlikte bu yöntemler belli bir ölçüde doğrusal olmayan çok giriş-çok çıkışlı sistemlerin aday giriş-çıkışlarının belirlenmesinde fikir sahibi olunmasını sağlamaktadır. Bu çalışmada bu amaca yönelik olarak korelasyon yöntemi kullanılmıştır. Bu yöntem iki sinyalin eş zamanlı olarak benzer eğilimle değişmesinin ölçüsünü vermektedir. Her ne kadar eş zamanlı olarak benzer karakter gösteren iki sinyal arasında nedensellik bulunması kesin olarak söylenemese de, korelâsyonun yüksek bulunması bu ihtimali artırmakta ve incelenen girişin model için bir aday olmasını sağlamaktadır. Ayrıca korelasyon yöntemi ile seçilen girişin ayrık zamandaki geçmiş örneklemelerinden hangisinin çıkışı daha fazla uyardığı da tahmin edilebilmekte ve böylece model mertebesinin de belirlenmesi sağlanmaktadır. Ayrıca, giriş-çıkış bileşenlerinin seçilmesinde modelleyicinin yani uzmanın, da sistem hakkında bilgisi istatistiki korelasyon yöntemini desteklemektedir. Mesela, sürücünün yola göre pozisyonu ve sapma açısının yanal kontrol çıkışı olan direksiyon simidi açısını etkileyeceği önceden tahmin edilerek bu girişler ve gecikmelerinin direksiyon simidi üzerindeki etkisi korelasyon analizi ile incelenmektedir. Ayrıca, aslında sistem girişi olmamasına rağmen çıkış ile yüksek korelasyon gösteren bazı girişler yine uzman bilgisi ile sistem giriş adaylığından elenmektedir. Mesela, yanal savrulma yönündeki ivmelenme ve direksiyon simidi açısı eş zamanlı benzer eğilimle değişim gösterdiğinden aralarında yüksek korelasyon bulunmaktadır, Fakat, bu ivmelenme aslında direksiyon açısının aslında nedeni değil sonucudur ve bu nedenle yanal kontrol girişi adaylığından elenmiştir. Doğrusal kontrol modelinin çıkışı için farklı bir yaklaşım düşünülmüş ve çıkış gaz ve fren pedal pozisyonlarının farkı olarak alınmıştır. Böylece, yanal sürücü kontrolör modellemesinde gaz ve fren çıkışları iki ayrı model olarak değil de tek bir model olarak çıkarılmıştır çünkü gaz ve fren pedal pozisyonlarının benzer giriş bileşenlerinden etkilendiği düşünülmektedir. Bunun sonucunda model çıkışında gaz ve frene aynı anda basılması engellenerek gerçek sürücü karakteristiğinden uzaklaşılmamıştır. İnsan sürücüden toplanan verilerde gaz ve fren pedalından birinde basıldığında diğerinin pozisyonu sıfır'da kalmaktadır. Bu nedenle doğrusal kontrol modelinin çıkışı olan (gaz pedal pozisyonu-fren pedalı pozisyonu) bileşeni sıfır'dan büyükse benzetim ortamında gaz pedalı pozisyonu olarak alınırken, sıfır'dan küçükse fren pedalı pozisyonu olarak alınmıştır.

Çıkarılan sürücü modellerinin sınanması iki aşamadan oluşmaktadır. İlk aşamada insan sürücünün sanal benzetim ortamında aracı kontrol etmesiyle toplanan veriler çevrim dışı olarak modellemede kullanılmıştır. Daha sonra bu verilerle eğitilen modelin çıkışlarının, eğitim verileri ile yüzde uyumu incelenmiştir. Aynı zamanda çıkarılan modellerin performansının modellemede kullanılan eğitim verilerinden bağımsız olan veri kümesi ile de sınanması gerekmektir. Bu bağlamda modellemede kullanılan yol profilinden farklı olan yol profilinde insan sürünün aracı kontrol etmesiyle benzetim ortamında modellemeden bağımsız bir veri toplama işlemi gerçekleştirilmiştir. Eğitimde kullanılan yol profili mümkün olduğu kadar çeşitli eğrilik yarıçaplarına sahip yol parçalarının birleşiminden oluşmakta iken, bağımsız veri kümesinin elde edilmesi için kullanılan yol profili CarMaker® yazılımı üzerinde gelen standart yol profillerinden biri olarak seçilmiştir. Sınamanın bu aşaması 2. aşamaya geçilmeden önce bir hazırlık mahiyetindedir ve çıkarılan model performansının yeterliliği hakkında bir ilk tahmin sağlamaktadır. Performansın kabul edilemez bulunması durumunda giriş-çıkış bileşenlerinin seçimi ve modelleme algoritmasının tasarım parametreleri tekrar gözden geçirilmektedir. Böylece daha uğraştırıcı olan 2. aşamaya geçilmeden önce zaman tasarrufu sağlanmıştır. İlk aşama değerlendirmesi sonucunda yanal kontrol için eğitim verileri ile model çıkışı arasında 86%'ya ve doğrusal kontrol için 68%' e kadar yüzde uyum görülürken, yanal kontrol için bağımsız veri kümesi ile model çıkışı arasında 53%'e ve doğrusal kontrol için 24%'e kadar yüzde uyum görülmüştür. Sınamanın ikinci aşamasında çıkarılan sürücü modelleri Simulink® bloklarına dönüştürülmüştür. Oluşturulan bu bloklar

CarMaker® üzerinden benzetim ortamını oluşturan genel Simulink® yapısı içerisine gömülmüştür. Daha sonra sürücü modellerinin bağımsız veri kümesini toplamak için seçilen standart CarMaker® yol profilini otomatik olarak kontrol etmesi beklenmiştir. Sürücü modellerinin aracı kontrol etmedeki başarısı CarMaker® yazılımının video animasyonu üzerinden gözlenmiştir. ANFIS ve ANN ile oluşturulan bu modellerin tasarım parametreleri ve giriş-çıkış bileşenlerinin son seçimine karar verilene kadar iki aşamadan oluşan bu iterasyon devam etmiştir. Son seçilen model yapısında ANFIS ve ANN ile oluşturulan sürücü modelleri aracın doğrusal ve yanal kontrolünde oldukça başarılı olmuş ve insan sürücünün önceden sürüş yapmadığı yol profilini başarı ile takip edebilmiştir.

Çalışmanın gelecek aşamalarında neler yapılabileceği hakkında öneriler bölüm 6'da verilmiştir. Bu bölümde yapılan çalışma neticesinde çalışmadaki hedef olan doğrusal ve yanal kontrolü sağlayan sürücü modelinin başarılı bulunmasına ek olarak, sürücünün dinamik karakteristiği hakkında edinilen bilgiler de sunulmuştur. Bu bölümde ayrıca çalışma kapsamında geliştirilen fakat pratik uygulama için daha fazla geliştirilmesi gereken iki yeni hibrit akıllı modelleme yapısından da bahsedilmiştir.

1. INTRODUCTION

With advances in technology in the last century, it has been possible to create numerous machines and tools that make life easier for human beings. Many of these technological devices interact with human, so consideration of human interaction to operation of these machines is an important design criterion for both safety requirements and highly capable machine development. At this point, it is important to imitate human reactions beforehand and include this behavior as a part of design process.

Human-machine interaction is one of the most attractive areas for researchers to achieve different goals. The control actions of human operators have been intensively studied since 1940s to compare pilot ratings, develop new control methods based on human response, improve vehicle handling quantities, etc. These studies can be classified into three groups: (i) information, (ii) control and (iii) decision (Ertuğrul, 2009).

1.1 Purpose of Thesis

In this project, it is aimed to model human driver actions as a closed loop controller. For this purpose, intelligent modeling techniques in literature are reviewed and some hybrid models are developed for modeling human driver as a closed loop controller that interacts with dynamic vehicle models in simulation environment.

It is not intended in this study to develop complex mathematical models of human driver as done in literature since human driver is a complex nonlinear system, so theoretical modeling is expected to be tedious work, which generally results in empirical single input-single output models as in Plöchl and Edelmann(2007). Intelligent modeling techniques used in this study are based on data acquisition from real human driver so that human driver model can learn from data samples to imitate human driver.

In Mitschke (1996) , it is stated that there are two important aspects of human driver;

- Longitudinal control
- Lateral control

Longitudinal control consists of determining position of gas pedal, brake pedal and choosing appropriate gears. This determination is based on the distance to vehicle ahead and approaching speed. Shape of the road and targeted road track are the components that affect lateral control (Atabay, 2004; Delice,2004). In this thesis, lateral and longitudinal control aspects are studied as coupled multi-input single-output modeling.

Following simplifications are considered for modeling;

- Gear information is removed from modeling considering the trend to automatic transmission.
- Road inclination is not included in modeling. All data is collected in straight road.
- Traffic is not simulated in modeling, tracking the road shape by imitating reference human driver is considered.

1.2 Literature Review

Determining human driver dynamics is a difficult task because of human's internal randomness and vagueness. It is generally the case that human driver does not respond in the same way to same situations, which makes modeling task extremely hard. Prometheus (PROgraMmme for a European Traffic with Highest Eficiency and Unprecedented Safety), PATH Research (Partners for Advanced Transit and Highways) and ALVIN (Autonomous Land Vehicle In a Neural Network) in USA, , ASV (Advanced Safety Vehicle) in Japan make studies for human driver models. These studies are based on several purposes, which can be exemplified as traffic safety, reducing environment pollution, reducing driver load and controlling Traffic flow (Delice, 2004).

In the world, number of vehicles is growing as time passes, but new road construction is not increasing proportionally. This causes an unavoidable traffic jam and increasing number of accidents in turn. In Fenton (1994), it is stated that traffic accidents cost 70 billion dolars a year, which is expected worse in today's time because of increased traffic. Development of a virtual driver may be helpful in assisting and warning the real human driver for possible accident scenerios.

In Shim et al.(1995),Lateral control action of human driver is examined. a model for closed-loop simulation of a driver-vehicle-road system that employs the visual data processing method is suggested. In the model of a human driver, the human capability of learning to negotiate the curved road is simulated using a neural network technique. The learned neural network extracts an equivalent curvature from the viewing image of the frontal road. This equivalent curvature becomes a control cue for steering action. To validate the model of the driver-vehicle-road system, an actual driving test is performed and the results are compared with those from the computer simulation of the model.

In Fujioka et al. (1991), an Neural network based driver model is both trained by a simple driver model that maps the steering angle as a function of the time delayed lateral deviation and heading angle, and by a human driver in a simulator environment. The input to the Neural network is the lateral deviation from the desired course and is derived from a simplified view at four points 20, 40, 60 and 80m ahead of the vehicle; four views are memorized at each time step resulting in 16 data points for the complete input. Required manoeuvres for teaching the neural network are analysed and the accuracy of the neural network is compared.

The goal in MacAdamand Johnson (1996) is the identification of driver steering behaviour, based upon neural network. The relative lateral displacements si (i = 1, 2, 3) between the forward projection of the vehicle's longitudinal body axis and the road border,which can be seen in Figure 1.1, represent the previewed sensor inputs at time t . One sensor is placed very near to the front end of the vehicle, and the remaining two samples are located at two points ahead of the vehicle. By this specific choice of sensor location and measurements, information about vehicle motion and orientation with respect to the roadway is provided to the neural network. To extract further time derivative information from the same basic sensor measurements, time-delayed versions of these signals are utilized. Thus, the augmented set of network inputs, which include $\{s1(t), s2(t), s3(t)\}$ and two corresponding sets of their time delayed responses $\{s1(t - \tau1), s2(t - \tau1), s3(t - \tau1)\}$, $\{s1(t - \tau2), s2(t - \tau2), s3(t - \tau2)\}$ with $0.1 \leq \tau1 \leq \tau2 \leq 0.6$, also contains the key feedback cues required by 'classical' driver models for path regulation.

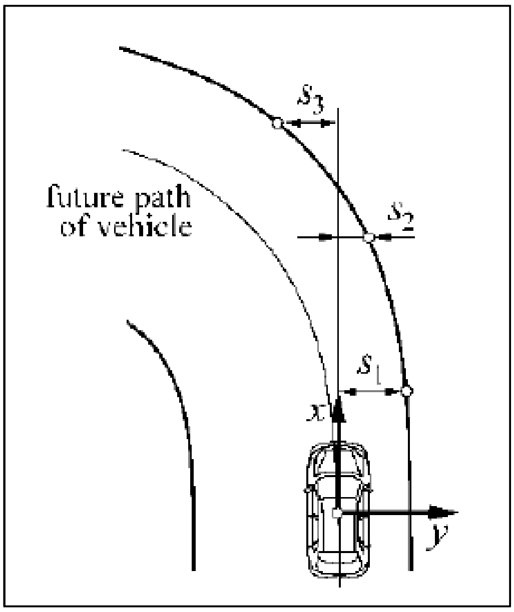

Figure 1.1 : Radar sensor measurements, adapted from (MacAdam and Johnson, 1996).

In (Tunçer, 2010), it is stated that a camera model is developed and integrated to CarMaker® driver simulator. Road preview data is acquired from the camera model to achieve lateral control of vehicle at different constant vehicle speeds with PID control architecture.

Delice (2004) used Adaptive Neural Network (ANN) architecture to train driver model based on data collected from real human driver via driving simulator. This study has a different aproach for driver model in the way that lateral and longitudinal control are combined as a single driver model. The perfromance of obtained driver model is validated based on data collected from the simulator with a different road profile than the one used for collection of training data. Offline validation results are considered satisfactory, but the driver model is not embedded in driving simulator for online simulation of the driver model.

One of the most important inputs for driver model is road information ahead of the driver. Two different approaches for feeding road information to the driver exist(Delice, 2004);

- Look down distance information
- Look ahead distance information

In look-down method, the driver model gets the distance between front end of the car and lane line. In look ahead method, driver gets road information from some road points ahead of him. In their study, Pilutti and Ulsoy (1999) estimates transfer function of driver by spectral analysis using real driver data. They observe that driver has approximately 40 dB/dec lead behavior, which shows the driver is looking ahead. It can be said that looking ahead distance is expected to be dependant on vehicle speed as in real driver, namely, looking ahead distance should be increasing with increasing vehicle speed because of decreasing view of sight. (Delice, 2004)

There are also studies in literature considering fuzy modeling approach. Sugeno andNishida (1985) have described fuzzy modeling for handling the vehicle trajectory. Kramer (1985) proposed to use a fuzzy model for the representation of perception characteristics of the driver. Yan and Weng (1988) suggested a linguistic controller related to the behavior of a human operator. This tooling has integrated fuzzy sets with human preview control. Optimal preview control model is built which reflects the human operator's planning, decision activities and his inexact or fuzzy knowledge about the system under control. A performance index is introduced to explain human operator's open-loop control behavior. Application is made to driver-vehicle system by building an internal model of the driver.

Most typical applications of fuzzy methods for driver modeling inliterature can be divided into following categories (Ivanov and Platz, 2011);

- Models of driver feeling of vehicle dynamics parameters such as trajectory, velocity, and road friction.
- Identifications and classifications of the drivers regarding fatigue, emotions, behavior, and other human attributes, including the procedures of the driver state recognition through visual processing.
- Simulation of driver reasoning both for conventional vehicles and for controllers of semi-automated, automated, and unmanned ground vehicles.
- Models of driver actions on vehicle controls (brake and throttle pedals, steering wheel) for authentic simulation of vehicle manouvers on driving simulators, hardware-in-the-loop and model-in-the-loop technique.
- Architecture and hardware part of driver assistance systems and devices of human machine interface.

- Advisory functions to support the driver in eco-driving and low-emission driving.

From classical control theoretic models point of view, Tustin (1947) performed an important study by trying to treat human operator as a combination of linear model and remmant (the remaining difference between approximation and real controller) The idea of Tustin was to find a description of the remmant properties which is independent from the experiment set-up, for example, in a stochastic sense. This study shows human control behaviour may not be produced at large by means of mathematical and system theoretic methods because human behaves differently with different plants and different types of inputs. Also, there are vast inter-individual differences as well as intra-individual variations interfere with a well-defined model. At this point, fuzzy modeling studies mentioned above can be powerful to some extent as this type of modeling aims to reflect human behavior.

It is worth mentioning about quasi-linear model since it is up to now the most general linear model of human as a controller. Following Tustin's work, McRuer and Krendel (1959) formulated quasi-linear model Equation(1.1). Quasi-linear model describes the human by means of a linear second order differential equation with dead time. Differences in task environment and in operator properties are described by means of five free parameters. τ (dead time=reaction time)and T_N (neuromuscular delay time) areconsidered as physiological parameters independent from the task. The parameters K, T_L, and ,T_I (Table 1.1) depend on the control plant and type of stochastic input signal. The dependency on the input signal and the term $e^{-i\tau s}$ casusing transcentality are the main reasons which causes the term "quasi".

$$H(s) = \frac{K.(T_L s + 1)}{(T_I s + 1)(T_N s + 1)} e^{-i\tau s} \tag{1.1}$$

Table 1.1 : Meaning of parameters of quasi linear models

τ	Reaction time
T_N	Neuromuscular delay
K,T_L,T_I	Adaptation to the controlled plant

A driver model should have specific properties for practical applicability (Delice, 2004);

- It should be adaptable to different driving conditions, namely, should be able to learn from those different conditions.
- The model should be independent from road and maneuvers so that it should be able to adapt new road conditions even if training of model is performed in different road conditions.
- Data used for driver modeling should be acquired from real human drivers, so that imitating human behavior can be assessed.

Determination of input-output parameters is one of the most important part of modeling procedure. In Anet al. (1994), An and Harris (1996), different inputs are chosen for car parking driver model and linear driver model (Delice, 2004). Input-output selection should be based on type and aim of driver model. If unnecessary inputs are added to the model, training of the model will be costly and accuracy may be affected negatively. If necessary inputs are omitted this will directly affect accuracy in case of practical application even if training performance seems good.

2. SIMULATION ENVIRONMENT

2.1 Software

IPG© CarMaker® commercial software is used for application of developed models and algorithms in this study. CarMaker® is a simulation environment based on MathWorks© MATLAB® and Simulink® software. CarMaker® is high degree of freedom, high confident commercial software which makes use of realistic vehicle models. Thanks to its realistic vehicle models, CarMaker® is used by many companies, universities and research centers.

CarMaker® is capable of simulating interaction of vehicle, driver, tires, road, and environment with detailed subsystems and components. Some of those subsystems can be replaced with real hardware like HIL (Hardware In the Loop) systems, gaming consoles or can be replaced with algorithm modules designed by user. As an overview, the software consists of virtual vehicle environment and user interface components as shown in Figure 2.1. The user interface can be used for starting and ending simulation, definition of vehicle parameters, visualization of results, animation of test drive, etc.

