

**REDUCTION OF TRANSVERSE VIBRATION OF
A SERPENTINE BELT SYSTEM: MODEL
PREDICTIVE CONTROL OF A BELT TENSIONER**

**AKSESUAR KAYIŞ SİSTEMLERİNDE ENİNE
(YANAL) TİTREŞİMİN AZALTILMASI: KAYIŞ
GERGİSİNİN MODEL ÖNGÖRÜLÜ KONTROLÜ**

VEYSEL MURAT ÖNAL

ASSIST. PROF. DR. CAN ULAŞ DOĞRUEK

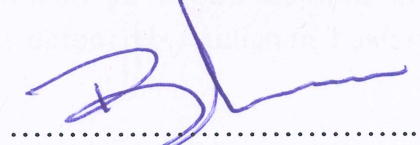
Supervisor

Submitted to Graduate School of Science and Engineering of
Hacettepe University as Partial Fulfillment to the Requirements for
the Award of the Degree of Master of Sciences in Mechanical
Engineering

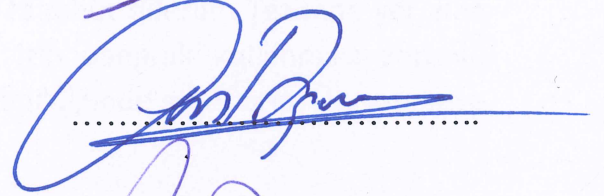
2017

This work named “**REDUCTION OF TRANSVERSE VIBRATION OF A SERPENTINE BELT SYSTEM: MODEL PREDICTIVE CONTROL OF A BELT TENSIONER**” by **VEYSEL MURAT ÖNAL** has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING** by the below mentioned Examining Committee Members.

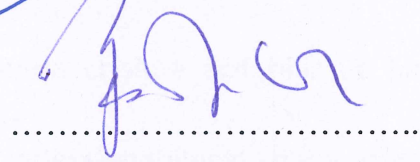
Prof. Dr. Bora YILDIRIM
Head



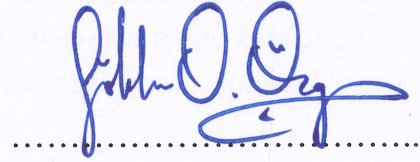
Assist. Prof. Dr. Can Ulaş DOĞRUER
Supervisor



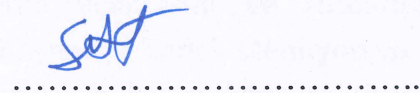
Assist. Prof. Dr. Kutluk Bilge ARIKAN
Member



Assist. Prof. Dr. Gökhan ÖZGEN
Member



Assist. Prof. Dr. Selçuk HİMMETOĞLU
Member



This thesis has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING** by Board of Directors of the Institute for Graduate School of Science and Engineering.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Director of the Institute of
Graduate School of Science and Engineering

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.

(Bu seçenекle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

- Tezimin 02/10/2017 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.

(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

- Tezimin/Raporumun tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.

- Serbest Seçenek/Yazarın Seçimi

21.09.2017



BEYSEL MURAT ÖNAL

To whoever supported to complete this thesis

ETHICS

In this thesis study, prepared in accordance with the spelling rules of Institute of Graduate Studies in Science of Hacettepe University,

I declare that

- All the information and documents have been obtained in the base of the academic rules
- All audio-visual and written information and results have been according to the rules of scientific ethics
- In case of using others works, related studies have been cited in according with the scientific standards
- All cited studies have been fully referenced
- I did not do any distortion in the data set
- And any part of this thesis has not been presented as another thesis study at this or any other university.

21/09/2017



VEYSEL MURAT ÖNAL

ÖZET

AKSESUAR KAYIŞ SİSTEMLERİNDE ENİNE (YANAL)

TİTREŞİMİN AZALTILMASI: KAYIŞ

GERGİSİNİN MODEL ÖNGÖRÜLÜ

KONTROLÜ

Veysel Murat ÖNAL

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Can Ulaş DOĞRUER

Haziran 2017, 141 sayfa

Serpantin kayış sistemi günümüz otomotiv endüstrisinde oldukça yaygın bir şekilde kullanılmaktadır. Motorda daha az yer kaplama, kayış kaymasını azaltma ve benzeri avantajları sebebiyle diğer sürüş sistemlerine nazaran daha fazla tercih edilmesi, serpantin kayışların titreşimlerinin kontrol edilmesini önemli kılmaktadır. Söz konusu titreşimin kontrol altına alınabilmesi adına şimdiye kadar çok sayıda gergi mekanizması kullanılmıştır. Bu çalışmada, Model Öngörülü Kontrol ile bir gergi kontrolü yapılmış ve sistemin performansı bahsi geçen kontrol sistemi kullanılmadığı haliyle karşılaştırılmak suretiyle bir çok örnek olay çalışmalarıyla değerlendirilmiştir. Bu çalışmada, bir çoklu yapı dinamik programı kullanılarak kayış-kasnak sistemi oluşturulmuştur. Sonraki adım ise; söz konusu dinamik modelin kullanılması suretiyle gerginin model öngörülü kontrolcü ile kontrolüdür. Söz konusu kayış kasnak sisteminin kontrolü için kullanılan model öngörülü kontrolcünün iç modelci kurulumunda ise yapay sinir ağlarından yararlanılmıştır. En son olarak, çoklu yapı dinamiği model sonuçlarının MPC'li, PID'li olarak karşılaştırılmasıyla çalışma tamamlanmıştır.