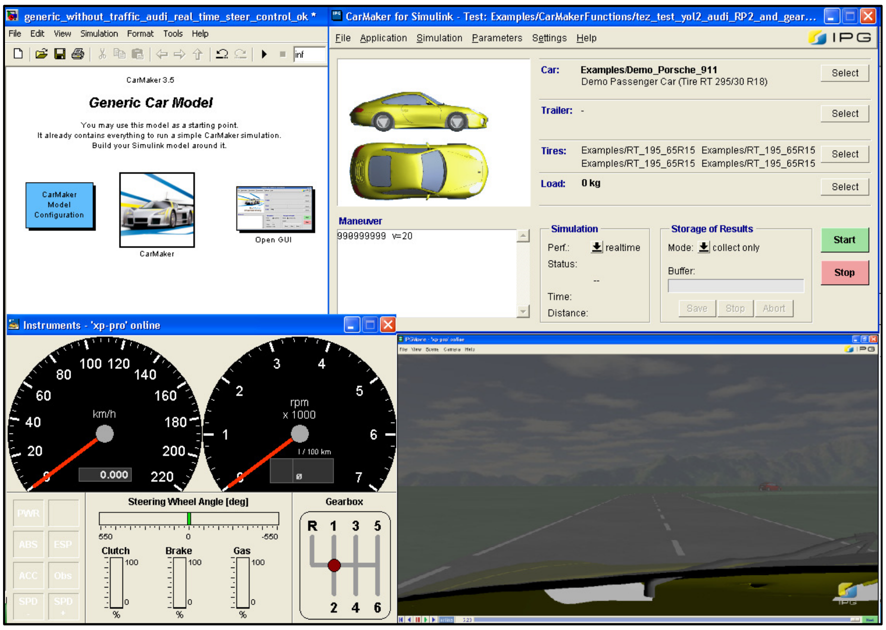

Figure 2.1 : CarMaker® user interface.

Road profile including curvature, right and left turns, lateral and longitudinal slope can be defined in the software numerically. Bumps, markers such as speed limit, stop, and wind as well as movie objects such as sign plates and tree strips can be added for realistic simulations. Moving traffic objects like vehicles traveling with predefined manuvers can be simulated. Also, Stationary traffic objects like surrounding buildings and traffic lights can be added.

Several sensors can be used in the software for virtual data acquisition, some of important ones are;

- Driver assistance sensors: these sensors work like radar system and detects existence of and distance to surrounding traffic objects and approaching speed to those traffic objects.
- Road preview sensors: these sensors can acquire road information ahead of vehicle at predefined distance. Road bend, deviation angle, deviation distance, lateral and longitidunal slope are some of the road information that can be acquired with these sensors.
- Body sensors: These sensors are used for acquisition of vehicle position, speed, and acceleration in different positions.

Virtual vehicle models in CarMaker® consist of basic engine dynamics, tire dynamics, steering wheel dynamics, longitidunal and lateral dynamics models.Alos, virtual vehicle includes mathematical models which define vehicle dynamics and kinematics. The mathematical models work parametrically by using design properties of vehicle being simulated.

Driving scenerios can be predefined by maneuver steps. virtual IPG driver packages or external driver inputs can be used for control of vehicle in simulation.IPG driver simulates behavior of real driver by deciding steering wheel angle, and brake pedal, gas pedal and clutch pedal positions. It is possible to log mored than 600 parameters during a virtual test drive in CarMaker®.

CarMaker® is based on MathWorks© MATLAB® and Simulink® software. Submodels and algorithms of CarMaker® are designed with Simulink® blocks. User can integrate his own submodels like controllers and driver models (Figure 2.2).In CarMaker® software, both available vehicle models of different trades and sizes or user parametrized vehicle models can be used. In this study exsisting vehicle model in CarMaker® is used.

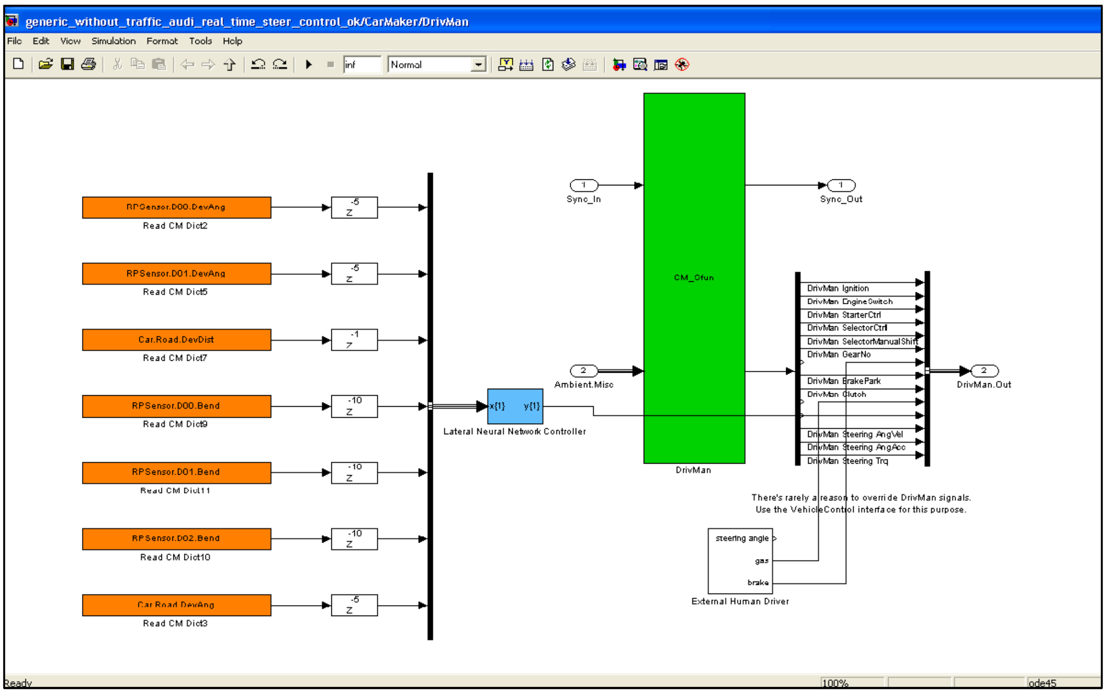

Figure 2.2 : CarMaker® software builtin driver block (green) and user added driver model.

MathWorks© MATLAB® and Simulink® software is used as another platform for developing models / algorithms, pre and postprocessing of results, and for hardware interfacing in this study.

2.2 Hardware

Logitech© Driving Force Pro® gaming console is used to enable human driver for control of vehicle steering angle, gas and brake pedal positions (Figure 2.3). Steering wheel allows 900º of rotation, which corresponds to 2.5 rotations lock-to-lock. This wide range of allowable steering wheel rotation enables realistic simulation of real driving conditions, which is very important in this study since imitation of human driver actions is the goal of this study.

The console is powered with 24 volts, 0.75 Amperes Logitech AC adapter. The external power activates servo DC motor integrated into steering wheel, so that resistance to steering the wheel is simulated as centering spring force. The strength of resistance force, maximum degree of rotation and other settings can be achieved through user interface of the gaming console (Figure 2.4).

Figure 2.3 : Logitech© Driving Force Pro® gaming console.

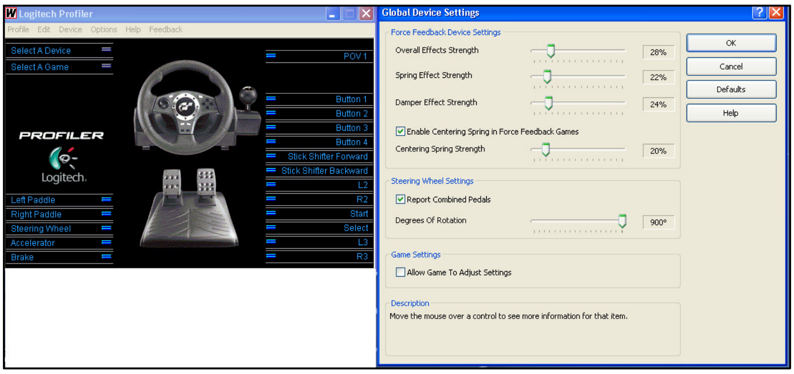

Figure 2.4 : Logitech© Driving Force Pro® user interface.

Interface of Logitech© Driving Force Pro® gaming console to CarMaker® is achieved via developed Simulink® model (Figure 2.5). This interface model maps the signals from Logitech© gaming console into gas pedal, brake pedal and steering angle positions. The outputs of the interface model are overwritten onto IPG driver outputs.

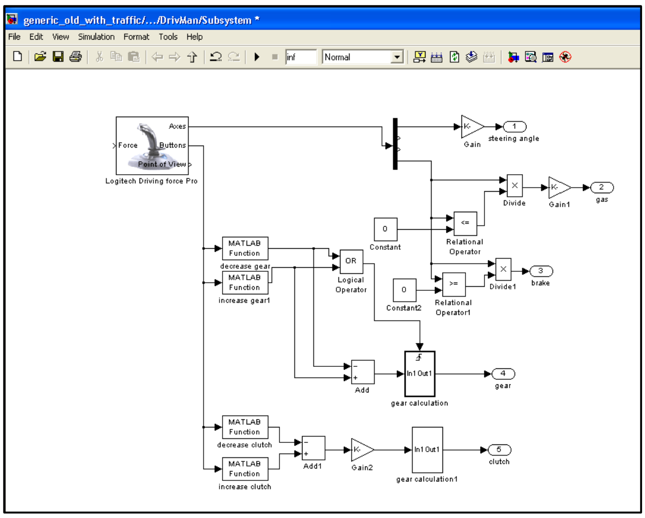

Figure 2.5 : Logitech© Driving Force Pro® interface with CarMaker®.

3. SYSTEM IDENTIFICATION

System identification can be defined as expression of a system model without indulging in physics rules and mathematical models by processing of data collected from the system. Therefore, complex nonlinear models can be defined easily for specific use by appropriate system identification methods (Delice, 2004).

3.1 Data Acquisition

Data acquisition is first and the most important step of system identification. The data collected from system should be rich, which means the data should represent system information accurately. Rich data should be persistent exciting and all frequency modes exciting.

3.1.1 Persistent and all frequency modes excitation

All frequency modes excitation means that the signals contains components in all frequencies for the range [-∞, ∞]. White noise is an example of all frequency modes exciting signal. From human driving modeling point of view, it is satisfactory if the input signals have components up to more than human operator bandwidth at least. This will ensure to obtain a robust model which will have a good performance for validation data as well as for training data. Power spectral analysis can be used to identify magnitude of frequency components of signals used in modeling.

Persistent excitation is a measure of information contained in input signal. The order up to which input signal is persistently exciting limits the selection of model order. Persistent excitation can be defined as follows;

If regression vector is $\varphi(t) = [u^T(t-1) \dots u^T(t-n)]^T$ then,

$$R_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi(t)^T \qquad \textbf{(3.1)}$$

Where N is the number of data samples.

If the above matrix is positive definite, then the input signal u is persistently exciting up to order n.

For estimation of parameters of an n^{th} order dynamical system the input signals should be persistently exciting up to $2 * n^{th}$ order (Söderström and Stoica, 1989).

3.2 Pre Processing of Data

Two kinds of preprocessing can be considered, which are filtering and scaling. If there is a suspection of high frequency noise, that should be eliminated by low-pass filtering techniques to avoid aliasing. Data acquisition should be performed with a sampling frequency at least bigger than 2 times of band limit to avoid aliasing according to Nyquist-Shannon sampling theorem.

3.2.1 Scaling

Intelligent modeling architectures such as Artificial Neural Networks (ANN) and Fuzzy logic are sensitive to absolute magnitude of the input-output data. It is possible that these architectures may adapt to the data which is bigger in magnitude. there are 3 scaling algorithms in the literature, which are Normalization, Linear scaling and Z-score (Arslan, 1999; Tsoukalas ve Uhrig 1997; Delice, 2004).

Z-score scaling is mentioned in this study since there is no risk of information loss in this type of scaling. The aim of z-score is to obtain data independent of unit with zero mean and unit standart deviation. Z-score is formulated in Equation (3.2).

$$z = \frac{x - m_u}{\sigma} \tag{3.2}$$

Where m_u is mean of data and σ is standard deviation of data.

Hovewer, There are some drawbacks of scaling;

- Mean and standard deviation are extracted by use of training data, but if there is an intention to use the model in practical application or for simulation purpose, then the mean and standard deviation may change with newly available data in practical application
- Especially, intelligent methods like fuzzy modeling includes rules to imitate human decision making. It is possible that scaling may cause loss of meaning

in data because of magnitude change in a way which may not be appropriate for human judgement.

For these reasons scaling is not applied in this study when satisfactory performance is obtained without scaling.

3.3 Model Structure Determination

Model structure definition is a critical step since it directly affects performance of the model developed. There are two aspects of structure determination in this study;

- Identification of input signals which correctly represents the output
- Model order selection including time delays

3.3.1 Identification of input signals

Input signals generally selected by expert which has knowledge about process. Statistical correlelation methods can be used as well as trial and error procedure. A related input can be given into model and performance increase for validation data can be analyzed since an improvement in validation results means that the related input has a contribution to model. Hovewer, this method should not be exploited because specific model parameters such as number of free parameters and number of layers for neural network or number of rules, type of membership functions for fuzzy model structures are based on selection of appropriate inputs. Therefore, actually there is two way interaction between appropriate input selection model parameters determination. Therefore, statistical methods are used in combination with validation performance judgement in this study.

3.3.2 Model order selection

Identification of model order is another critical step for system identification. Model order selection means deciding how many delayed samples of input-output will be used. There are various methods in the literature to identify model order.

3.3.2.1 Correlation analysis

In statistics, correlation refers to any of broad class of statistical relationships involving dependence. Correlation may be understood as depature of two random variables from independence. Correlation does not mean necessarily causality between the random variables analyzed, but it may be a clue of causality because if causality exists between two variables, correlation will be high while the opposite is not always true. Actually, correlation is a measure of two or more events are synchronous.

There are several correlation coefficient types used in literature, some of them being Spearman's rank correlation coefficienttau proposed in Spearman (1955), Kendall’s rank correlation proposed in Kendall (1955) and Pearson’s product-moment correlation coefficient proposed in Soper et al. (1917). Pearson’s correlation coefficient is capable of detecting linear dependence and widely used. Hovewer, Pearson’s correlation coefficient is also effective for detection of nonlinear relationships as exprerimented in this study. For an example, if x is taken as a random vector generated in MATLAB®, and y is defined as follows;

$$y = \sin(3x^2) - e^x \tag{3.3}$$

As seen, there is a nonlinear relationship between x and y. comparision of correlation results can be seen in Table 3.1. It can be seen that Pearson’s correlation coefficient can detect the strong nonlinear relationship better than other correlation coefficient types. For this reason Pearson’s correlation coefficient is used in this study as a measure of selecting input signals for models developped.

Table 3.1 : Comparison of different correlation coefficients for relation between x and y in Equation (3.3).

Correlation coefficient type	Coefficient value
Pearson’s product-moment correlation coefficient	-0.64
Kendall’s rank correlation	-0.24
Spearman's rank correlation coefficienttau	-0.38

Pearson correlation coefficient can be defined as follows;

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.4}$$

Where n is number of samples, $\bar{x}$ and $\bar{y}$ are mean of data.

3.3.2.2 Gram- Schmidt ortogonalization

Gram-Schmidt orthogonalization algorithm can be used for removal of effect of a signal from other signal. The signal to be orthogonalized is projected onto other signal and this projected vector is substructed from the first signal to achieve so called removal of effect of second signal. Gram-Schmidt algorithm can be defined as follows (Bau and trefethen, 1997);

$$proj_u(v) = \frac{\langle v, u\rangle}{\langle u, u\rangle} u \quad \textbf{(3.5)}$$

Where $\langle v, u\rangle$ means inner product of the vectors v and u. Therefore, orthogonalized v is given as;

$$v_{ortg} = v - proj_u(v) \quad \textbf{(3.6)}$$

The methodology of removing effect of a signal from another signal is used for identification of inputs of nonlinear automotive oxygen sensor model in connection with coherence analysis (Krishnaswami et al, 1995). It is arqued in this study that coherence analysis can represent nonlinear relation even if coherence analysis is intended for use with linear systems.

In this thesis, correlation analysis in Chapter 3.3.2.1 is used instead of coherence analysis because of the following reasons;

- correlation analysis is more direct than coherence by giving a total index in the range of [-1,1]
- coherence is expected to be more sensitive to noise since it is based on frequency band analysis, but correlation analysis is based on standard deviation which is more robust to low amplitude noise.
- Coherence detects relation between input and output of SISO (Single Input-Single Output) dynamical system. Coherence will be low if an output depends on other inputs.

3.4 Modeling and Validation

In this section the model structures used in driver modeling and validation measures to assess the performance of obtained model are explained. Specific modeling techniques will be detailed in Chapter 5.

3.4.1 Nonlinear auto regressive with eXogenous inputs (NARX) and nonlinear finite impulse response (NFIR) structure

NARX structure can be seen from Figure 3.1. Regressor vector for NARX can be composed as following (Abonyi and Feil**,** 2007);

$$\emptyset_{NARX}(k) = [y(k-1), \ldots, y(k-n_A), u(k-n_k), \ldots, u(k-n_k-n_B)] \quad \textbf{(3.7)}$$

Similarly,Regressor vector for NFIR structure in Figure 3.2, can be composed as;

$$\emptyset_{NFIR}(k) = [u(k-n_k), \ldots, u(k-n_k-n_B)] \quad \textbf{(3.8)}$$

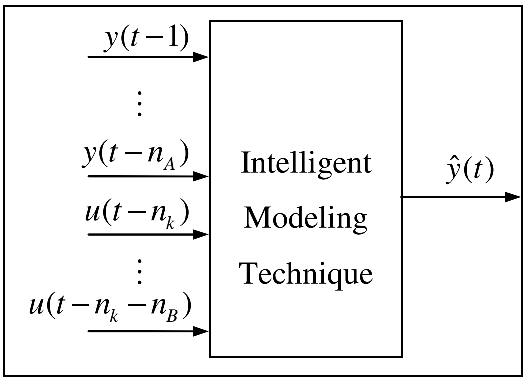

Figure 3.1 : Nonlinear ARX structure.

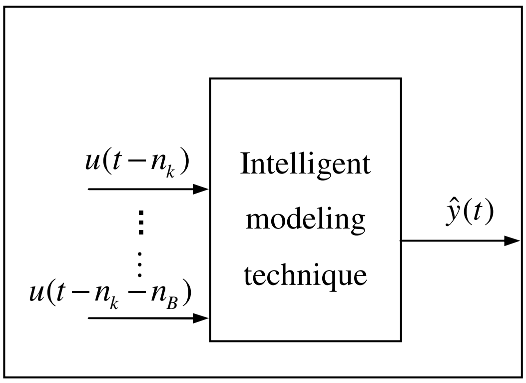

Figure 3.2 : Nonlinear FIR structure.