Anahtar Kelime: Aksesuar Kayışı, Serpantin Kayış, Model Öngörülü kontrol, Yapay Sinir Ağları, Kayış Gergisi, Çoklu Yapı Dinamiği, PID

ABSTRACT

REDUCTION OF TRANSVERSE VIBRATION OF A SERPENTINE BELT SYSTEM: MODEL PREDICTIVE CONTROL OF A BELT TENSIONER

Veysel Murat ÖNAL

**Master of Science Degree, Department of Mechanical
Engineering**

**Supervisor: Asst. Prof. Dr. Can Ulaş DOĞRUEK
June 2017, 141 pages**

The serpentine belt systems are widely used in today's automotive industry. The preference of serpentine belts in the market over the traditional belts due to the advantages such as consuming less space in the engine compartment, reduced slips etc. makes the vibration control of this type of belt crucial. In order to control the vibration of the belt, number of tensioner types are used. In this study, a tensioner which is controlled by model predictive algorithm is introduced to a belt system and system performance is compared to the belt system which is not controlled for various case studies. The belt and pulley system is designed with multibody dynamic simulation program in this study. The internal model of MPC, which is used to control the belt drive model, is established with the help of artificial neural networks. Finally, the study is completed by making comparisons on the results of the multi body dynamic model with MPC and the model controlled by PID.

Keywords: Accessory Belt, Serpentine Belt, Model Predictive Control, Neural Network, Tensioner, Multi Body Dynamic, PID

ACKNOWLEDGMENTS

Firstly, I would like to thank to my supervisor Asst. Prof. Dr. Can Ulař Dođruer for his guidance throughout this thesis procedure. Moreover, I wish to express my sincere thank to my dissertation committee members Prof. Dr. Bora Yıldırım, Asst. Prof. Dr. Kutluk Bilge Arıkan, Asst. Prof. Dr. Gökhan Özgen, Asst. Prof. Dr. Selçuk Himmetođlu for their supports and encouragement.

TABLE OF CONTENT

	Page
ÖZET.....	i
ABSTRACT.....	ii
ACKNOWLEDGMENTS.....	iv
LIST OF TABLES	vii
LIST OF FIGURES.....	vii
SYMBOLS AND ABBREVIATIONS	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Motivation	1
1.3 Thesis Objectives and Scope of Research.....	1
1.4 Organization and Content of Thesis.....	2
CHAPTER 2: LITERATURE REVIEW	3
2.1 Introduction.....	3
2.2 Serpentine Belt System	3
2.2.1 Tensioner Types	4
CHAPTER 3: THEORY	7
3.1 Multibody Dynamic	7
3.1.1 Belt Drive Model.....	8
3.1.2 Contact Forces.....	11
3.1.3 Distance Measurement	13
3.2. Introduction to Artificial Neural Networks	14
3.2.1 Network Architecture	14
3.2.2 Structures of Neural Network Data.....	22
3.2.4 Dynamic Neural Network Concept and NARX	26
3.3 Model Predictive Control, MPC.....	28
3.3.1 MPC Introduction.....	28
3.3.2 Controller Horizon	31
3.3.3 Cost Function	31
3.3.4 Optimizer.....	33

CHAPTER 4: METHODOLOGY	35
4.1 Belt Drive System with MBD	35
4.2 Plant Variables	44
4.2.1 Pre-Tension	46
4.2.2 Driver Velocity.....	48
4.2.3 Tensioner Force.....	48
4.3 Co- Simulation	49
4.4 Case Studies and Velocity Profies	52
4.4.1 Case Studies	52
4.4.2 Velocity Profiles.....	54
4.5 Building Artificial Neural Network for Plant Model	58
CHAPTER 5: RESULTS and DISCUSSION	65
5.1 Belt Drive System Behavior under Different Driver Velocity Inputs	65
5.1.1 Impulse Input.....	65
5.1.2 Ramp Input.....	71
5.1.3 Sinusoidal Input.....	77
5.1.4 Step Input	83
5.2 Effect of Sensor Numbers on the Belt Drive System Control.....	90
5.3 Controller Comparison.....	96
5.3.1 Parametric Study on MPC.....	96
5.3.2 Parametric Study on PID.....	103
5.3.3 MPC and PID Comparison.....	106
5.4 Case Studies	108
5.1.1 Elementary Level Case Studies.....	110
5.1.2 Intermediate Level Case Studies	118
5.1.3 Advanced Level Case Studies	124
CHAPTER 6: CONCLUSION.....	133
REFERENCES.....	134
APPENDIX:.....	135

LIST OF TABLES

	Page
Table 4.1 Beam belt properties.....	40
Table 4.2 Material properties of the belt	41
Table 4.3 Geometric and material properties of pulleys	41
Table 4.4 Geometric and material properties of flanges	42
Table 4.5 Position of pulleys and flanges for prototype model.....	43
Table 4.6 Position of pulleys and flanges for four pulley system	43
Table 4.7 Contact properties of pulleys and flanges	44
Table 4.8 Properties of connecting force.....	44
Table 4.9 Properties of artificial neural network used for the belt drive system identification.....	59
Table 5.1 Parametric study for mpc design	97
Table 5.2 Effects of control horizon on mpc reference line tracking	101
Table 5.3 Effects of prediction horizon on mpc reference line tracking .	101
Table 5.4 Tuned parameters for pid controller design.....	103
Table 5.5 Comparison of controllers	108
Table 5.6 Case studies	109

LIST OF FIGURES

	Page
Figure 2.1 An example for manual tensioners	5
Figure 2.2 A semi- automatic belt tensioner with double eccentric	6
Figure 2.3 Basic auto tensioner parts	6
Figure 3.1 Coordinate system of the belt body	8
Figure 3.2 Nodal mass and moment of inertia	9
Figure 3.3 Dimension of a pulley	10
Figure 3.4 Dimension of a flange	10
Figure 3.5 Normal and friction force of contact	12
Figure 3.6 Relationship between μ and v	13
Figure 3.7 Sensing point of distance sensor	13
Figure 3.8 Schematic of neural network training process	15
Figure 3.9 Schematic of simple neurons	16
Figure 3.10 Linear and Log-sigmoid transfer functions	17
Figure 3.11 Neuron with vector input	17
Figure 3.12 Neuron with vector input in abbreviated notation	18
Figure 3.13 Single layer	18
Figure 3.14 Single layer in abbreviated notation	20
Figure 3.15 Multiple layer	20
Figure 3.16 Multiple layer in abbreviated notation	21
Figure 3.17 Static network	22
Figure 3.18 Dynamic network with a delay input	23
Figure 3.20 Block diagram of NARX	27
Figure 3.21 NARX architecture with two layer	27
Figure 3.22 Tapped delay line diagram	28
Figure 3.23 MPC block diagram	29

Figure 3.24 MPC working principle.....	30
Figure 3.25 MPC block diagram	34
Figure 4.1 The belt-pulley system used as prototype in the thesis	35
Figure 4.3. The belt system used in the simulation	37
Figure 4.4 Step input used for the determination of beam element number	37
Figure 4.5 Belt Deflections for the belt models with 50, 75,90,105,120,135,150,175,200,225,250,275,300 elements	38
Figure 4.6. Convergence analysis for belt segment.....	39
Figure 4.7. The belt-pulley system used in the intermediate level case study.....	40
Figure 4.8 Serpentine belt system components	45
Figure 4.9 A screenshot from the simulation used for pre- tension determination	47
Figure 4.10. A chirp wave	48
Figure 4.11 A sine- stream wave	49
Figure 4.12 Co-simulation fundamentals	50
Figure 4.13 Co-simulation steps	51
Figure 4.14 Elementary level configuration	52
Figure 4.15 The belt-pulley system used in the intermediate level case study.....	53
Figure 4.16 Driver pulley for advanced case.....	53
Figure 4.17 Driver velocity for case 1	54
Figure 4.18 Driver velocity for case 2	54
Figure 4.19 Driver velocity for case 3	55
Figure 4.20 Driver velocity for case 4	55
Figure 4.21 Full-car model block diagram in simulink	56
Figure 4.22 Detailed view of full-car model block diagram in simulink ..	57

Figure 4.23 Narx structure for 2 hidden layers.....	58
Figure 4.24 Hidden layer transfer function	60
Figure 4.25 Output layer transfer function	60
Figure 4.26 Artificial neural network used in this thesis.....	61
Figure 4.27 Graphical diagram representation of neural network.....	62
Figure 4.29 Co-simulation block diagram.....	64
Figure 5.1 The training inputs for impulse driver velocity.....	65
Figure 5.2. Narx responses for the impulse training inputs.....	66
Figure 5.3. Narx performance for the impulse training inputs	66
Figure 5.4. Narx regression for the impulse training inputs	67
Figure 5.5 Narx confirmation for the extended training inputs	68
Figure 5.6 Narx model confirmation trained with impulse input in case of no tensioner force is applied	69
Figure 5.7 Narx model confirmation trained with impulse input for different driver velocity in case of no tensioner force is applied	70
Figure 5.8 The training inputs for ramp driver velocity	71
Figure 5.9 Narx responses for the ramp training inputs	72
Figure 5.10 Narx performance results for the ramp training inputs	72
Figure 5.11. Narx regression results for the ramp training inputs	73
Figure 5.12 Narx confirmation for the extended training inputs	74
Figure 5.13 Narx model confirmation trained with ramp input in case of no tensioner force is applied	75
Figure 5.14 Narx model confirmation trained with ramp input for different driver velocity in case of no tensioner force is applied	76
Figure 5.16 Narx responses for the sinusoidal training inputs	78
Figure 5.17 Narx performance for the sinusoidal training inputs.....	78
Figure 5.18 Narx regression for the sinusoidal training inputs	79
Figure 5.19 Narx confirmation for the extended training inputs.....	80

Figure 5.20 Narx model confirmation trained with sinusoidal input in case of no tensioner force is applied.....	81
Figure 5.21 Narx model confirmation trained with sinusoidal input for different driver velocity in case of no tensioner force is applied	82
Figure 5.23 Narx responses for the step training inputs	84
Figure 5.24 Narx performance for the step training inputs	84
Figure 5.25 Narx regression for the step training inputs	85
Figure 5.26 Narx confirmation for the extended training inputs	86
Figure 5.27 Narx model confirmation trained with step input in case of no tensioner force is applied.....	87
Figure 5.28 Narx model confirmation trained with step input for different driver velocity in case of no tensioner force is applied	88
Figure 5.29 Mpc results for impulse, ramp, sinusoidal and step inputs	89
Figure 5.31 Unitless comparison of measured points.....	91
Figure 5.32 Simulink diagram for comparison of model predictive controllers	93
Figure 5.33 Comparison between multiple and single sensor cases.....	94
Figure 5.34 Figure 5.34 Detailed view of comparison for different number of sensor use	95
Figure 5.35 Driver Velocity Input with Respect to Time Used For Controller Comparison	98
Figure 5.36 Sensor reading in case of no tensioner force.....	98
Figure 5.37 Comparison of model predictive controllers.....	99
Figure 5.38 Mpc controller' s reference line tracking performance for different control horizons	100
Figure 5.39 Comparison of Mpc for step reference trajectory	102
Figure 5.40 Html report screen prepared for kd effect.....	105
Figure 5.41 PID block used for comparison.....	106