Here, n_A is order of output, n_B is order of input, and n_k is delay of input.

3.4.2 Percentile variance (PV)

In this study percentile variance is used for offline performance assessing criteria. This means that training and validation results are checked for fitting to real process outputs to have an initial idea about performance of model before practical application in simulator, which saves time.

Percentile variance can be formulated as follows (Ljung, 1999);

$$PV = [1 - \frac{\|y - y_m\|_2}{\|y - \bar{y}\|_2}] * 100 \quad \textbf{(3.9)}$$

Where y_m is model output vector, y is real process output and $\bar{y}$ is the mean of real process output.Euclideannorm is used in the formula.

4. INTELLIGENT MODELING METHODS

Intelligent modeling methods are useful for modeling of complex nonlinear systems like human. Artificial intelligence (AI) techniques were attracting interest in the past decades since these techniques are scoped to mimic human mind.

With the advance in technology of hardware, software and sensors, it has been possible that soft computing and AI methods evaluated from traditional predicate logic to intelligent systems techniques may be used in solution of real world complex industrial modeling problems.

There are various kinds of intelligent modeling techniques in the literature such as fuzzy logic, neural networks, probabilistic reasoning, genetic algorithms, and some combination of these techniques which results in hybrid intelligent methods. Some of these intelligent modeling techniques are used for the human driver modeling problem In this thesis.

4.1 Artificial Neural Networks (ANN)

Artificial neural networks consist of simple processing units connected in parallel to each other, which is inspired by human neurological system. Structure of a simple artificial neuron can be seen from Figure 4.1.

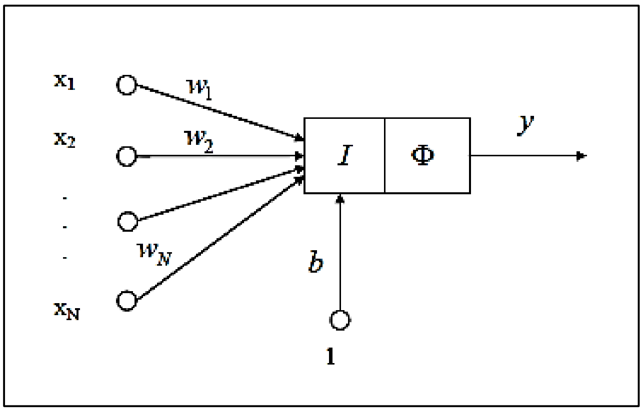

Figure 4.1 : Artificial neuron structure, adapted from (Hagan and Menjah, 1994).

$\boldsymbol{x} = [x_1, x_2, \dots, x_N]^T$ represents inputs into neuron. The weight vector $w = [w_1, w_2, \dots, w_N]$ is applied into individual inputs. b is called bias and applied as weighting function of constant input.Therefore, the mapping between input (I) and output y can be defined as;

$$I = w_1x_1 + w_2x_2 + \cdots + w_Nx_N + b = \sum_{i=1}^{N} \boldsymbol{wx} + b \quad \textbf{(4.1)}$$

$$y = \Phi(I) \quad \textbf{(4.2)}$$

Where Φ is called activation function for which first derivative shoud exist. Several activation functions such as tangent sigmoid, gausian, logarithmic sigmoid and linear are used in literature.

Supervised multi layer feedforward ANN structure with Levenberg-Marquardt training algorithm due to its fast convergence is used in this thesis. The general structure of a 3 input-3 output feedforward neural network with 1 hidden layer can be seen in Figure 4.2.

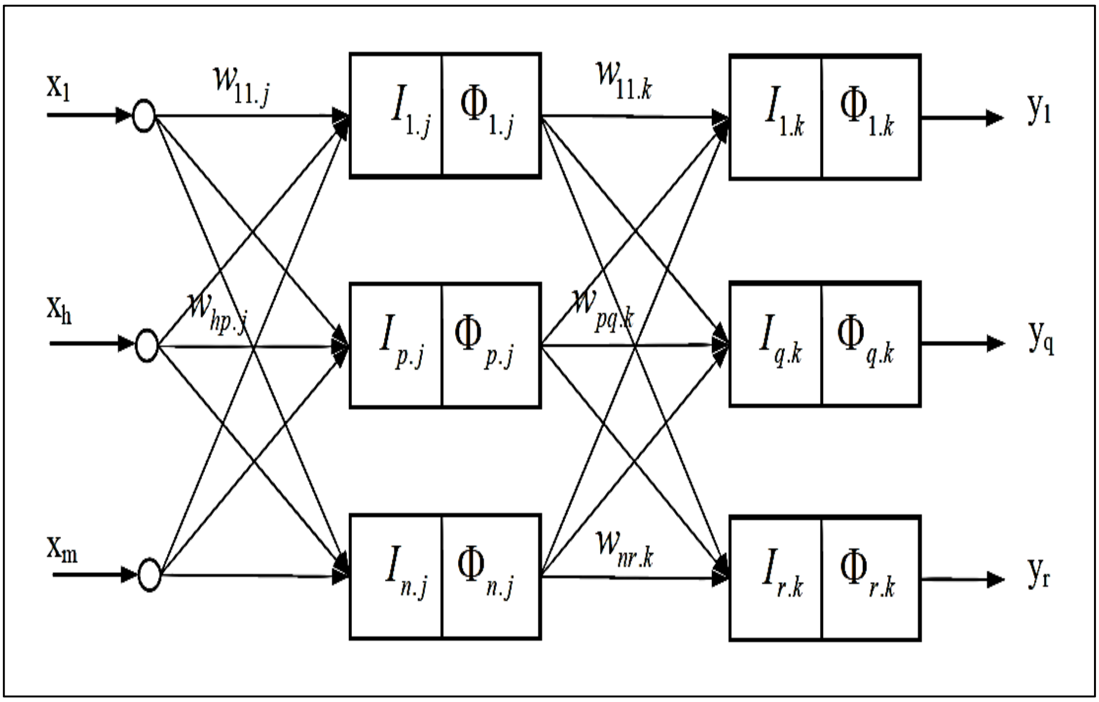

Figure 4.2 : 3 layer feedforward ANN structure, adapted from (Hagan and Menjah,1994).

Levenberg-Marquardt (LM) algorithm can be defined as follows (Hagan and Menhaj, 1994);

$$\Delta\boldsymbol{\theta} = -(\boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta}) + \lambda I)^{-1}\boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{e}(\boldsymbol{\theta}) \quad \textbf{(4.3)}$$

$$
\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial e_1(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_1(\boldsymbol{\theta})}{\partial \theta_2} & \cdots & \frac{\partial e_1(\boldsymbol{\theta})}{\partial \theta_n} \\ \frac{\partial e_2(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_2(\boldsymbol{\theta})}{\partial \theta_2} & \cdots & \frac{\partial e_2(\boldsymbol{\theta})}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial e_N(\boldsymbol{\theta})}{\partial \theta_2} & \cdots & \frac{\partial e_N(\boldsymbol{\theta})}{\partial \theta_n} \end{bmatrix} \qquad \textbf{(4.4)}
$$

Where $\boldsymbol{e}$ is error vector for all samples of an output node. J is Jacobian of error vector. For multi-output ANN, error derivatives of other nodes should also be inserted between the rows in Equation (4.4). $\boldsymbol{\theta}$is the parameter vector which contains all weights and bias values of the ANN. The derivatives in Jacobian matrix can be calculated by use of back propagation algorithm. The details of back propagation algorithm can be found in Delice (2004).

LM algorithm is used in following way;

1. All the inputs are fed into the network to calculate outputs and errors.
2. Jacobian matrix in Equation (4.4) is calculated.
3. $\Delta\boldsymbol{\theta}$is calculated from Equation (4.3)
4. Cost function is recalculated as Mean Squred Error (MSE) by using new parameter vector $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$. If the value of cost fuınction is smaller, then λ factor should be made smaller and iteration should be started from step 1 again. If the value of cost function is bigger, than step 3 should be repeated with increase of λ.
5. The iteration above is repeated until maximum iteration number is reached or until desired cost function values is achieved.

4.2 First Order Takagi-Sugeno Fuzzy Model

The TS model is a combination of a logical and a mathematical model. This modelis also formed by logical rules; it consists of a fuzzy antecedent and a mathematicalfunction as consequent part. The antecedents of fuzzy rules partition the inputspace into a number of fuzzy regions, while the consequent functions describe thesystem behavior within a given region;

$$R_j\text{:}\ \boldsymbol{IF} z_1\ is\ A_{1,j}\ \boldsymbol{and} \dots \boldsymbol{and}\ z_n\ is\ A_{n,j}\ \boldsymbol{then}\ y = f_j(q_1, \dots, q_m) \qquad \textbf{(4.5)}$$

Where$\boldsymbol{z} = [z_1, \dots, z_n]^T$ is the n-dimensional vector of the antecedent variables,$\boldsymbol{z} \epsilon \boldsymbol{x}, \boldsymbol{q} = [q, \dots, q_m]^T$ is the m-dimensional vector of the consequent variables$\boldsymbol{q} \epsilon \boldsymbol{x}$, where **x** denotes the set of all inputs of the$y = f(\boldsymbol{x})$ model. $A_{i,j}(z_i)$ denotes the antecedent fuzzy set for the i_{th} input. The antecedents of fuzzy rulespartition the input space into a number of fuzzy regions, while the $f_j(\boldsymbol{q})$ consequentfunctions describe the system behavior within a given region.

Usually, the f_j consequent function is a polynomial in the input variables, butit can be any arbitrary chosen function that can appropriately describe the outputof the system within the region specified by the antecedent of the rule.When $f_j(\boldsymbol{q})$is a first-order polynomial,

$$f_j(\boldsymbol{q}) = \theta_j^0 + \theta_j^1 q_1 + \dots + \theta_j^m q_m = \sum_{l=0}^{m} \theta_j^l q_l, \quad q_0 = 1 \qquad \textbf{(4.6)}$$

The resulting fuzzy inference system is called first-order Takagi–Sugeno model.

Using fuzzy inference based upon product-sum-gravity at a given input (Yager and Filev, 1994),the final output of the fuzzy model, y, is inferred by taking theweighted averageof the consequent functions:

$$y = \frac{\sum_{j=1}^{N_r} \beta_j f_j(q)}{\sum_{j=1}^{N_r} \beta_j} \qquad \textbf{(4.7)}$$

Where the weight, $0 \leq \beta_j \leq 1$, represents the the overall truth value (degree offulfillment) of the i_{th} rule calculated based on the degrees of membership. $\beta_j$can be explained as follows;

$$\beta_j(\boldsymbol{z}) = \prod_{i=1}^{n} A_{i,j}(z_i) \qquad \textbf{(4.8)}$$

Where n is the number of input dimension and $A_{i,j}$ is the fuzzy set for z_i input (Abonyi and Feil, 2007).

4.3 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS architecture was proposed by Jang (1993) .It makes use of neural network algorithms to tune parameters of TS fuzzy system explained in Chapter (4.2), which results in a hybrid learning algorithm. Therefore, with a so called analogy, if ANFIS is considered as human mind, neural network properties of ANFIS represent learning of human mind while the fuzzy part represents judgement.

ANFIS structure for a 2 input-1 output first order TS fuzzy system with 2 rules can be shown in the following way (Jang et al,1997);

Rule-base for the first order TS fuzzy system in Figure 4.3 is given as,

Rule 1: if x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$ **(4.9)**

Rule 2: if x is A_3 and y is B_2, then $f_2 = p_2x + q_2y + r_2$ **(4.10)**

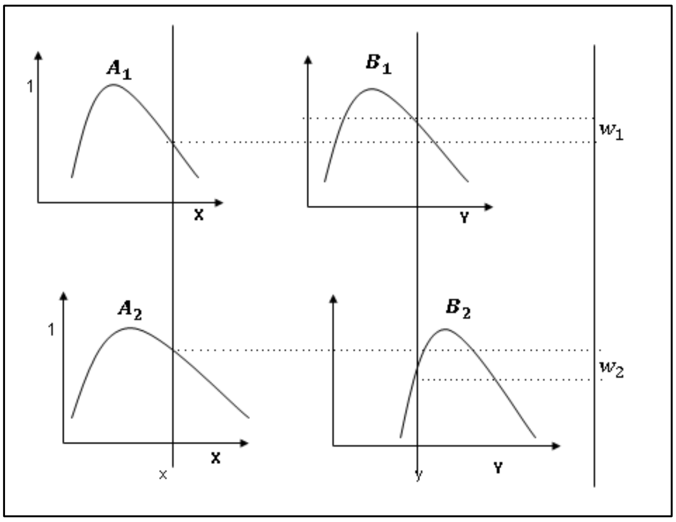

Figure 4.3 : Two input first order Sugeno fuzzy model with two rule.

The equivalent ANFIS architecture which corresponds to this rule base can be seen in Figure 4.4. Nodes of the same layer has same kind of functions. The processes in each layer are explained as following;

Layer 1;

Every node i in this layer is an adaptive node with a node function

$$O_{1,i} = \mu_{A_i}(x) \; for \; i = 1,2, or \quad \textbf{(4.11)}$$

$$O_{1,i} = \mu_{B_{i-2}}(y)\ for\ i = 3,4 \qquad \textbf{(4.12)}$$

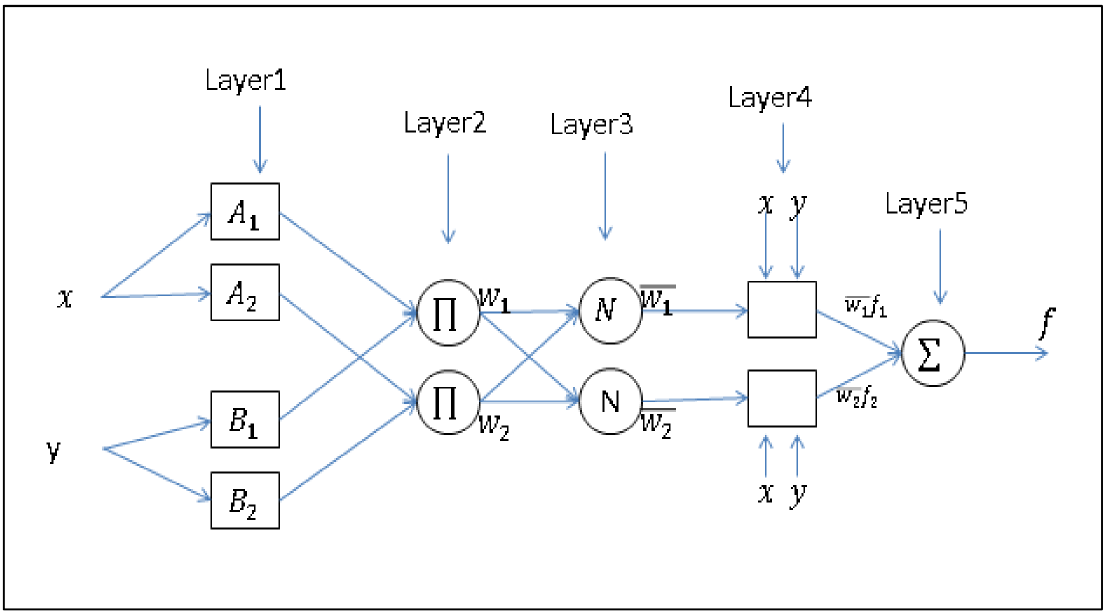

Figure 4.4 : ANFIS structure for two input first order Sugeno fuzzy model with two rules.

Where $O_{l,i}$ is the output of i_{th} node in layer l. x or y is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as "small" or "large") associated with this node. In other words, $O_{1,i}$ is the membership grade of a fuzzy set A (= A_1,A_2, B_1, or B_2) and it specifies the degree to which the given input x(or y) satisfies the quantifier A.

Layer 2;

Output of the nodes in this layer is the product of all the input signals coming:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \qquad i = 1,2 \qquad \textbf{(4.13)}$$

Each node output represents the firing strength of a rule. Multiplication represents AND operator that appears in TS fuzzy model.

Layer 3;

This layer is used for normalization of firing strengths that are calculated in Layer 2. It is formulated as;

$$O_{3,i} = \overline{w}_i = \frac{w_i}{\sum_{j=1}^{n} w_j}, \qquad i = 1,2 \qquad \textbf{(4.14)}$$

Where n is number of the nodes in this layer.

Layer 4;

Every node i in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \quad \textbf{(4.15)}$$

Where $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are refered to as consequent paramaters.

Layer 5;

The single node in this layer is a fixed node labeled $\sum$, which computes overall output as the summation of all incoming signals:

$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i \overline{w}_i} \quad \textbf{(4.16)}$$

Equation (4.16) can be expanded as below for the two input 2 rule TS fuzzy system;

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \quad \textbf{(4.16)}$$

$$f = \overline{w}_1(p_1 x + q_1 y + r_1) + \overline{w}_2(p_2 x + q_2 y + r_2) \quad \textbf{(4.17)}$$

$$f = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \quad \textbf{(4.18)}$$

Equation (4.17) is linear in parameter $p_1, q_1, r_1, p_2, q_2,$ and r_2. Hybrid learning algorithm of ANFIS has two phases. In the forward pass, node outputs go forward until layer 4 and the consequent parameters are identified by the least squares method. In the backward pass, the error signal is propagated back and the premise parameters of fuzzy sets are updated by gradient descent method. Table 4.1 summarizes the procedure of the hybrid learning algorithm.

4.4 Fuzzy Clustering

An effective approach to the identification of complex nonlinear systems is to partition the available data into subsets and approximate each subset by a simple model. Fuzzy clustering can be used as a tool to obtain a partitioning of data where the transitions between the subsets are gradual rather than abrupt.The objective of fuzzy clustering is the classification of objects according to similarities among them.

Clustering techniques are among the unsupervised learning methods (Babuska, 1998).

Table 4.1 : Two passes in the hybrid learning procedure for ANFIS.

Forward pass			Backward pass		
Premise parameters	Consequent parameters	Signals	Premise parameters	Consequent parameters	Signals
Fixed	Least squaresestimator	Node outputs	Gradient descent	Fixed	Error Signals

There are various clustering algorithms defined in literature (Babuska,1998; Abonyi and Feil, 2007):

- Fuzzy c-means (FCM) clustering
- Gustafson-Kessel (GK) algorithm
- Fuzzy Maximum Likelihood Estimates (FMLE) clustering
- Gath-Geva (GG) clustering algorithm

FCM clustering is the basis for most of the clustering algorithms and simple to program on computer. For this reason FCM clustrering technique is used in this study although it may not be as effective as other clustering methods for detecting linear subspaces.