Figure 5.42 MPC vs PID controller.....	107
Figure 5.45 Regression Plot of the System for Elementary Level Case Studies	111
Figure 5.52 Neural network response for intermediate level case studies	118
Figure 5.55 Test data 1 for intermediate level case studies.....	120
Figure 5.56 Test data 2 for intermediate level case studies.....	121
Figure 5.61 Figure 5.61 Performance values of the system for advanced level case studies	127
.....	128
Figure 5.62 Test data 1 for advanced level case studies.....	128
Figure 5.63 Test data 2 for advanced level case studies.....	129
Figure 5.64 Test data 3 for advanced level case studies.....	131
Figure 5.65 Test data 4 for advanced level case studies.....	132
Figure 5.66 Test Data 4 for Advanced Level Case Studies.....	132

SYMBOLS AND ABBREVIATIONS

Symbols

ρ	Density
L	Belt Element Length
A	Cross Sectional Area
I_{xx}	The Area Moment of Inertia
W	Belt Width
T	Belt Thickness
k	The Spring Coefficients
c	The Damping Coefficients
δ	Penetration Amount
$\dot{\delta}$	Time Differentiation of the Penetration
m_1	The Non-Linear Contact Force Exponent for Stiffness
m_2	The Non-Linear Contact Force Exponent for Damping
m_3	Indentation Damping Effect Exponent
r_s	The Position of Sensor Center

r_p	The Position of the Closest Point Of Body on the Direction and in the Range of the Sensor
d_{min}	The Minimum Distance from Sensor Center to Mid- Plane
θ	The Angle Between the Direction of d_{min} and the Direction of the Output Desired
p	Scalar Input
w	Scalar Weight
b	Bias
n	Output of Sum
f	Activation Function
a	Scalar Neuron Output
R	Number of Individual Inputs
$w_{l,l}$	Weights
$IW^{i,j}$	Input Weights
$LW^{i,j}$	Layer Weights
Q	Number of Sequences
$p_i (i=1,2, \dots, Q)$	Neural Network Input

TS	Number of Timesteps Used for the Input
F	Mean Square Error
N	Number of Target Values
t_i	Target Values or Outputs
e_i	Corresponding Errors for Each Network Output and Target Values
H	Hessian Matrix
μ	Scalar
g	Gradient
I	Identity Matrix
J	Jacobian Matrix
$J_y(z_k)$	Output Reference Tracking Term
$J_u(z_k)$	Manipulated Variable Tracking Term
$J_{\Delta u}(z_k)$	Manipulated Variable Move Suppression Term
$J_q(z_k)$	Constraint Violation Term
k	Current Control Interval
p	Prediction Horizon

n_y	Number of Plant Output Variables
$y_j(k + i k)$	Predicted Value of jth Plant Output at ith Prediction Horizon Step
$r_j(k + i k)$	Reference Value for jth Plant Output at ith Prediction Horizon Step
s_j^y	Scale Factor for jth Plant Output
$w_{i,j}^y$	Tuning Weight for jth Plant Output at ith Prediction Horizon Step
n_u	Number of Manipulated Variables
s_j^u	Scale Factor for jth Manipulated Variable
$w_{i,j}^u$	Tuning Weight for jth Manipulated Variable at ith Prediction Horizon Step
$w_{i,j}^{\Delta u}$	Tuning Weight for jth Manipulated Variable Movement at ith Prediction Horizon Step
ϵ_k	Slack Variable at Control Interval K
ρ_ϵ	Constraint Violation Penalty Weight

Abbreviations

MPC	Model Predictive Control
-----	--------------------------

MBD	Multibody Dynamic
EOM	Equation of Motion
DAE	Differential Algebraic Equations
BDF	Backward Differentiation Formula
DOF	Degree of Freedom
ODEs	Ordinary Differential Equations
CNF	Contact Normal Force
CFF	Contact Friction Force
TMR	Theoretical Manual of Recurdyn
ANN	Artificial Neural Network
PID	Proportional- Integral- Derivative
MSE	Mean Square Error
NARX	Nonlinear Autoregressive Network with Exogenous Inputs
LMPC	Linear Model Predictive Controller
NMPC	Non- Linear Model Predictive Controller
compet	Competitive Transfer Function
elliotsig	Elliot Sigmoid Transfer Function
hardlim	Positive Hard Limit Transfer Function

hardlims	Symmetric Hard Limit Transfer Function
logsig	Logarithmic Sigmoid Transfer Function
netinv	Inverse Transfer Function.
poslin	Positive Linear Transfer Function
purelin	Linear Transfer Function
radbas	Radial Basis Transfer Function
radbasn	Radial Basis Normalized Transfer Function
satlin	Positive Saturating Linear Transfer Function
satlins	Symmetric Saturating Linear Transfer Function
softmax	Soft Max Transfer Function
tansig	Symmetric Sigmoid Transfer Function
tribas	Triangular Basis Transfer Function
TDL	Tapped Delay Line

CHAPTER 1: INTRODUCTION

1.1 Background

The means of power transmission in automotive industry are the belt drive systems. As the most popular belt drive system, the serpentine belt drive system requires a well design for the control of the belt transverse vibration due to the undesired consequences such as decrease in the performance of the accessories and the early failure of the drive system parts. These consequences has motivated automotive industry to consider tensioners in their design.

There are two main classifications of automotive belt drive tensioners, namely, passive and active belt tensioners. If the belt tension is adjusted with the help of purely mechanical power, it is called as passive one. However, if electronic actuations are used, it is called as active one or automatic tensioners.

1.2 Motivation

Although there are variety of belt tensioner models, the tendency to automatic tensioners are increasing. Nevertheless, the development of the automatic belt tensioners are restricted to the specific regions due to the mass production of the belt tensioner makers. It is possible to find a better solutions from pre-existing ones to decrease the transverse vibration of the belt spans. In addition to the producer's behaviour, studies in belt drive system area are mainly around understanding of the belt behaviour rather than control of the vibration. Although there are small amount of studies related to vibration control of the belt drive system for which main interest is on longitudinal vibration rather than transverse vibration, no control system includes model prediction. This idea is the motivation to use Model Predictive Control (MPC) in this study.

1.3 Thesis Objectives and Scope of Research

The objective of this study is to control the transverse vibration of a serpentine belt system which includes the fundamental parts of a belt system. The fundamental parts mentioned here are one driven, one driver, one tensioner pulleys and belt. In order to use MPC, either a linear mathematical model should be found or a model should be constructed

with the help of training data and this model should be linearized later. The second approach is preferred in this study. Then, results with and without MPC control are compared.

1.4 Organization and Content of Thesis

Transverse vibration control of a belt drive pulley system in this study is explained in six chapters. General idea about the thesis structure is given in introduction. A brief literature review is presented in chapter 2. The basic theories regarding the MBD used in this study, neural networks, and MPC are explained in chapter 3. The most important part of this study, methods used for this thesis, are described in chapter 4. This chapter includes description of the system components, building an MBDS, controlled parameters and inputs decision, data collection, building neural network for plant model, MPC design with NN approximation, use of MPC with MBD. Chapter 5 includes results and related discussions. Finally, the work presented in this study is summarized and future works are discussed in chapter 6.

CHAPTER 2: LITERATURE REVIEW

There are several study concepts regarding the belt such as axially moving materials, steady state belt- pulley mechanics, serpentine belt drive studies which deals with rotational vibration of pulleys by assuming belt as linking springs or transverse vibration of individual spans without considering the pulley effect or coupled vibration for limited range and situations [1]. The method used in this study is to employ a multi-body dynamic simulations to model the belt drive and design a predictive controller for this model. Thus, the literature reviews related to abovementioned concepts are not given and the classification of this chapter is made according to the general market information.

2.1 Introduction

Power is transmitted from a driver shaft to one or more driven shaft by means of mechanical elements such as belts, ropes, chains and gears. Although gears are good for small distances, they cannot be used for long distances because it makes the drive system bulky which yields increase in weight and cost. From the perspective of the strength and slippage, the rope cannot be used to transmit torque in driving the accessory pulleys on the engine. Furthermore, since the cost and the comfort are as important factors as the performance for today's consumers, the chain is not preferred as much as belt. The chain is expensive as although there is almost no need for replacement cost, it is costly in terms of its production and the material used for it. The chain is not comfortable as belts because it works more loudly and it does not transmit the torque as smooth as belt. Although there are advantages of the belt drive system, the excessive transverse vibration of this system is an open question. To solve this issue, transverse vibration of a commonly used belt driving system, serpentine belt system, is considered in this thesis and the literature review depends on this belt system.

2.2 Serpentine Belt System

The advantages of using belt over the other power transmission systems are mentioned in the introduction. However, the main focus of this thesis is on serpentine type belts. The reasons why it is considered relies on the priority of this type belt system over the conventional multiple belt systems in terms of consuming less space in the engine, reduced slip, easier replacement and maintenance, increase in power transmission and decrease in cost due to the previous advantages. The serpentine type belt systems consist of four main

parts which are driver pulley, driven pulley, tensioner and belt. The number of driven pulleys depend on the number of the accessories. The idler pulley term may be used as fixed tensioner or the assistive pulley for tensioning other than active tensioner in the literature. Several advantages of this belt system is mentioned up to this point; however, it has a lifespan like. Since this type of belt is used commonly in the market, increase in life of its system is crucial. There are many reasons for the failure. The reasons may be misalignment, belt pulley contact, belt slip and so on. One of the most important challenge to increase the life of the system is controlling the transverse vibration of the belt which also cause the noise. In order to eliminate this problem, many tensioner types have been proposed and they are explained in this section.

2.2.1 Tensioner Types

Belt tension is the key to proper functioning of the accessory belt drive system. The tensioner ensures the sufficient tension to power the accessories. If the tension is less than the desired level, this situation causes belt slip which results in noise. Furthermore, excessive slip cause excessive heat for accessories, especially on accessory bearings. Thus, accessory performance reduces. Moreover, excessive heat reduces the performance and the life of the belt. On the other hand, if the level of the tension is more than enough, the bearings starts to bend which results in the early failure of both bearings and their corresponding accessories. In some cases, the result can reach dangerous level at which belt becomes free of contact with the pulleys. In this case, the functions of the accessories are lost. The loss of contact between the belt and the accessories like the water pump, power steering pump, and alternator not only cease their functioning but also the entire system because the vehicle becomes quickly unusable due to loss of engine cooling[2].