4.4.1 Fuzzy C-means clustering

FCM algorithm is based on minimization of the c-means functional formulated as (Bezdek,1981):

$$J(Z;U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m\,\|z_k-v_i\|^2 \tag{4.19}$$

Where

$$U=[\mu_{ik}]\in M_{fc} \tag{4.20}$$

$$V=[v_1,v_1,\dots,v_c],\ v_i\epsilon\mathbb{R} \tag{4.21}$$

$$D_{ik}^2=\|z_k-v_i\|^2=(z_k-v_i)^T(z_k-v_i) \tag{4.22}$$

U is a fuzzy partition matrix of **Z** and V is is a matrix of cluster prototypes (centers), which have to be determined. D_{ik}^2 is a squared inner-product distance norm, and m is a weighting exponent which determines the fuzziness of the resulting clusters. **Z** is the data matrix to be partitioned.

The FCM algorithm that minimizes the fuzzy c-means functional is based on constrained minimization function by Lagrange multipliers in Equation (4.23) (Babuska,1998):

$$J(Z;U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D_{ik}^2 + \sum_{k=1}^{N}\lambda_k\left[\sum_{i=1}^{c}\mu_{ik]} - 1\right] \quad \textbf{(4.23)}$$

The FCM algorithm that minimizes this cost function is given as following:

Given the data set **Z,** number of clusters, the weighting exponent m>1, the termination tolerance $\epsilon > 0$ are chosen. The partition matrix $\boldsymbol{U}$ is initialized randomly and the below steps are repeated for $l = 1,2, \ldots$

Step 1;

Compute the cluster prototypes (means):

$$v_i = \frac{\sum_{k=1}^{N}(\mu_{ik}{}^{(l-1)})^m z_k}{\sum_{k=1}^{N}(\mu_{ik}{}^{(l-1)})^m} \quad \textbf{(4.24)}$$

Step 2;

Compute the distances:

$$D_{ik}^2 = \|z_k - v_i\|^2 = (z_k - v_i)^T(z_k - v_i), \qquad 1 \le i \le c, \quad 1 \le k \le N \quad \textbf{(4.25)}$$

Step 3;

Update the partition matrix:

$$u_i = \frac{1}{\sum_{j=1}^{c}(D_{ik}^2/D_{jk}^2)^{2/(m-1)}} \quad \textbf{(4.26)}$$

4.4.2 Weighted least squares method to estimate consequent parameters of TS fuzzy models based on FCM clustering

The ordinary weighted least-squares method can be applied to estimate the consequent parameters of the local linear models$\theta_i$separately by minimizing the criteria in Equation (4.27) (Abonyiand Feil, 2007):

$$\min_{\theta_i}(\boldsymbol{y}-\boldsymbol{X_e}\boldsymbol{\theta}_i)^T\boldsymbol{\Phi_i}(\boldsymbol{y}-\boldsymbol{X_e}\boldsymbol{\theta}_i) \qquad \textbf{(4.27)}$$

Where $\boldsymbol{X_e}=[\boldsymbol{X}\ \mathbf{1}]$is the regressor matrix extended by a unitary column and $\boldsymbol{\Phi_i}$ **is** amatrix having the membership degrees on its main diagonal for the subsystem considered:

$$\boldsymbol{\Phi_i}=\begin{bmatrix}\mu_{i,1} & 0 & \cdots & 0\\ 0 & \mu_{i,2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \mu_{i,N}\end{bmatrix} \qquad \textbf{(4.28)}$$

Then, the consequent parameter vector that minimizes the criteria becomes:

$$\boldsymbol{\theta}_i=(\mathbf{X}_e^T\boldsymbol{\Phi}_i\boldsymbol{X}_e)^{-\mathbf{1}}\mathbf{X}_e^T\boldsymbol{\Phi}_i\boldsymbol{y} \qquad \textbf{(4.29)}$$

4.5 Genetic Algorithms (GA)

Genetic algorithms (Goldberg, 1989; Holland, 1975) are derivative-free stochastic optimization methods based loosely on the concepts of natural selection and evolutionary processes.

GAs are popular because of the following reasons;

- GAs are parallel-search procedures that can be implemented on parallel-processing machines for speeding up operation.
- GAs are applicable to both continuous and discrete(combinatorial) optimization problems.
- GAs are stochastic and less likely to get trapped in local minima, which inevitably are present in any practical optimization application.
- GAs' flexibility facilitates both structure and parameter identification in complex models such as neural networks and fuzzy inference systems.

GAs encode each point in a parameter space into a binary bit string called chromosome, and each point is associated with a “fitness” value that is usually equal to the objective function evaluated at the point. Instead of single point, GAs usually keep a set of points as a population, which is then evolved repeatedly toward a better overall fitness value. In each generation, the GA constructs a new population using genetic operators such as crossover and mutation; members with higher fitness values are more likely to survive and participate in mating (crossover) operations. After a number of generations , the population contains members with better fitness values.

Major components of GA algorithm can be explained as following:

Encoding schemes; These transform points in a parameter space into bit strig representations. For instance, a point (11,6,9) in a tree dimensional parameter space can be represented as a concatenated binary string:

$$\underbrace{1011}_{11}\underbrace{0110}_{6}\underbrace{1001}_{9}$$

In which each coordinate value is encoded as a gene composed of four binary bits using binary coding.

Fitness evaluation; The first step after creating a generation is to calculate the fitness value of each member in the population. The fitness value f_i of the i_{th} member is usually the objective function evaluated at this member. Another approach can be to use rankings of members as fitness value when objective functions are not accurate enough.

Selection; The selection operation determines which parents participate in producing offspring for the next generation. Usually, the members are selected for mating with a selection probability proportional to their fitness values. The most common way to implement this is to set the selection probability equal to $\frac{f_i}{\sum_{k=1}^{n} f_k}$ where n is the population size.

Crossover; Crossover is used to generate new chromosomes. It is important to retain good features from the previous generation and to satisfy the potential of gene pool. Crossover is usually applied to selected pairs of parents with a probability equal to a given crossover rate. Crossover point on genetic code is selected at random and two parent chromosomes are interchanged at this point.

Mutation; if the population does not contain all the encoded information needed to solve a particular problem, no amount of gene mixing can produce a satisfactory solution. For this reason, a mutation operator for spontaneously generating new chromosomes is applied. The mutation operation can prevent the GA from stucking at local optima. The mutation operation flips the bits of population members with a given mutation rate. The mutation rate should be very small to prevent loss of chromosomes obtained from crossover (Jang and Sun, 1997).

5. DRIVER MODELING FOR LATERAL AND LONGITUDINAL CONTROL OF VEHICLE

Human driver modeling is a complex work because while a real human driver is capable of achieving driving task in many different driving scenerios, a driver model can only be successful in specific driving scenerios. For example, a driver model can be very successful in tracking a road profile with no inclination and no traffic, but may fail for parking task. To solve this problem, there should be sub-modules prepared for specific tasks and each module should perform its task in a synchronously or asynchronously coordinated way with other modules.

In many studies in the literature, human driver is modeled for only one sceneria such as lane-keeping, tracking a road profile with constant radius of curvature in constant speed. In this thesis, human driver is modeled as 2 separate controllers that work in parallel simultaneously, which are;

- Lateral controller
- Longitudinal controller

The general overview of driver modeling structure can be seen in Figure 5.1. Lateral and longitudinal aspects of the human driver are modeled separately for simplification purpose. If these two modeling approaches are unified in a single MIMO (Multi Input Multi Output) structure, the single model will be complicated and possibly performance will be worse because it can be seen in Chapter 5.3.1 that the two control aspects are sensitive to different input variables. In a unified MIMO structure for human driver modeling, all of the inputs will be fed into model although some of them only excite one of the lateral or longitudinal control acitons.

Another reason of studying two separate models for longitudinal and lateral control actions is that the characteristics of the human driver will be identified more easily by discovering the input variables special to lateral or longitudinal control behavior of the human driver seperately. This is done by combining the validation procedure defined in Chapter 5.4 and input-output selection criteria defined in Chapter 5.3.1.

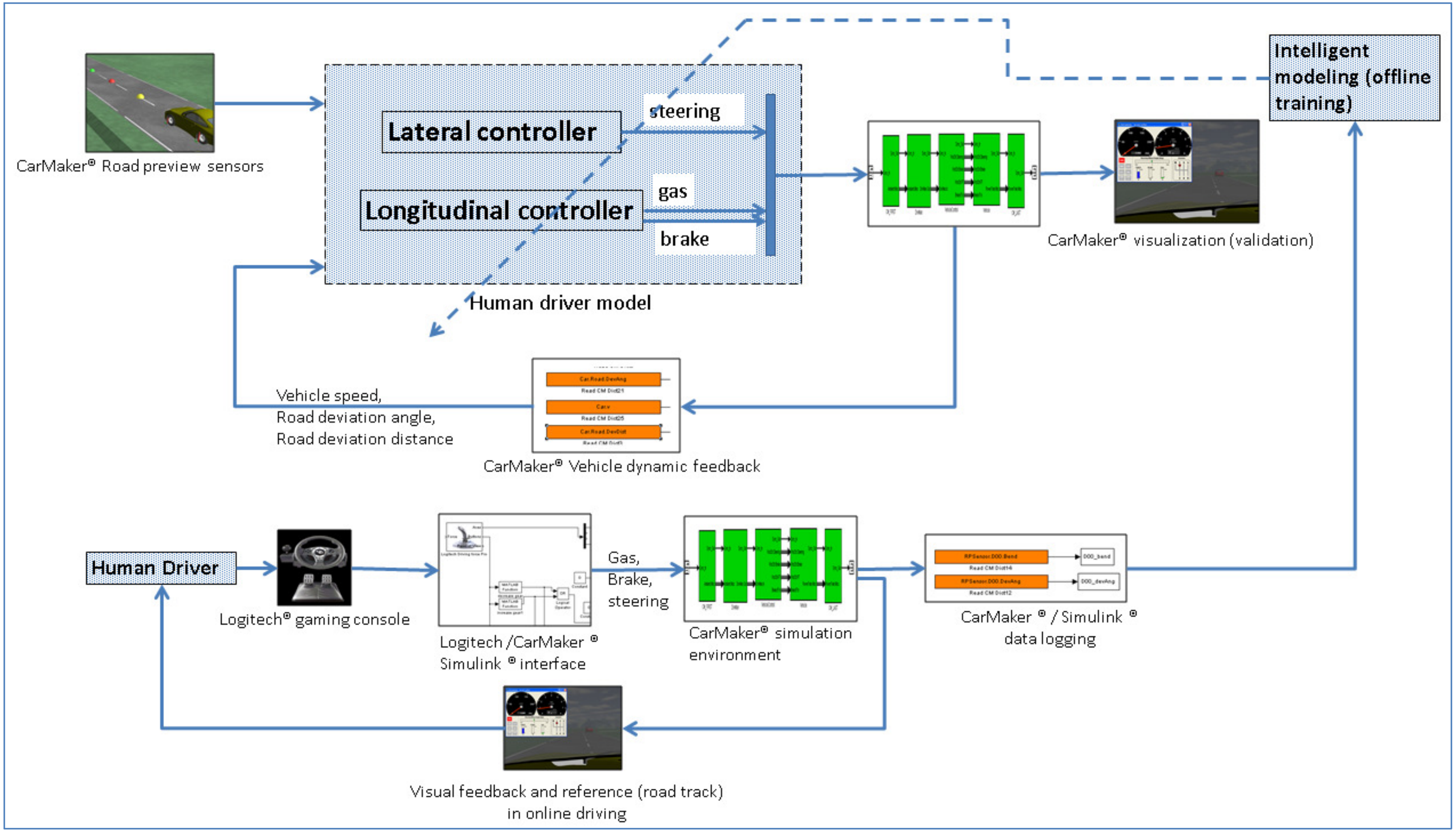

Figure 5.1 : General overview of driver modeling structure.

It may be argued that separating the longitudinal and lateral actions of human driver in two different model structures will cause decoupling of these two behaviors, but it is shown in validation results (Section 5.4) that these two separate models can accomplish the driving task simultaneously like real human driver.

5.1 Road Profile

IPG © CarMaker® software allows generation of custom road profiles as well as providing oppurtunity of using built-in road profiles. The following road segments can be connected to each other to form desired road profile;

- Straight with defined length,
- Right-left turns with defined angle and radius,
- Clothoid right-left turns with defined angle, different start and end radius

2 different road profiles for training of the driver model and for validation of the driver model are considered in this study. Bird eye view of the road profile that is used for training of the driver model can be seen in Figure 5.2. This road profile is formed by using the user defined road segments as mentioned previously. While forming the road profile, it is considered that maximum possible variability is included in road shape without considering the rules of standard highways. This will ensure that the human driver model that learns from the data collected in this road profile will learn maximum possible different kind of driving scenarios.

For validation purposes, one of the CarMaker® built-in road profiles is used (Figure 5.3). CarMaker built-in road profiles are formed by using real world road coordinates which are consistent with standard highway rules.

5.2 Collection of Data for Modeling

In this study, intelligent modeling algorithms for driver model are trained by using data from driving simulation with real human driver in the loop. The collected data from human driver should be rich, which means the driver model obtained from real human driver data will exhibit human driver dynamic characteristics independently from the collected data.

Rich data should be persistent and all frequency mode exciting as defined in Chapter 3.1.1. The persistent excitency analysis of input signals with Equation (3.1) shows that all the inputs selected for lateral and longitudinal control driver models are persistent excitant up to order of 50 when data is collected with 10 Hertz sampling frequency. If the requirement in Chapter 3.1.1 is checked, it is seen that the input signals should be persistent excitant up to order of at least $2 * n^{th}$, where n^{th} is the order of the input signal considered. The persistent excitency order of 50 satisfies this condition for all of the selected input signal orders.

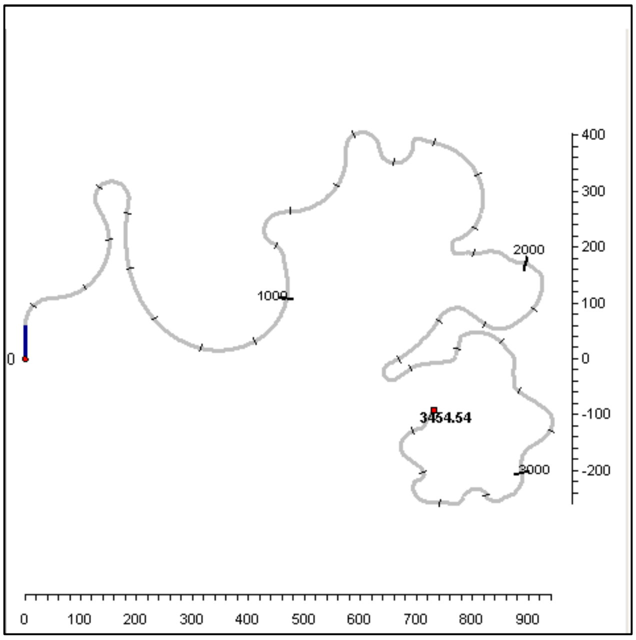

Figure 5.2 : Bird eye view of the rod profile for training.

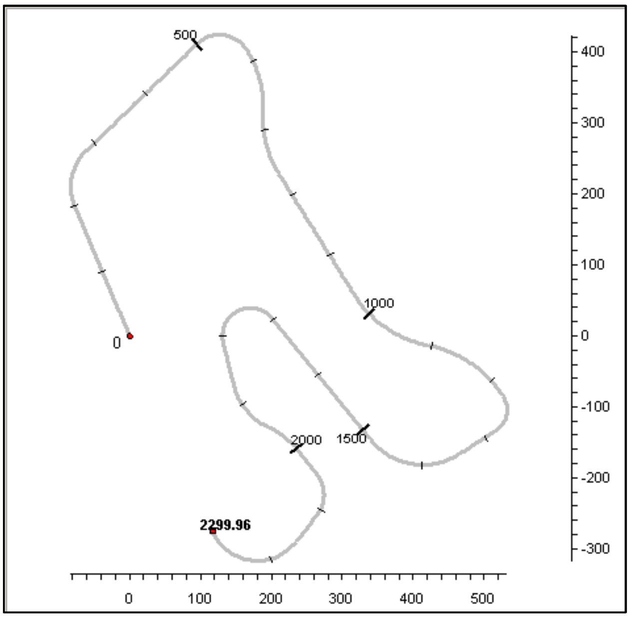

Figure 5.3 : Bird eye view of the road profile for validation.

For ideal rich data, the signal should be exciting all frequency modes, but in reality the frequency band of human driver in this study is up to only 1.5 Hertz as can be seen from Figure 5.4 for steering wheel, gas and brake pedal actions of the human driver. This shows that the even though the maximum variability of road bend is considered in generation of road profile for modeling, it still doesn't excite all frequency band range of the human driver, which means the driver model may not show sufficient performance for the input signals with higher frequencies. However, it is expected that real world standard highways will not include higher variability than the road profile that is used for modeling purposes in this study. Therefore the driver model is expected to achieve driving task for standard highway profiles.

Data is collected with a sampling frequency of 10 Hertz in this study. It is mentioned in Chapter 3.2 that according to Nyquist-Shannon sampling theorem, the sampling frequency should be at least sampling frequency at least bigger than 2 times of band limit to avoid aliasing. The upper limit of frequency band can be seen as 1.5 Hertz in Figure 5.4, so the sampling frequency should be bigger than 3 Hertz. This means 10 Hertz sampling frequency is enough to avoid aliasing and to ensure highest frequency components of signals are included in modeling.