In order to solve the tension adjustment problem, several tensioner types are proposed. The tensioners can be classified into three main categories which are manual, semi-automatic and automatic tensioners.

2.2.1.1 Manual Tensioners

Manual tensioners can be thought as a primitive version of the tensioners. A typical manual tensioner can be seen from Figure 1.

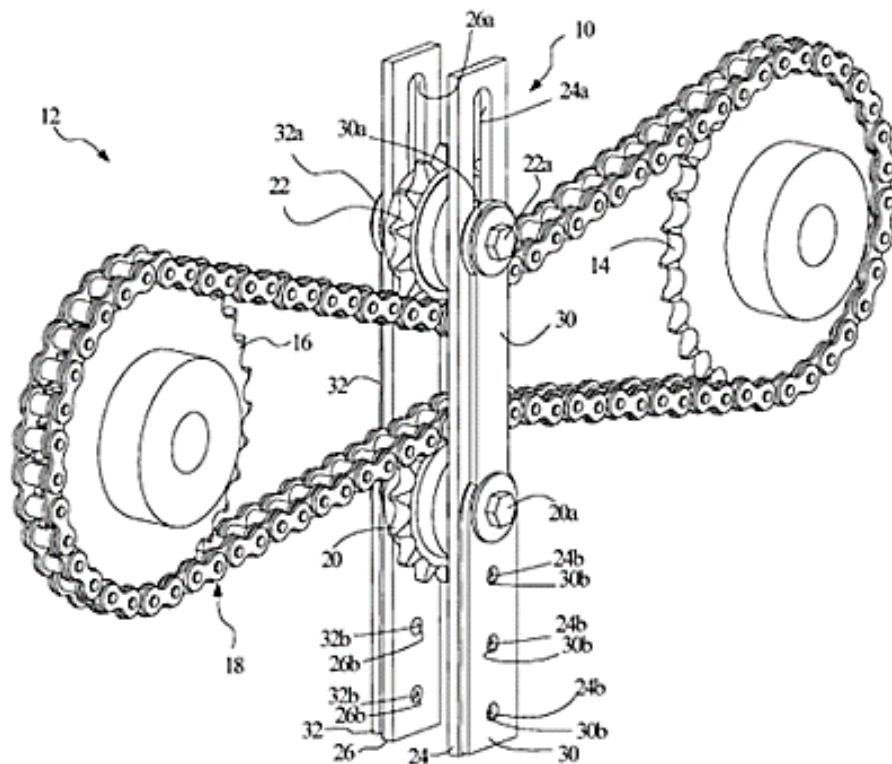


Figure 2.1 An example for manual tensioners [3]

As it can be seen from the figure, there is a slot inside the tensioner block. After adjusting the position of the tensioner arm by moving it in slot manually, the tensioner position can be fixed by means of mechanical joints like bolts.

The advantage of using manual tensioner is its compact design, its being cheap and easily producible. However, since belt motion does not follow a simple way and fluctuation is not that much simple using manual tensioner does not seem to be a good idea. In addition to this, belt tension should be adjusted manually.

2.2.1.2 Semi- Automatic Tensioners

In this type of tensioners, springs are used to compensate for belt elongation. This is achieved by predefined spring force adjusted at ambient temperature.

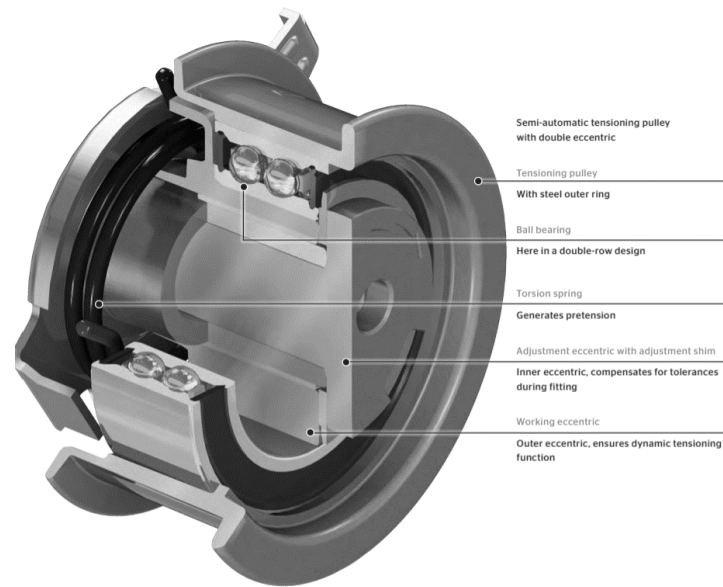


Figure 2.2 A semi- automatic belt tensioner with double eccentric [4]

Although temperature fluctuations, load changes and belt elongation are somewhat compensated, it is compulsory to adjust tension manually.

2.2.1.3 Automatic Tensioners

Automatic tensioners have a set of springs and due this set, it provides an additional integral mechanical damping function. It is similar to semi- automatic tensioners but it does not need manual adjustment. This set provides almost constant belt tension by self-adjusting to load and temperature changes.