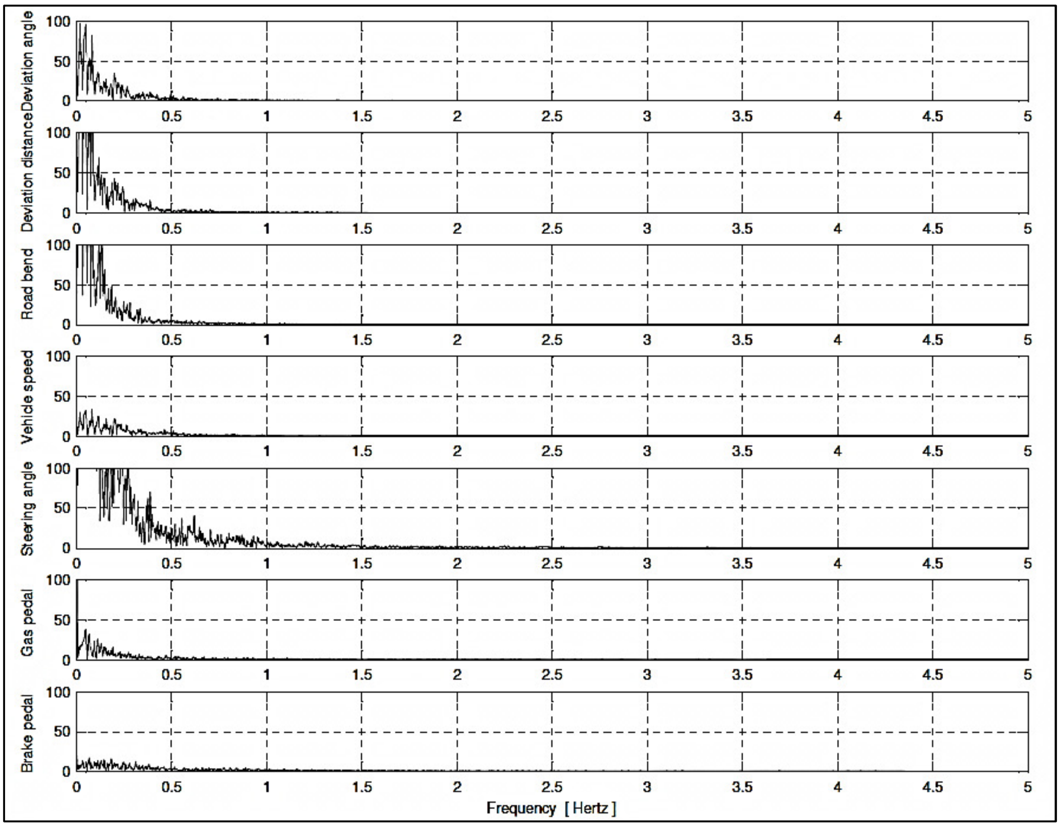

Figure 5.4 : Frequency band of input-output signals that are used in modeling.

In this study, road preview information is gathered from the IPG© CarMaker® sensors that are placed at specific distances from the front of the vehicle. Actually, driver's look-ahead distance changes with velocity of vehicle, that is, when the velocity increases the driver will preview road at longer distances. However, it is hard to identify from the analysis of data at what distances the driver previews road in relation with vehicle velocity since it depends on perception of human and there is no formal way of understanding it. For this reason, the vehicle speed is taken as one of the inputs in modeling. It is expected that the intelligent modeling methods used will infer the relation between vehicle speed and road preview information at specific distances and validation results show that this approach is consistent.

Scaling of input-output data has been performed with z-score scaling approach that is mentioned in withEquation (3.2), but no improvements has been identified in offline training and validation results when compared with the modeling results without any scaling. Therefore, the scaling has not been used for modeling in this study due to the reasons mentioned in Chapter 3.2.1.

5.3 Lateral and Longitudinal Control Models of Human Driver

Human driver model in this study is obtained as combination of two independent MISO (Multi Input Single Output) models, which are longitudinal and lateral controllers (Figure 5.1). These the controllers work in parallel simultaneously to achieve driving task, so that they act like a single MIMO (Multi Input Multi Output) model.

Two different model structures NARX (Nonlinear Auto-Regressive with eXogenous inputs) and NFIR (Nonlinear Finite Impulse Response) are studied for modeling that can be seen in Figure 3.1 and Figure 3.2.The difference between these two structures is that past samples of output values is not fed as input signals in NFIR structure.NFIR structure is chosen as modeling structure since better validation performance has been observed with this structure.

Different types of intelligent modeling algorithms that are reviewed in Chapter 4 are applied to driver modeling with NFIR structure. The results for ANN (Artificial Neural Networks) and ANFIS (Adaptive Neuro-Fuzzy Inference system) are presented in this Chapter.

In addition to ANN and ANFIS, 2 different hybrid algorithms which are combination of some of the intelligent modeling methods in Chapter 4 are developed and studied in this thesis. However, it has been seen that these two newly proposed modeling algorithms has not been appropriate for practical application to complex nonlinear human driver modeling due to computational problems. These two algorithms are discussed further in Chapter 6.

5.3.1 Selection of input and output variables

Identification of appropriate input-output variables that represent dynamic characteristics of the human driver is one of the most important steps for modeling in this study because the driver model is obtained from system identification and intelligent modeling methods instead of pure mathematical relations.

If one of the necessary input variables is neglected in modeling, the performance will be degraded, while addition of extra unnecessary inputs will complicate the modeling work. When number of inputs increase, number of hidden neurons increase for ANN, which brings long training duration and memorization problems. Also, number of rules for ANFIS increases exponentially with increasing number of inputs, which causes computational memory problems in computer application.

Driver model can be obtained with different types of outputs such as steering wheel angle, steering moment, gas pedal position, brake pedal position, and gear. In this study, driver model is obtained for a vehicle with automatic transmission, so the model does not decide gear level. The output of Lateral control model in Figure 5.1 is taken as steering angle since there is no way of measuring steering moment applied by driver with the hardware used. The longitudinal control model output is taken as the difference between gas and brake pedal positions (gas pedal position-brake pedal position) by considering these two quantities are affected by same input variables. This output selection approach for longitudinal control simplifies the model and increases accuracy by combining similar characteristics in a single output. Also, if these two quantities are considered as separate outputs of a combined MIMO or two separate MISO models, then there may be a contradiction that both the gas pedal and brake pedals are activated simultaneously, which is far away from real human driver behavior. In the unified approach of longitudinal control model, the output of positive value is fed into simulation environment as gas pedal position while keeping the brake at zero position and the opposite is done for output of negative value.

The selection of the inputs is based on the techniques defined in Chapter 3.3and validation results. Pearson's product-moment correlation coefficient is used to identify which inputs are correlated more with the output considered. Pearson's product-moment correlation coefficient is also useful to identify the correlation of delayed values of input signals with the output. This analysis for effect of delayed input values on output is critical to decide model order.

Although the correlation analysis is effective in catching the nonlinear relation between input-output variable to some extent, it has some deficiencies. For example, a high negative correlation is expected between steering angle and vehicle speed since the driver will decelerate while taking a sharp turn with a big steering wheel angle. However, steering angle data is collected as negative and positive measurement values for right and left turns respectively, which distorts the existant correlation between them. This problem is solved by taking absolute value of steering angle measurement. Also, correlation does not give any information about causality, which is a problem for selection of inputs. For example, there is a high correlation between engine speed and gas pedal position, but this does not mean the engine speed is a critical input for longitudinal control because actually high gas pedal position is not caused by high engine speed, while high engine speed is caused by high gas pedal position. Another problem with correlation is that high correlation of an unnecessary input may be just because of its dependency with another high correlated necessary input. For example, vehicle speed is a necessary and high correlated input with steering angle, but engine speed is an unnecessary input for steering angle. However, engine speed shows a high correlation with steering angle due to its dependency with vehicle speed. This problem of dependency has been tried to be eliminated with Gramm-Schmidt orthogonalization defined in Equation (3.5) and Equation (3.6), but it has distorted correlation by setting new mathematical relation between inputs. Despite all these problems, correlation analysis is useful for making and initial estimate of input variables and their orders and preferred over coherence analysis due to reasons mentioned in Chapter 3.3.2.2. This initial estimate is then supported with longer validation procedure with the obtained model. The correlation analysis is important for accelerating the input selection procedure before validation step by eliminating many unuseful possibilities.

The final input-output selection for lateral and longitudinal control can be seen in Figure 5.5 and Figure 5.6 respectively. The inputs are supplied to the models by using CarMaker® Simulink® blocks. Simulink® discrete time delay blocks are used to supply past samples of inputs considered.

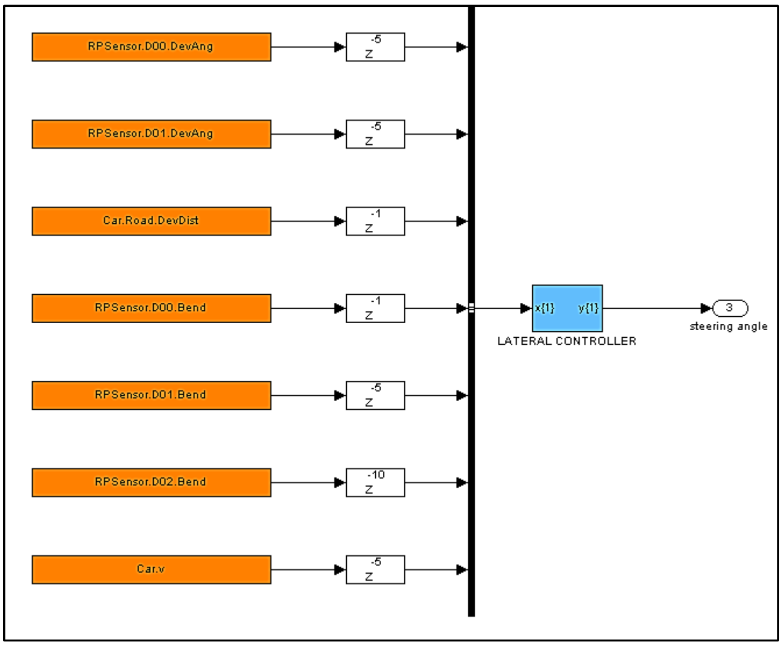

Figure 5.5 : Input-output selection for lateral human driver model.

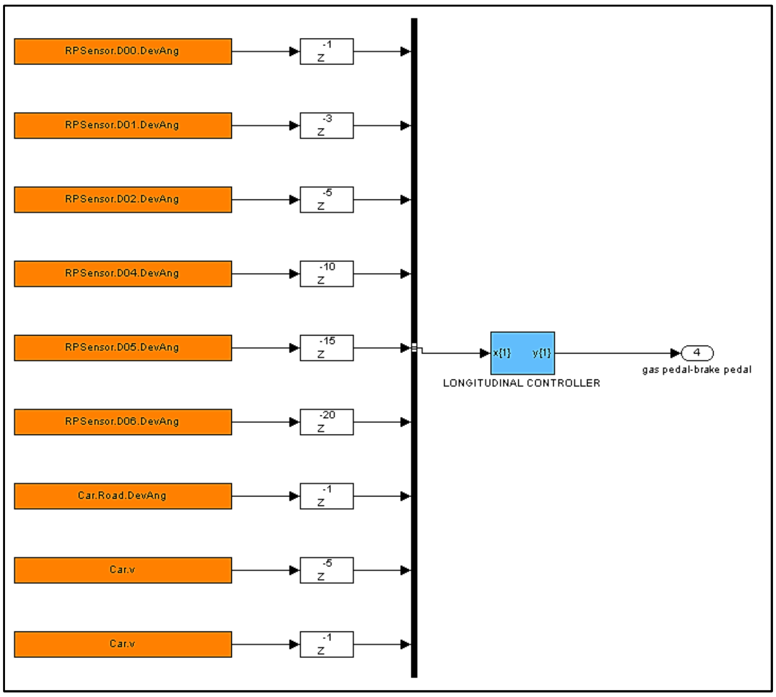

Figure 5.6 : Input-output selection for longitudinal human driver model.

The terminology of selected inputs is defined as follows;

RPSensor.D0X.DevAng :Deviation angle acquired from road preview sensor with identity X.

RPSensor.D0X.Bend : Road bend acquired from road preview sensor with identity X.

Car.v : Vehicle velocity

Car.Road.DevDist : Deviation distance of car from driving lane at vehicle position

Car.Road.DevAngle : Deviation distance of car from driving lane at vehicle position

where X corresponds to number of preview sensors positioned at specific distances in front of vehicle. The concept of Road Bend is shown in Figure 5.7, while concept of Deviation angle and Deviation distance is shown in Figure 5.8.

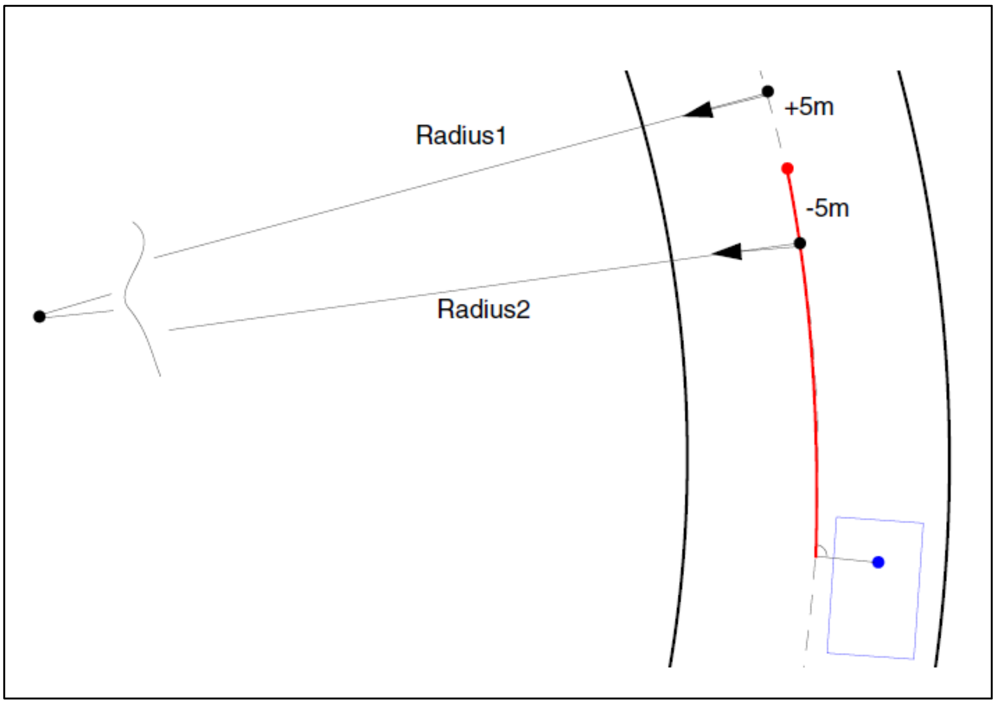

Figure 5.7 : Concept of road bend.

5.3.2 Driver modeling using artificial neural network (ANN)

ANN (Artificial Neural Network) is one of the techniques to identify relation between inputs and outputs selected in Chapter 5.3.1. Levenberg-Marquardt algorithm reviewed in Chapter 4.1 is used as training algorithm for ANN in this thesis. MATLAB® Neural Network Toolbox is used for application of ANN modeling technique to human driver modeling.

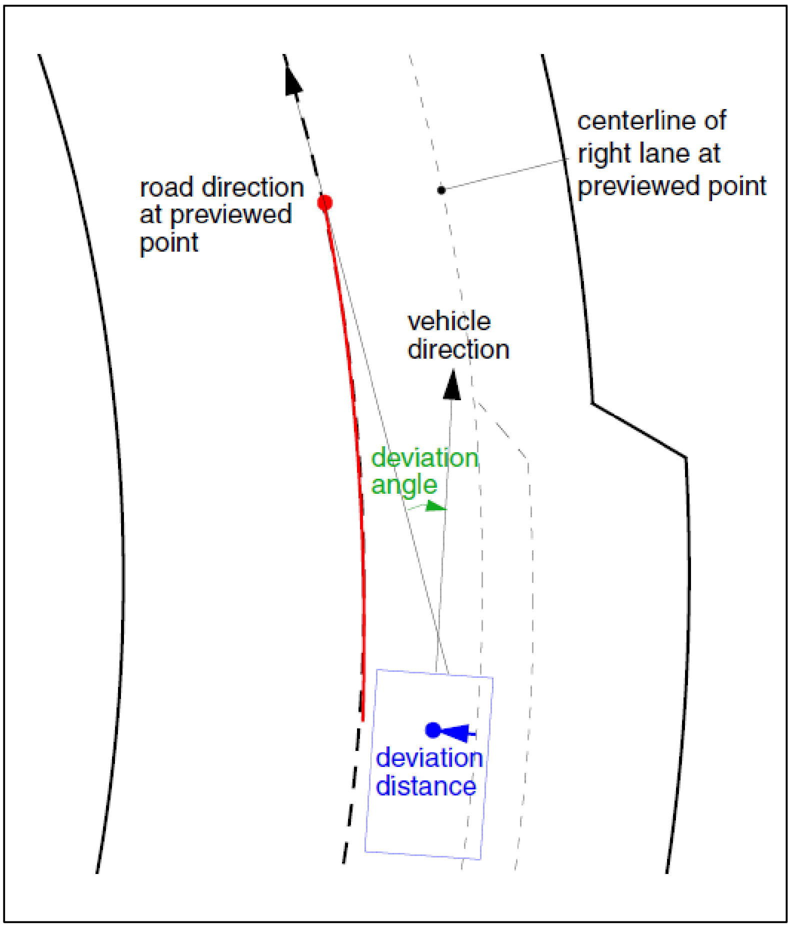

Figure 5.8 : Concept of deviation angle and deviation distance.

There are several design parameters of ANN architecture to be decided. These parameters should be selected carefully to obtain a robust model. The critical design parameters can be listed as;

- Number of layers
- Number of neurons in each layer
- Types of activation functions used in neurons
- Number of iterations for offline (batch) training

There are no strictly defined criteria to decide design parameters, so they are mainly selected with trial and error procedure based on training and validation performance. It is expected that number of free parameters (number of layers and number of neurons in layers) of ANN tends to increase for robust modeling when number of inputs increases. This is because of the fact that if outputs of a dynamic system are affected from many inputs then there should be more free parameters of ANN to capture complex relation between inputs and outputs. The number of training iterations should be low enough for ANN being able to generalize the input-output relation without over fitting and high enough to capture dynamic relation between inputs and outputs. Types of activation functions are generally selected as saturating differentiable functions to restrict input range for training, which increases speed of convergence and sensitivity of training algorithms. Linear activation functions are preferred in output layer without saturation so that dynamic relation between inputs and outputs can be reserved and output prediction error can always be propagated with constant derivative coefficient.

Based on these assumptions several trial-error iterations are performed for selection of design parameters. Each trial is cross checked for training and validation performance. The final decision for ANN design parameters can be seen in Table 5.1.

Table 5.1 : Selection of design parameters for ANN.

Number of Neurons		Activation function				Number of training iterations	
Layer 1 (Hidden layer)		Hidden layer		Output layer			
Lateral controller	Longitudinal controller	Lateral controller	Longitudinal controller	Lateral controller	Longitudinal controller	Lateral controller	Longitudinal controller
10	20	Tangent sigmoid	Tangent sigmoid	Linear	Linear	50	100

5.3.3 Driver modeling using adaptive neuro fuzzy inference system (ANFIS)

ANFIS structure and algorithm are reviewed in Chapter 4.3. Similar to ANN modeling structure, ANFIS also needs some design parameters to be decided. These parameters can be listed as;

- Types of input membership functions
- Number of input membership functions for each input

- Number of iterations for offline (batch) training

As with ANN modeling approach, there are no strictly defined criteria to decide design parameters, so they are mainly selected with trial and error procedure based on training and validation performance. Number of membership functions for each input defines the number of rules and should not be selected high because number of rules increases exponentially with increasing number of inputs and number of membership functions. The type of all input membership functions are selected as Gaussian membership function since it is close to normal distribution and appropriate from both human decision making and statistical considerations point of view.