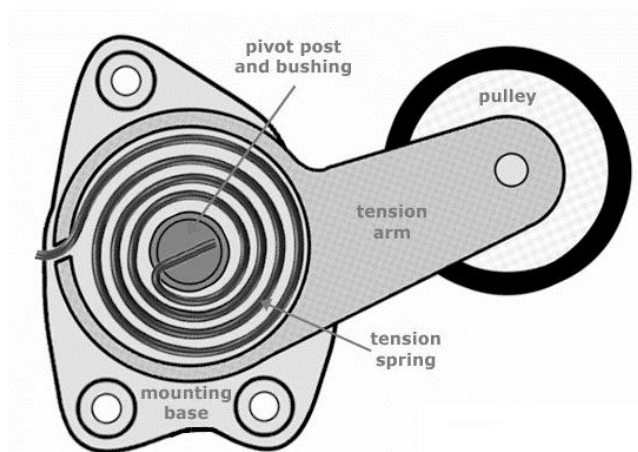


Figure 2.3 Basic auto tensioner parts [5]

CHAPTER 3: THEORY

The theory of the thesis is divided into three main categories. Firstly, the general information related to the MBD used in this study is given. Secondly, artificial neural networks are introduced and a fundamental summary of MPC is mentioned finally.

3.1 Multibody Dynamic

If the area of the study in the system analysis is to know the position, velocity and acceleration of a part in a mechanical design, it is the case for dynamic analysis. Dynamic analysis enable users to evaluate the abovementioned kinematic properties of a mechanical system under the effect of dynamic loads such as translational forces. The dynamic analysis can be done by rigid and flexible elements. Rigidity is an assumption which says the rigid bodies cannot be deformed. If the interest of the analysis is to see the static analysis effects like deformation, buckling effect, fatigue, bending, twisting, flexibility in the dynamic analysis; flexible or deformable elements can be used [6].

As the flexible elements increase in a dynamic analyses, there can be several consequences such as increase in CPU consumptions, extra memory usage and long run times due to the need for extra calculations for each time step. As the complexity of the system increases, the consequences will be more significant. In addition to the complexity, if the system works in a transient region, simulation's being slow will be at the top limits. In order to overcome this issue, there are two methods. The first method says that faster computers, more processors and parallel solvers should be preferred; however, this method cannot be the solution for every time. The second method suggests that if the main focus is not to get the values gathered as a result of deformation like stress- strain, it is better to assume the bodies as rigid bodies [6].

The common method used in the industry is to use a hybrid model which is combination of both rigid and flexible elements. The belt drive system used in this study composed of pulleys, flanges and belt. Whereas belt is modeled with flexible beam elements, the other components are modeled as rigid. The beam element used in the simulation program is named as Beam2 Element under the title of flexible finite elements.

3.1.1 Belt Drive Model

The simulation system composed of the all fundamentals parts of a serpentine belt system. It includes a driver pulley, a driven pulley (*accessory pulley*), a tensioner and the belt. The idler pulley is not the fundamental part of the serpentine belt system. Thus, it does not exists in our simulation system. The increase in the number of pulley means an increase in the length of the belt and calculation procedure which result in longer simulation time and nonessential cpu consumption for this thesis.

3.1.1.1 Belt

The belt is composed of number of beam elements which are connected with a beam force. The advantage of using beam element is its 6 degree of freedom which represents a belt movement better than truss.

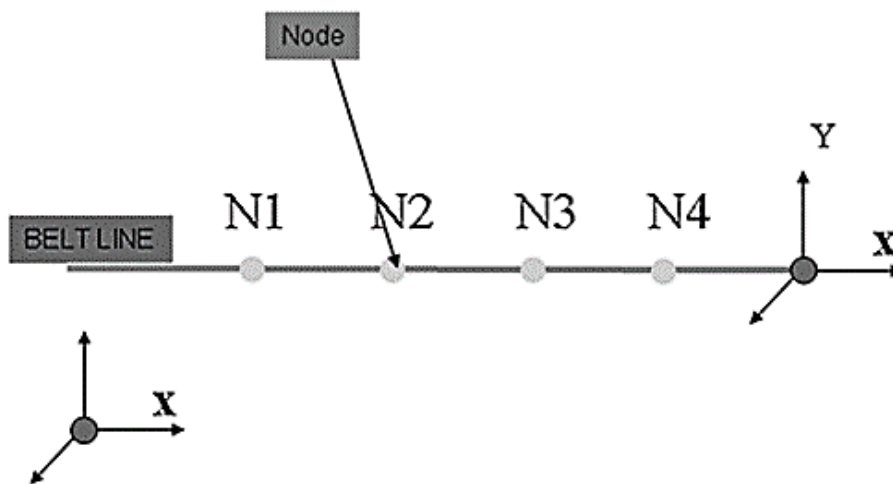


Figure 3.1 Coordinate system of the belt body

Belt is composed of several belt bodies and these bodies are connected at the nodes. As can be seen from the figure, the orientation of center marker of node depends on the direction used. The positive initial velocity triggers the belt in positive x- direction for the given figure above.

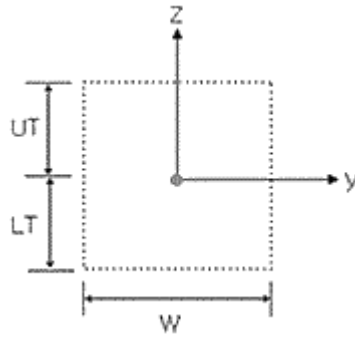


Figure 3.2 Nodal mass and moment of inertia

The terms related to nodal masses and moments of inertia are explained below.

The mass formula can be written as follows

$$m = \rho LA \quad (1)$$

and moment of inertia formula is as given below

$$I_{xx} = I_{yy} + I_{zz} \quad (2)$$

$$I_{yy} = \frac{WT^3}{12} \quad (3)$$

$$I_{zz} = \frac{WT^3}{12} \quad (4)$$

where,

ρ is density

L is belt element length

A is cross sectional area

I_{xx} is the area moment of inertia

W is belt width

T is belt thickness

3.1.1.2 Pulleys and Flanges

All the pulleys whether it is a driver, driven, or tensioner pulley are represented by a roller. Simply, it is a cylinder which is contacted to the bottom contact nodes or the top contact nodes of the belt depending upon whether it is located the inside or the outside of belt assembly loop.