Determination of number of rules can be supported with other methods like fuzzy clustering and cluster validity measures that can be found in literature (Babuska,1998; Abonyi and Feil, 2007). This approach may support determination of number of input membership functions initially as rough estimation. In addition, if there is an estimation of number of operation regions for an input, this could be useful for determining number of membership functions for the input considered. However, these methods does not provide exact results. The final decision for ANFIS design parameters can be seen in Table 5.2.

Table 5.2 : Selection of design parameters for ANFIS.

Number of membership function for each inputs																Membership function type				Number of training iterations	
Lateral controller							Longitudinal controller									Inputs		Outputs		Lateral controller	Longitudinal controller
Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Lateral	Longitudinal	Lateral	Longitudinal		
2	1	1	2	2	1	2	2	2	2	1	2	1	2	1	2	Gaussian	Gaussian	Linear	Linear	5	5

5.4 Comparision of Results and Validation of Driver Model on CarMaker®

Validation is an essential step to be able to assert that the driver model exhibits characteristics of real human driver. Percentile Variance (PV) that is defined with Equation (3.9) is used for initial estimation of how much validation results for driver model and real human driver fits each other. PV validation methodology is useful to assess performance of driver model before application on CarMaker®, which saves time. PV measures can be seen in Table 5.3 for ANN and ANFIS showing both training and validation fitting of driver models with real human driver. The considerable drop of performance for validation can be considered as memorization problem at first look, but the real performance can be understood when application of models are performed on CarMaker®.

Table 5.3 : Percentile variance (PV) measures for obtained models.

Training PV [%]				Validation PV [%]			
Lateral control		Longitudinal control		Lateral control		Longitudinal control	
ANN model	ANFIS model	ANN model	ANFIS model	ANN model	ANFIS model	ANN model	ANFIS model
86	86	68	65	42	53	21	24

As another comparison, outputs of ANN and ANFIS models together with human driver outputs are plotted in Figure 5.9 for training data and in Figure 5.10 for validation data. Data for training is collected from road profile in Figure 5.2 and data for validation is collected from road profile in Figure 5.3 with real human driver in the loop simulation on CarMaker®. Samples of the collected data with real human driver are used to simulateoutputs of driver models without CarMaker® simulation. The distortions of driver model outputs from human driver outputs for validation supports the low PV measures in Table 5.3.

If PV measures show reasonable fit in training, then the models are applied on CarMaker® for final validation. Simulink® models of ANFIS structure for driver model is obtained by MATLAB® Fuzzy Logic Toolbox® and Simulink® models of ANN are obtained by MATLAB® Neural Network Toolbox®. The obtained Simulink® models are then embedded into CarMaker® for Simulink® to test performance of models in closed loop operation with vehicle dynamics. It is seen from online video animations that the driver models for ANN and ANFIS can successfully track the road profile that is selected for validation. This shows that human driver characteristics are modeled successfully. Therefore, the low PV measures in Table 5.3 for validation can be assumed as internal vagueness of the human driver, which means human driver may change his driving characteristics in

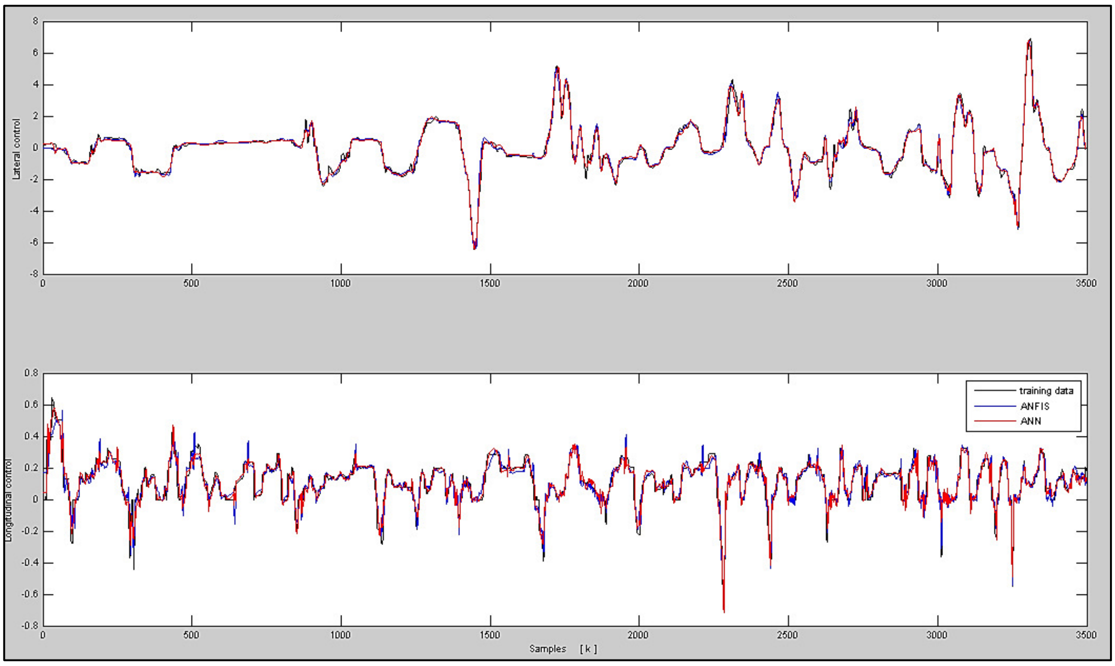

Figure 5.9 : Model and human driver outputs for training data.

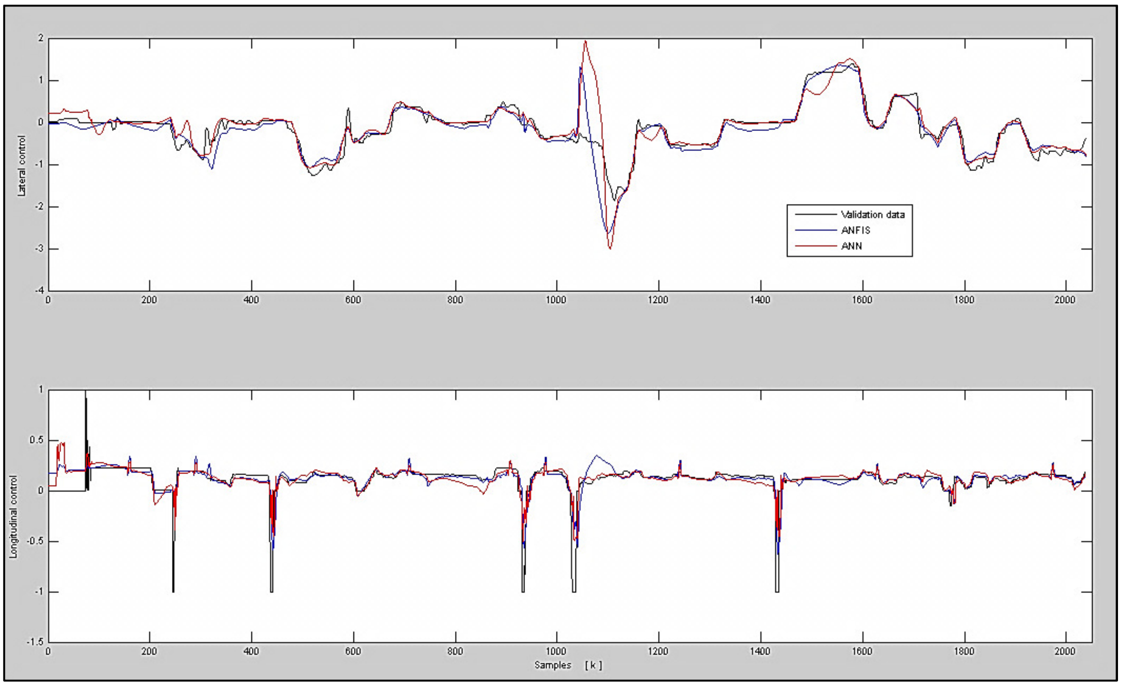

Figure 5.10 : Model and human driver outputs for validation data.

different environment conditions. It is also interesting to note that longitudinal control has much lower PV measures than lateral control for both training and validation although it is seen from CarMaker® application that longitudinal controller is successful. This means human driver has much more internal vagueness for longitudinal control actions.

The performance of driver models, human driver and IPG driver for tracking the road profile used for validation (Figure 5.2) can be seen in Figure 5.11. Tracking performance for individual road segments marked in Figure 5.11 can be seen in Appendix A as zoomed views. Vehicle speed profile comparision between driver models, human driver and IPG driver for tracking road profile used for validation can be seen in Figure 5.12. Also, Figure 5.13 shows deviation distance of car from road centerline for driver models, human driver and IPG driver while trackingroad profile used for validation. Figure 5.11, Figure 5.12 and 5.13 are plotted using the results from closed loop CarMaker® simulations with driver models, human driver and IPG driver in the loop seperately. Figure 5.12 can be used for assesment of longitudinal control performance while Figure 5.11 and Figure 5.13 can be used for assesment of lateral control performance of driver models, human driver and IPG driver.

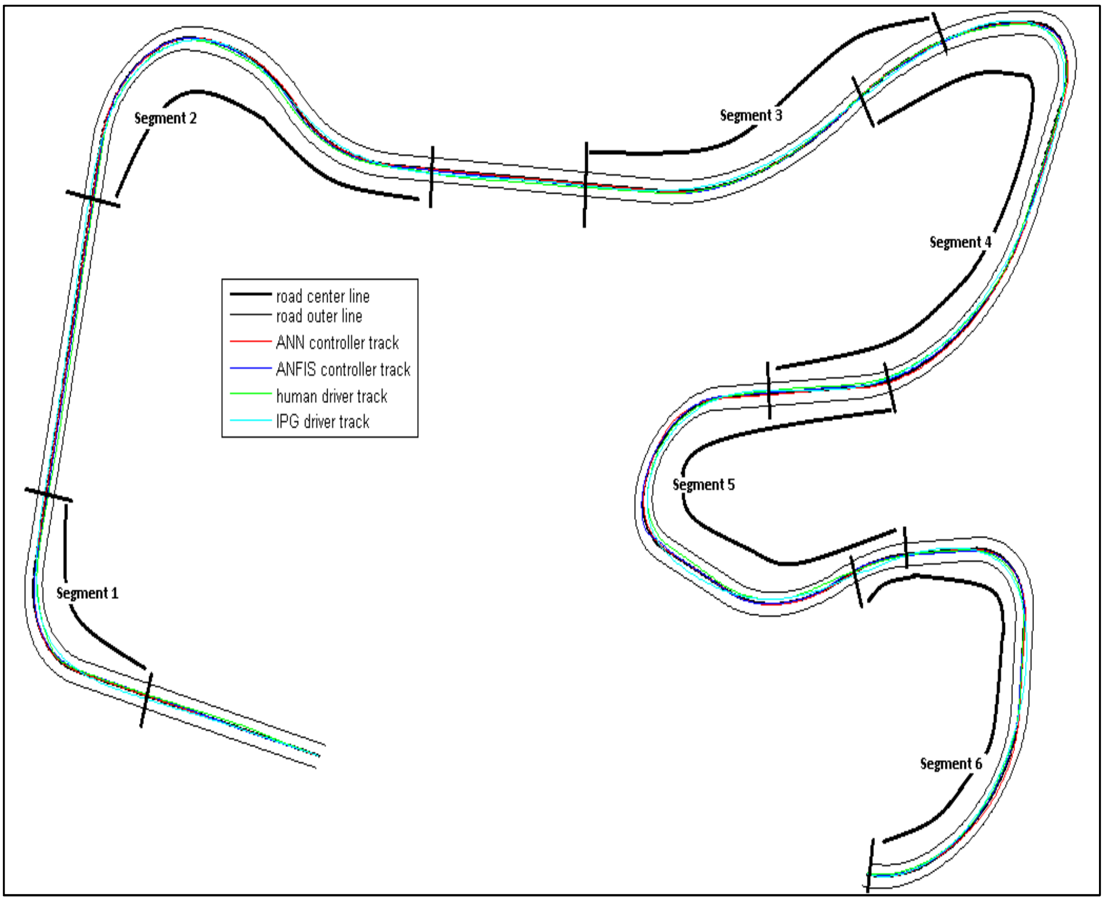

Figure 5.11 : Performance of driver models, human driver and IPG driver for tracking the road profile used for validation.

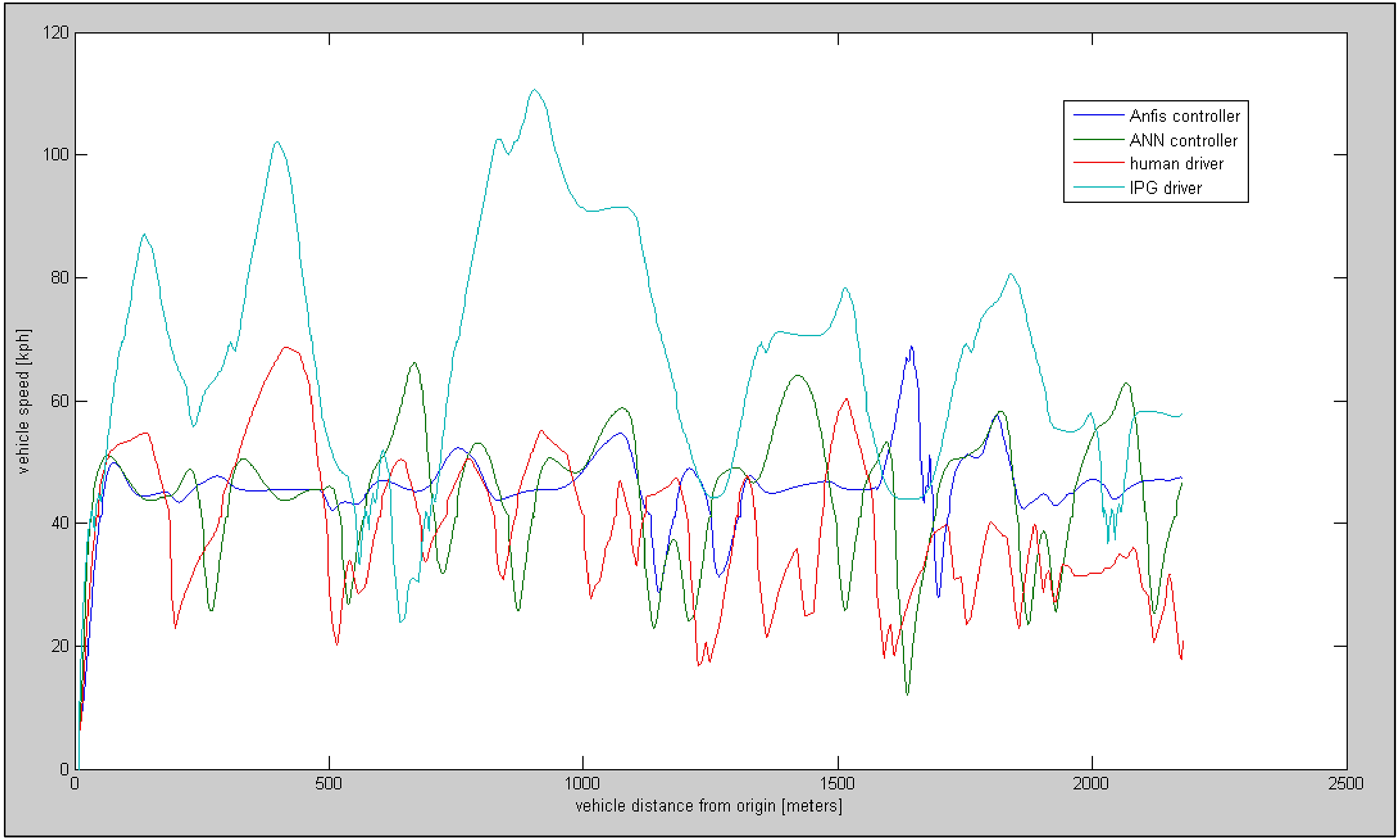

Figure 5.12 : Vehicle speed profile comparision between driver models, human driver, and IPG driver for tracking road profile used for validation.

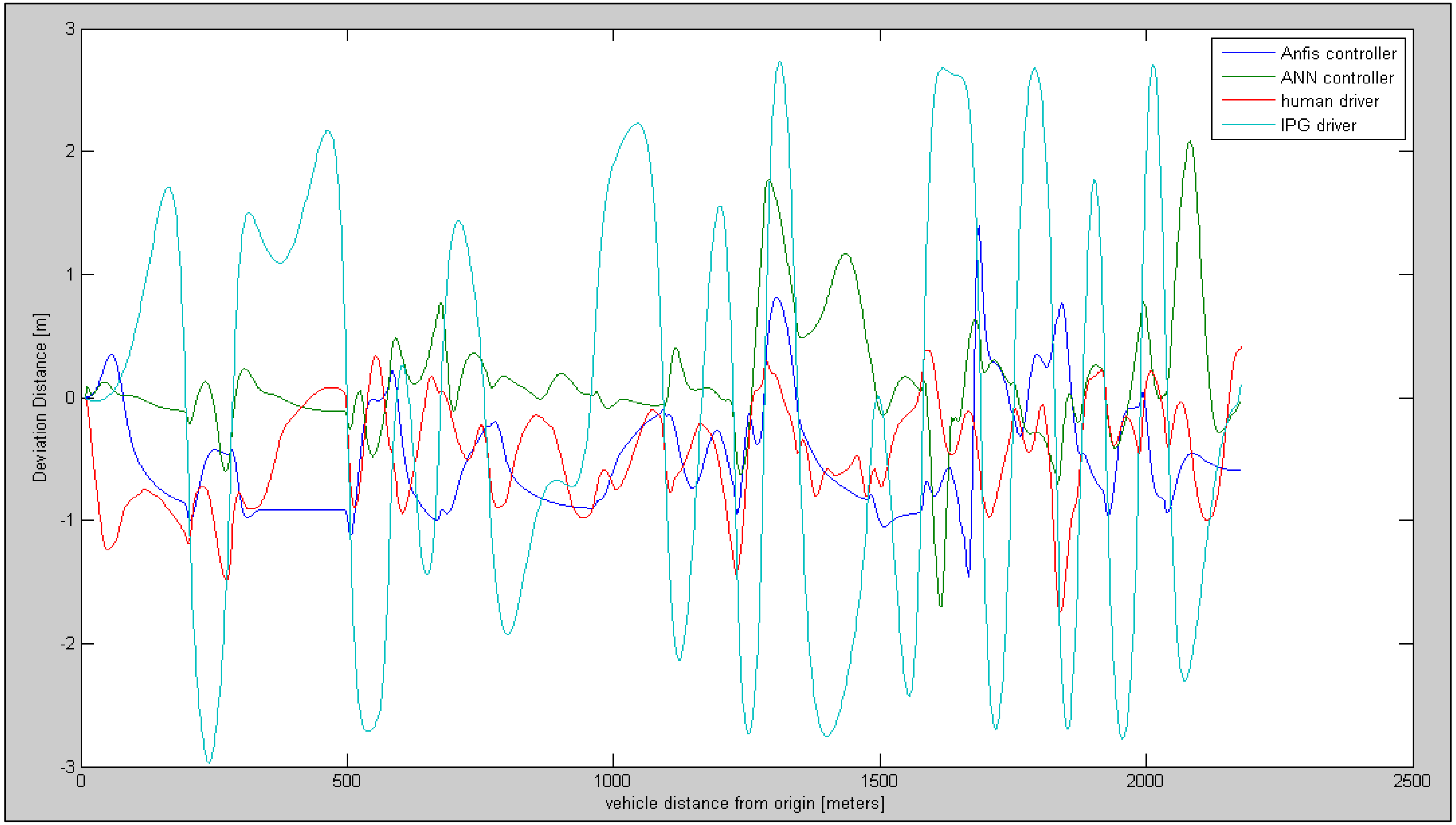

Figure 5.13 : Deviation distance comparision between driver models, human driver, and IPG driver for tracking road profile used for validation [left lane (+), right lane(-)].

6. CONCLUSIONS AND RECOMMENDATIONS

In this study, intelligent modeling algorithms are reviewed for application to human driver modeling. Driver model is obtained as two simultaneously working controllers as lateral and longitudinal by using two intelligent modeling techniques ANFIS and ANN. Several advantages of modeling human driver lateral and longitudinal control actions are considered;

- Driver model is simplified by using the opportunity of input signal identification separately for different human driver characteristics.
- Statistical approaches can be combined with expert knowledge for identification by decoupling complex nonlinear human driver into sub modules. Therefore, physical understanding of human driver behavior can be integrated into modeling work more easily.

However, decoupling of human driver model may cause instability when intended to be used simultaneously as a single driver model. Therefore, validation is an essential step to assess performance of obtained models. PV method is used to assess initial performance of driver models obtained. The final validation is performed with closed loop operation of obtained driver models with vehicle dynamics. For this purpose, driver models are embedded into CarMaker® software and it is seen from visual online animations that performance of driver model to track different road profile is very high. Therefore, it is seen that two decoupled models obtained from real human driver data by using intelligent modeling techniques act as a single driver model successfully.

Identification of correct input-output variables for human driver model is one of the most critical steps for human driver modeling. The identification methods of input variables in literature are mostly limited to application for SISO linear systems. There are also some methods developed for MIMO nonlinear systems such as FNN (False Nearest Neighbour) algorithm defined in Abonyi et al. (2004) and Lipschitz coefficient algorithm defined in Bomberger and Seborg(1998), but their performance are not satisfactory. Also, these algorithms are based on the approach that all the inputs candidates for the system are available before the analysis, which arises many combinations. In addition, these algorithms do not propose certain measures for input set selection possibilities. Therefore, statistical correlation analysis is performed to identify the effect of individual inputs and their delay on output in combination with validation procedure. Some of the input-outputs are selected with expert knowledge. For example,vehicle speed is considered as a critical input for steering control because driver tends to make small steering action at velocity to preserve vehicle stability. Also, steering angle is selected instead of steering moment, so human decision is modeled as output instead of neuro-muscluar actions.

Design parameters of ANN and ANFIS are mostly used by trial and error procedure. The purpose is to obtain simplest structure that is effective to exhibit human driver dynamic characteristics. This approach will empower generalization capability of the obtained driver model to achieve good validation performance. If design parameters are selected in a way that results in unnecessary number of free parameters, then memorization problem can be encountered. Overtraining of the model structures with selected intelligent modeling algorithm is another reason for memorization problem. Especially, ANFIS two way learning algorithm is stuck with computer memory problems in computational application when number of rules is big due to big number of input membership functions. Determination of number of rules for ANFIS can be supported with cluster validity measures in literature (Babuska, 1998).

In addition to ANFIS and ANN, two new hybrid intelligent modeling algorithms are studied in this thesis. These algorithms are;

- FCM clustering based on GA algorithm for projection.
- Online MTLA (Minimum Tracking Learning Algorithm) application to TS fuzzy structure.

Both algorithms are developed as general MIMO intelligent modeling techniques. The first algorithm is intended to achieve practical application of fuzzy cluster projection approach that is proposed in Babuska (1998). The clusters obtained by FCM algorithm should be projected into axes of input variables as antecedent membership functions. Hovewer, this is a complex problem for practical computational application since cluster data is distributed and it is very difficult to fit this data into selected prototype memberhip functions on axes of input variables. At this point, GA algorithm is used to tune width of Gaussian membership functions while centers of the Gaussian membership functions are obtained from FCM clustering. A membership matrix is generated by FCM algorithm using collected human driver model data. Another membership matrix is generated by Gaussian membership functions whose centers correspond to centers of FCM obtained clusters while their widths are members of GA population. GA algorithm tries to optimize cost function that is norm of difference between these two membership matrices. Consequent parameters are calculated with weighted least squares method using the membership matrix calculated by FCM clustering algorithm.

The second algorithm is based on application of MTLA algorithm that is proposed in n Zhao (1996). This algorithm is based on minimum tracking a cost function based on the current sample error of model output that is time-dependent. The algorithm ensures exponential convergence. This algorithm is originally proposed for application to ANN structure, but in this thesis it is revised for application to TS fuzzy model parameter identification. All antecedent and consequent parameters of TS fuzzy model are considered like weights of an ANN. Therefore, with this algorithm, it is aimed that a structure similar to ANFIS is derived with an online training approach.

Both of the two algorithms developed in this thesis are computationally very demanding and could not be applied to complex nonlinear driver modeling problem for that reason. The first offline algorithm is very slow due to nature of GA algorithm while second online algorithm has big symbolic matrices to be processed in MATLAB®, which is very slow for online application. These algorithms can be studied further from computational point of view with more efficient compilers or with high performance workstation PC.

Another result derived in the context of this thesis is that characterizing longitudinal driving actions of human driver is much more difficult than lateral control behavior. The PV analysis shows that both training and validation results of ANFIS and ANN architectures are worse than lateral control performance of driver model. However, it is interesting to note that obtained longitudinal driver models are very successful in closed loop simulation with vehicle dynamics on CarMaker®. This shows that internal vagueness of human driver for longitudinal control of vehicle is much more than lateral control actions. Actually, this makes sense because in lateral control action, driver has strict tolerances to track road shape and steering angle position can be considered as a function which depends greatly on road curvature. In longitudinal control, driver can decide to take same turn with different speeds in different driving loops even if all environmental and dynamic conditions are same. This internal vagueness directly affects control of gas and brake pedals.The effect of internal vaguneness in longitudinal control actions can also be observed from Figure 5.12. In this graph, it is seen thatinstantaneousvehicle speed is different for driver models and human driver in the same positions of road profile used for validation, while IPG driver speed profile distorts much more from speed profile of human driver and driver models.

The driver model in this study is just intended to characterize longitudinal and lateral control action of a driver to track a defined road profile. For future studies, traffic information and road inclination can also be included in the driver modeling task. Traffic and road inclination simulations are possible with CarMaker® with all necessary data logging. Also, driver model can be improved by integrating driving time as one of the inputs, which may be a new approach to include tiring of human driver in long driving. In addition, CarMaker® software does not support sound feedback from engine, so a module can be developed for CarMaker® to make the human driver can also have sound feedback. This module can be developed using sound processing methods on real engine sound data obtained from real driving loops. Moreover, if simulation environment is supported with more advanced hardware systems like Stewart platform incorporated with hydraulic or other types of actuators, then more physical effects of real driving conditions can be included in human driver modeling.

REFERENCES

Abonyi, J. and Feil B. (2007). Cluster analysis for data mining and system identification.Boston: Birkhauser.

Abonyi, J., Feil, B. and Szeifert, F. (2004). Model order selection of nonlinear input-output models-a clustering based approach, *Journal of Process Control,**14***, 593-602.

An, P.E., Brown, M., Harris, C.J. (1994). On real time driver modeling and vehicle guidance within Prometheus, *IFAC Transportation Systems Tianjin*, 91-96.

An P.E. and Harris C.J. (1996). An intelligent driver warning system for vehicle collision avoidance. *IEEE transactions on systems, man, and cybernetics-Part A: systems and humans*, ***26***(2), 254-261.

Arslan, H. (1999). Yapay sinir ağları ile robotlarda hareket kontrolü, (MSc Thesis), İ.T.Ü. Fen BilimleriEnstitüsü.

Atabay, O. (2004). Sürücü-Taşıt-Çevre etkileşim etütlerine yönelik bir s ürüş simülatörü geliştirilmesi, (PhD Thesis), İ.T.Ü. Fen BilimleriEnstitüsü, İstanbul.

Bau, D. and Trefethen, L.N. (1997). Numerical linear algebra. Philadelphia: Society for Industrial and Applied Mathematics.

Babuska, R. (1998). Fuzzy modeling for control. Boston: Kluwer Academic Publishers.

Bezdek, J. (1981). Pattern recognition with fuzzy objective function. New York: Plenium Press.

Bomberger, J.D. and Seborg D.E. (1998). Determination of model order for NARX models directly from input-output data. *J. Proc. Cont.*, ***8***(5), 459-468.

Delice, İ. (2004). Taşıtın yanal ve doğrusal kontrolü için sürücünün modellenmesi, (MSc Thesis), İ.T.Ü. Fen BilimleriEnstitüsü, İstanbul.

Ertuğrul, H. (2009). İnsan operatörlerin modellenmesi yöntemi ile bulanık denetleyici tasarımı, (MSc Thesis), İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

Fenton, R. E. (1994). IVHS/AHS: Driving into the future, *IEEE Control Systems Magazine*, 14(6), 13-20.

Fujioka, T., Nomura, K., Takubo, N. and Chung, D.H. (1991). Driver model using neural network system. *Proceedings of Sixth International Pasific Conference on Automotive Engineering*, **2**, 931-938.

Golberg D.E. (1989). Genetic algorithms in search, optimization, and machine learning. Boston: Addison-Wesley. Michigan: University of Michigan Press.

Hagan, M.T. and Menjah, M. (1994). Training feedforward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks*, ***5***(6), 989-993.

Ivanov, V. and Platz, G.K. (2011). Fuzzy methods in ground vehicle engineering: State-of-the-Art and advanced applications. *Proceedings of the 8th International Conference on Structural Dynamics*, EURODYN 2011, Leuven, Belgium, 4-6 July 2011.

Jang, J.S. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference systems, *IEEE Transactions on Systems, Man and Cybernetics*, ***23***, 665-685.

Jang, J.S., Sun T. and Mizutani, E. (1997). Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. New Jersey: Prentice Hall.

Kendall, M.G. (1955). Rank correlation methods. Charles Griffin & Co.

Kramer, U. (1985). On the application of fuzzy sets to the analysis of the system driver-vehicle-environment, *Automatica*, ***21***, 101-107.

Krishnaswami, V., Kim, Y.W. and Rizzoni, G. (1995). A new model order identification algorithm with application to automobile oxygen sensor modeling, *Proceedings of the American Control conference*, ***3***, 2113-2117.

Ljung, L. (1999). System identification-Theory for the user. New Jersey: Prentice Hall.

MacAdam, C. and Johnson, G.E. (1996). Application of elementary neural networks and preview sensors for representing driver steering control behavior, *Vehicle System Dynamics*, ***25(1)***, 3-30.

McRueer, D.T. and Krendel, E.S. (1959). The human operator as a servo system element. Part I + II, *Journal of the Franklin Institute*, ***267***(5/6), 381-403, 511-536.

Mitschke, M.(1996). Verifikation des fahrermodells für kritische fahrsituationen (3. Fahrerverhalten bei normaler Kurvenfahrt). Bericht Nr. 744. Institut für Fahrzeugtechnik, Technische Universitaet Braunschweig.

Plöchl, M. and Edelmann, J.(2007).Driver models in automobile dynamics application, *Vehicle System Dynamics, **45***, 699-741.

Shim, J. S., Yoon, Y.Y., Heo, S.J. andYoo, Y.M. (1995). Closed loop simulation of a vehicle system with an artificial driver, *Mech. Struct. & Mach.*, **23**(1), 87-113.

Soper, H.E., Young, A.W., Cave, B.M., Lee, A. and Pearson, K. (1917). On the distribution of the correlation coefficient in small samples. Appendix II to the papers of "Student" and R.A. Fisher. A co-operative study, *Biometrika*, ***11***, 328-413.

Söderström, T. and Stoica, P. (1989). System identification. New York: Prentice Hall.

Spearman C.(1904). The proof and measurement of association between two things, *American Journal of Psychology,**15***, 72-101.

Sugeno M. and Nishida M. (1985). Fuzzy control of model car, *Fuzzy sets and Systems*, **16**, 103-113.

Tsoukalas, L.H., Uhrig, R. E. (1997). Fuzzy neural approaches in engineering. New York: Wiley.

Tunçer, Ö. (2010).Gerçek zamanlı şerit takip desteği sisteminin geliştirilmesi, (MSc Thesis), İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

Tustin, A. (1947). The nature of the operator's response in manual control and its implications for controller design, *IEEE Journal*, ***94***, 190-202.

Yan X. and Weng Z. S. (1988). An internal model approach in the modeling of human preview control, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, ***2***, 839-843.

Yager R.R. and Filev D.P. (1994). Essentials of fuzzy modeling and control. New York: Wiley.

Zhao, Y. (1996). On-line neural network learning algorithm with exponential convergence rate, *IEEE Electronics Letters Online*, ***32***(15), 1381-1382.

APPENDICES

APPENDIX A: Road tracking performance of human driver, driver models and IPG driver on road profile segments used for validation.

APPENDIX A: Road tracking performance of human driver,driver models and IPG driver on segments of road profile used for validation.

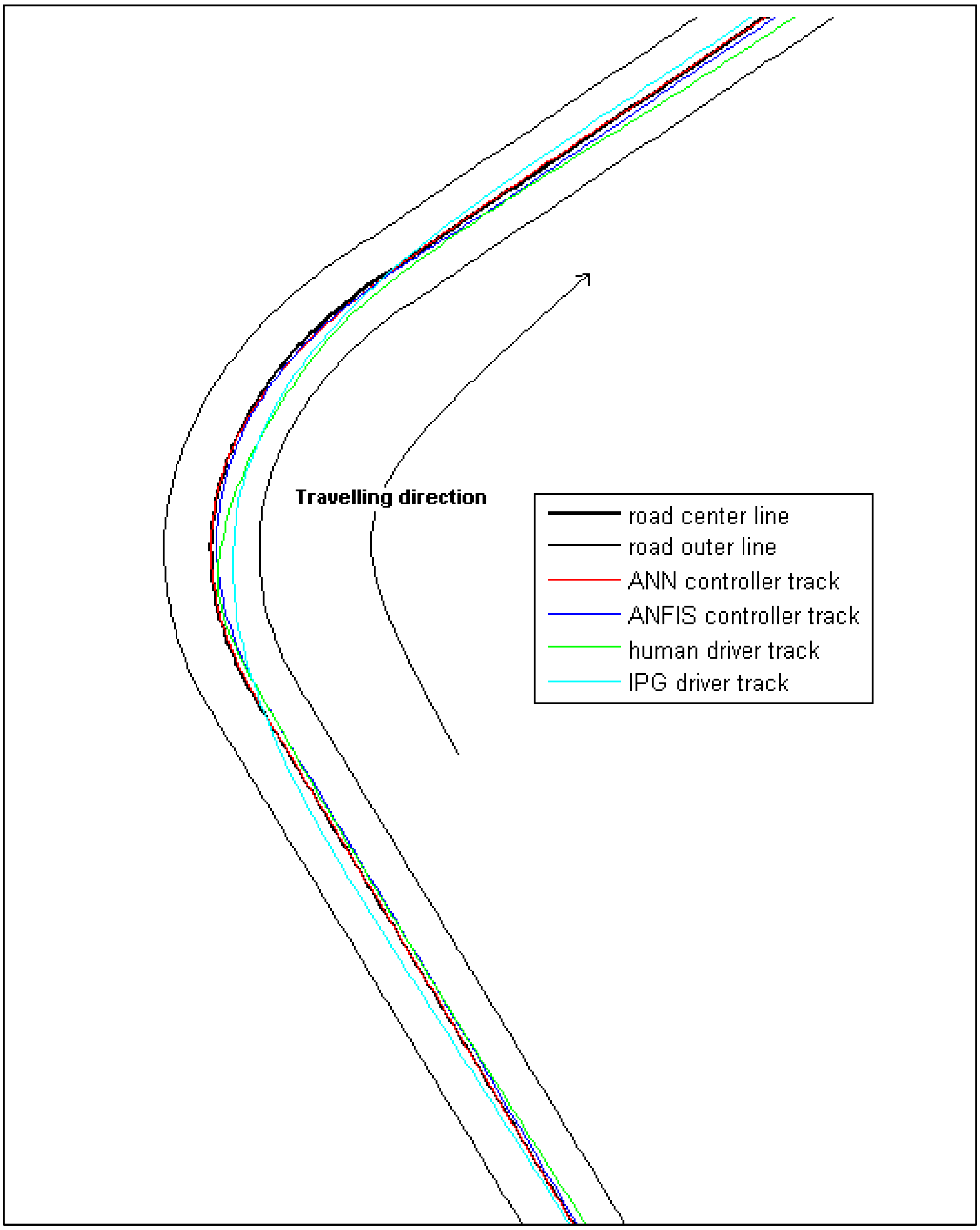

Figure A.1 : Performance of driver models, human driver and IPG driver for tracking the roadprofile segment 1.

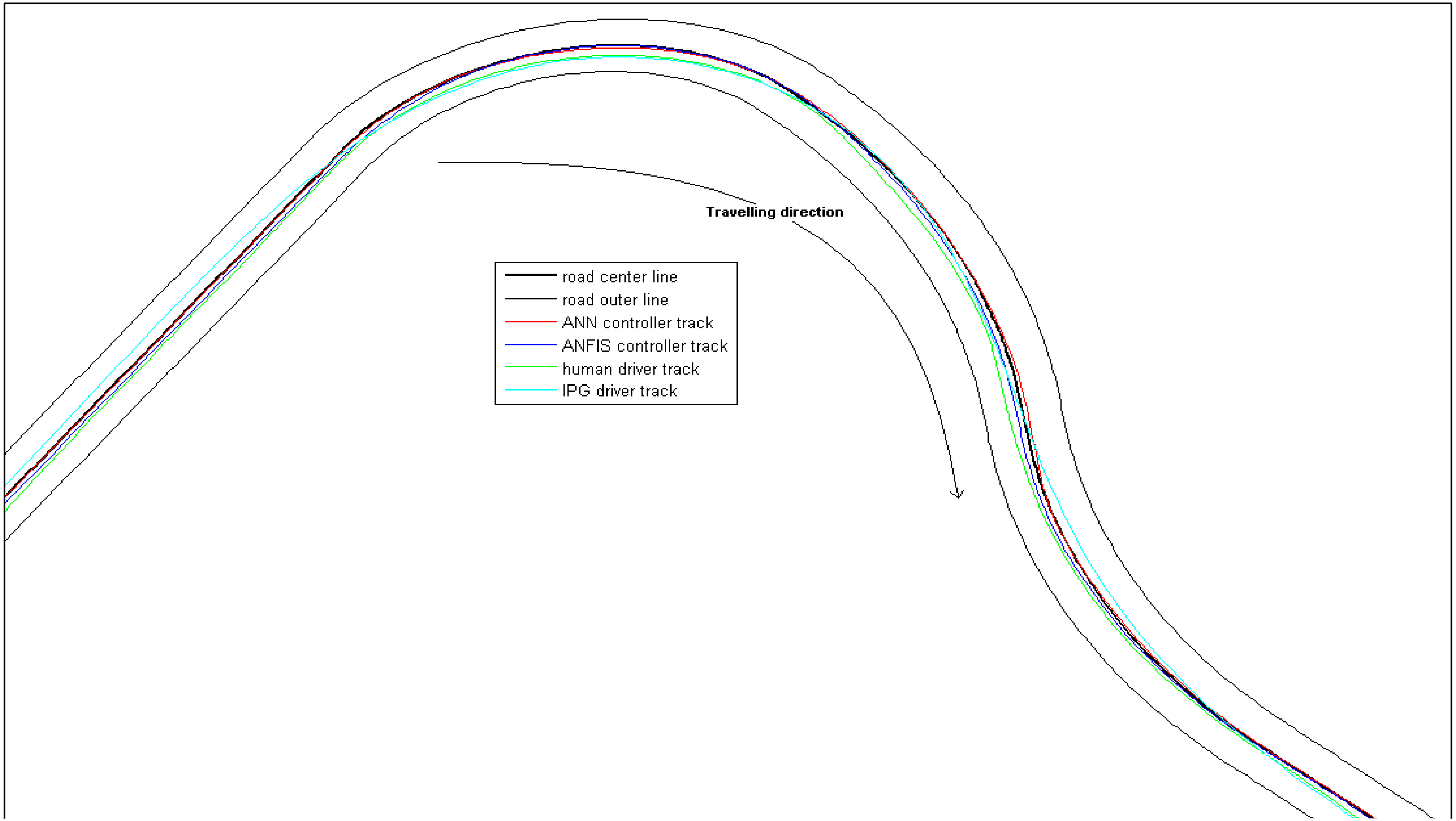

Figure A.2 : Performance of driver models, human driver and IPG driver for tracking the road profile segment 2.

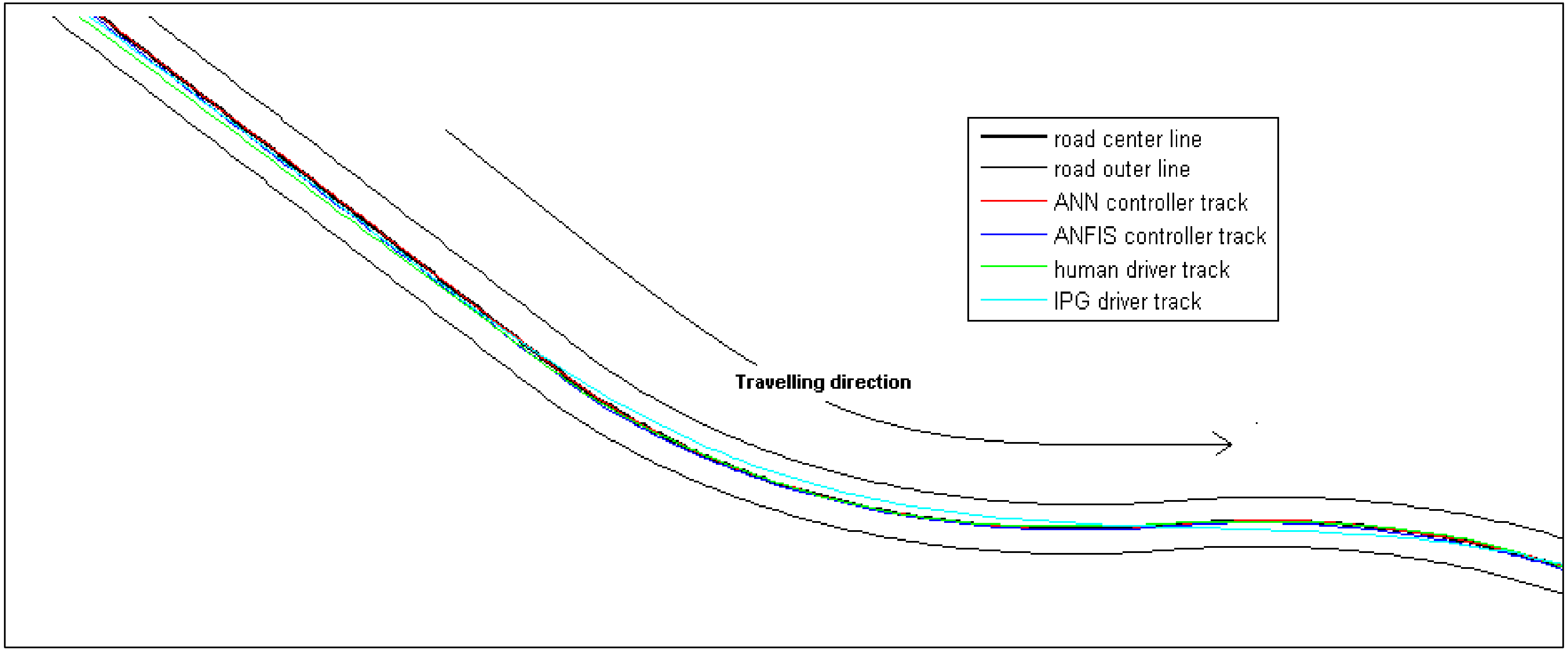

Figure A.3 : Performance of driver models, human driver and IPG driver for tracking the road profile segment 3.

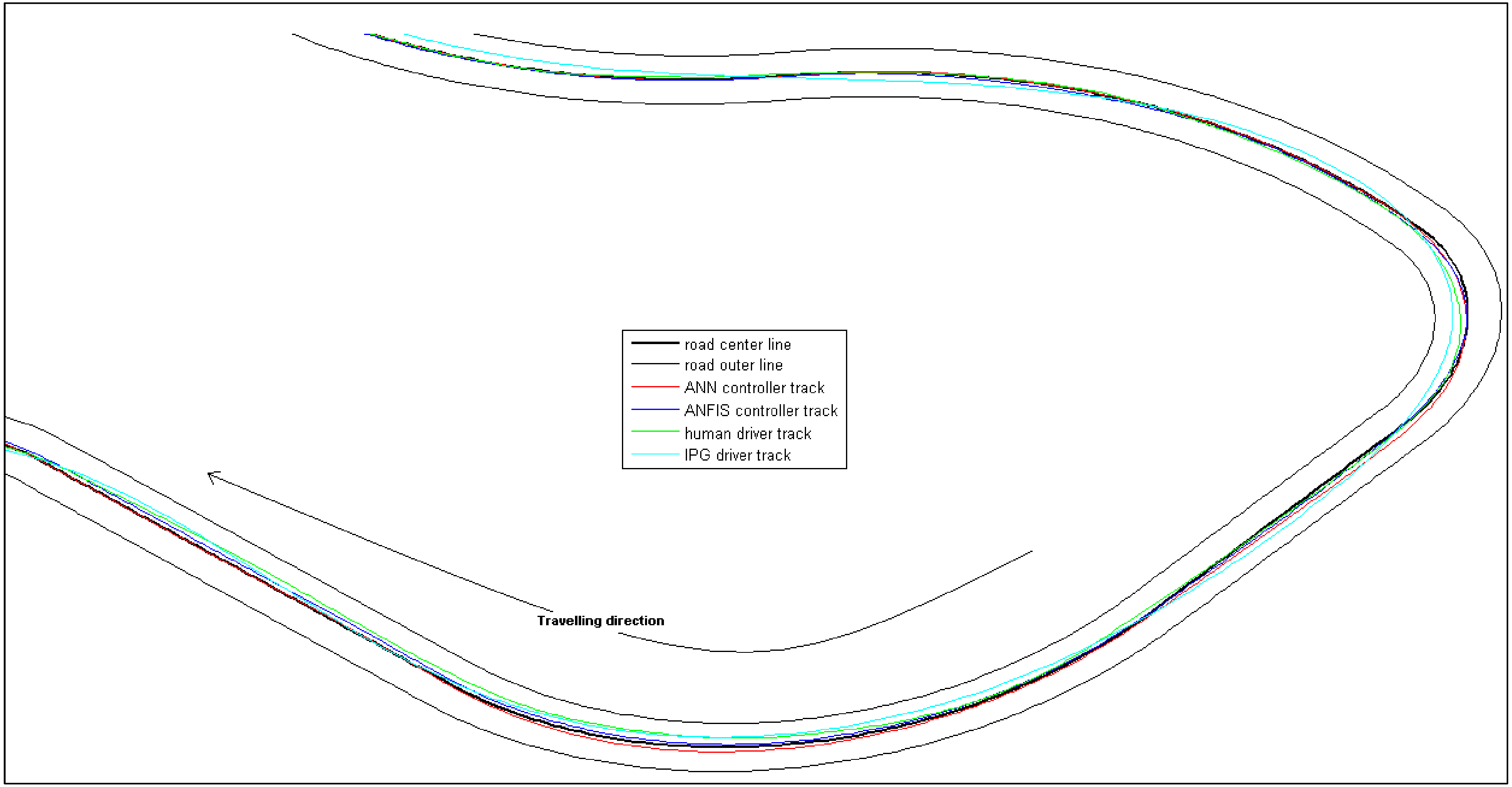

Figure A.4 : Performance of driver models, human driver and IPG driver for tracking the road profile segment 4.

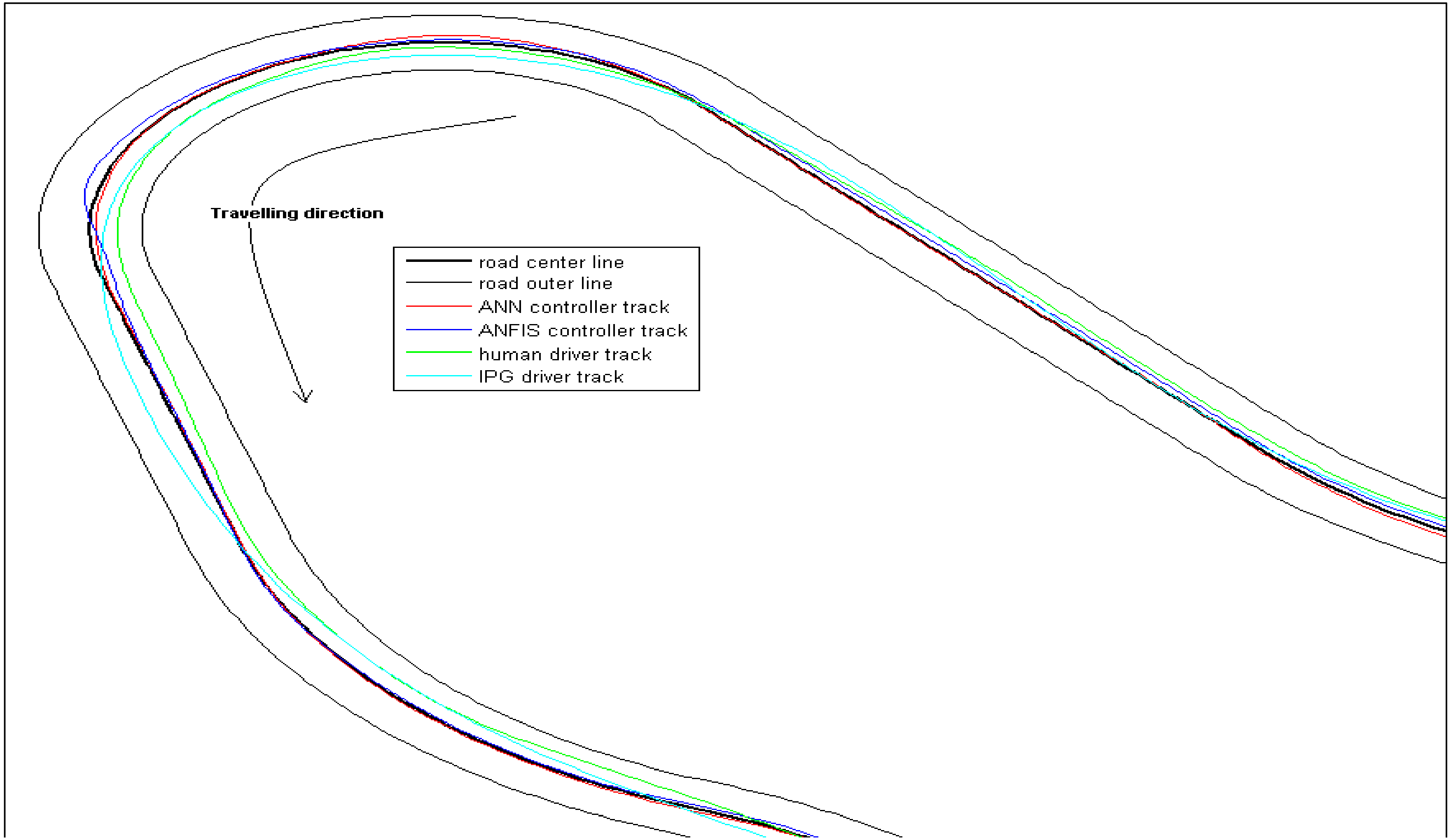

Figure A.5 : Performance of driver models, human driver and IPG driver for tracking the road profile segment 5.

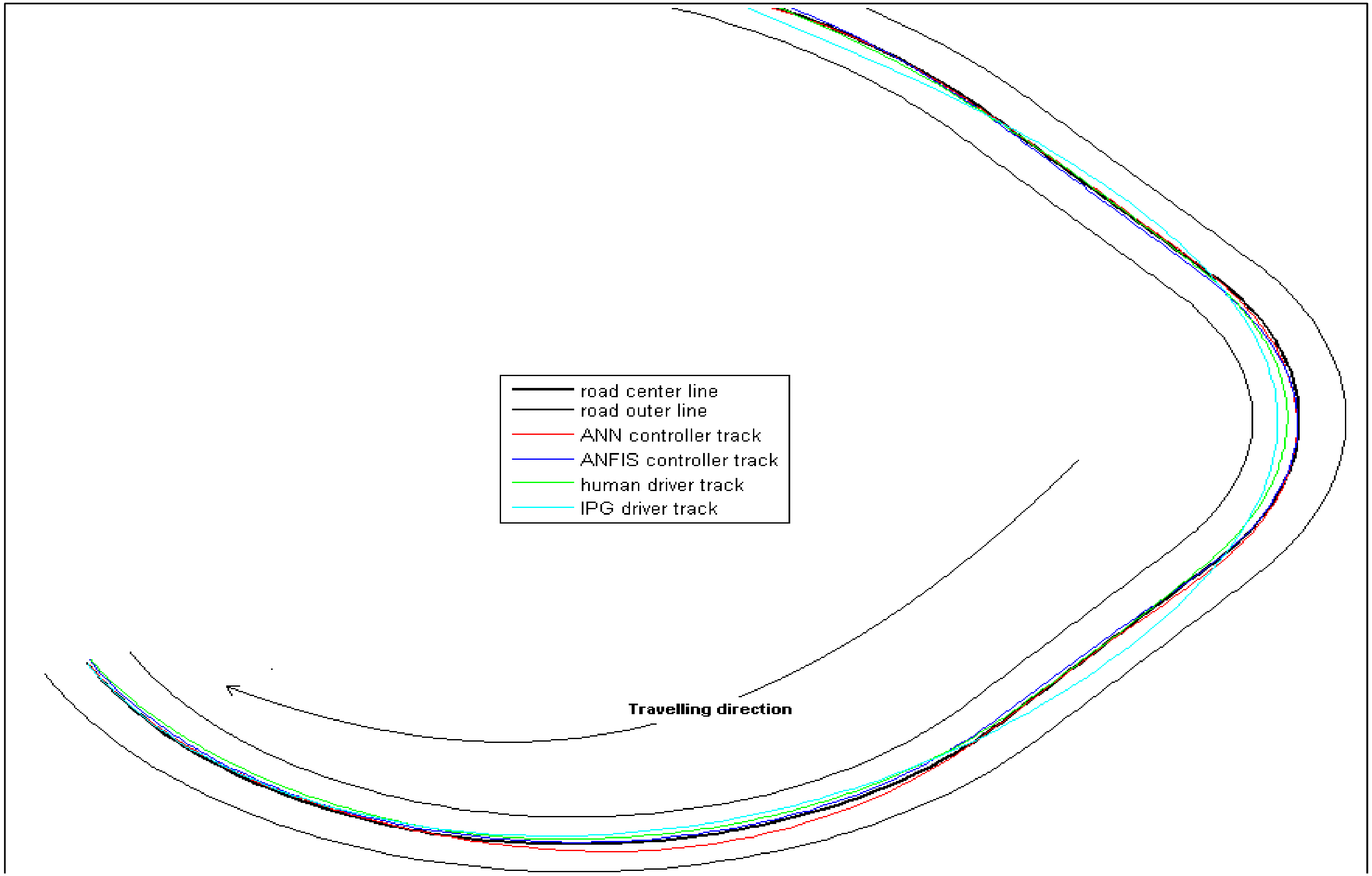

Figure A.6 : Performance of driver models, human driver and IPG driver for tracking the road profile segment 6.

CURRICULUM VITAE

Name Surname : **Emre Tekin**

Place and Date of Birth :**Gebze/ 08.07.1986**

Address :**Tatlıkuyu M. Ahmet PembegüllüBulvarı, No: 23/8, FulyaApartmanı, Gebze/Kocaeli/Turkey**

E-Mail :**marem41@hotmail.com**

B.Sc. :**[2004-2009] Middle East Technical University, Ankara/Turkey**

Professional Experience

- [2010-Present]Ford Otomotiv San. A.Ş., Gebze/Kocaeli, Calibration Engineer, Engine Calibration & Control Division.