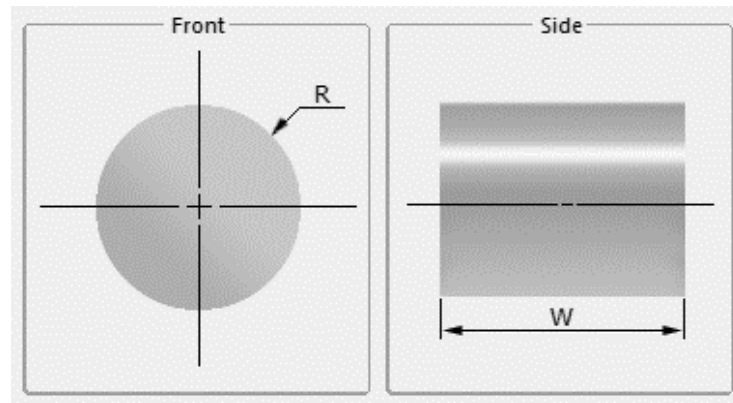


Figure 3.3 Dimension of a pulley [7]

Flanges can be thought as a kind of lid for the rollers. They restrict the motion of the belt so that in case of belt movement over a roller, belt do not lose the contact with pulley. This is achieved by contact between the sloped surface of flange and belt contact nodes. Depending on the configuration, flange can be in contact with lower side contact nodes or upper side contact nodes of the belt.

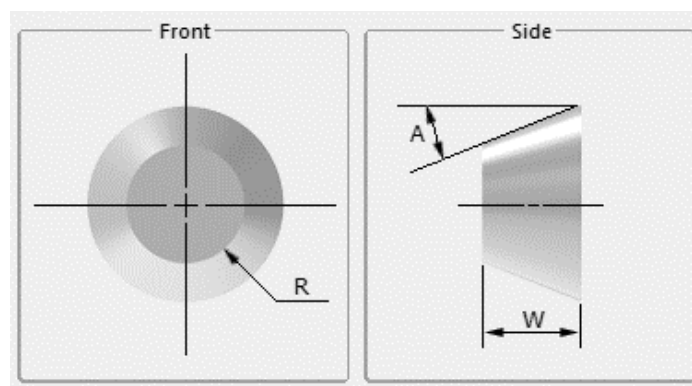


Figure 3.4 Dimension of a flange [7]

3.1.2 Contact Forces

In order to understand how the contact between the belt segment and pulleys are established, the contact force and the formula has to be addressed.

Owing to the interaction between belt system and solid bodies (*pulleys, flanges*), there are large contact forces. These forces are calculated with the contact parameters. The explanations of the parameters used in RecurDyn are given below [7].

The contact normal force can be written as

$$f_{nn} = k\delta^{m_1} + c \frac{\dot{\delta}}{\delta} \left| \dot{\delta} \right|^{m_2} \delta^{m_3} \quad (5)$$

where

k is the spring coefficients

c is the damping coefficients

δ is the penetration

$\dot{\delta}$ is time differentiation of the penetration

m_1 is the non-linear contact force exponent for stiffness

m_2 is the non-linear contact force exponent for damping

m_3 indentation damping effect exponent

In order to overcome the unrealistic case (*negative contact force*) which occurs in case of small penetration, m_3 is greater than 1.

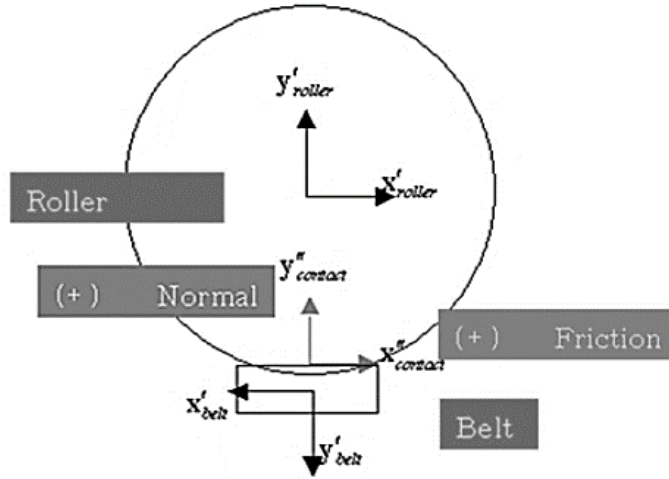


Figure 3.5 Normal and friction force of contact [7]

Since contact normal force is known, the friction force can be written as follows

$$f_f = \mu(v) \cdot |f_n| \quad (6)$$

$$\mu(v) = \begin{cases} \text{step}(v, 0, 0, v_s, \mu_s) \cdot \text{sign}(-v) & |v| < v_s \\ \text{step}(v, v_s, \mu_s, v_d, \mu_d) \cdot \text{sign}(-v) & |v| \geq v_s \end{cases} \quad (7)$$

where

f_n is the contact normal force

μ is the friction coefficient

- μ_s is the static friction coefficient
- μ_d is the dynamic friction coefficient

v_s is the static threshold velocity

v_d is the dynamic threshold velocity

The relationship between tangential velocity of v and the friction coefficient of μ is revealed in figure 3.6

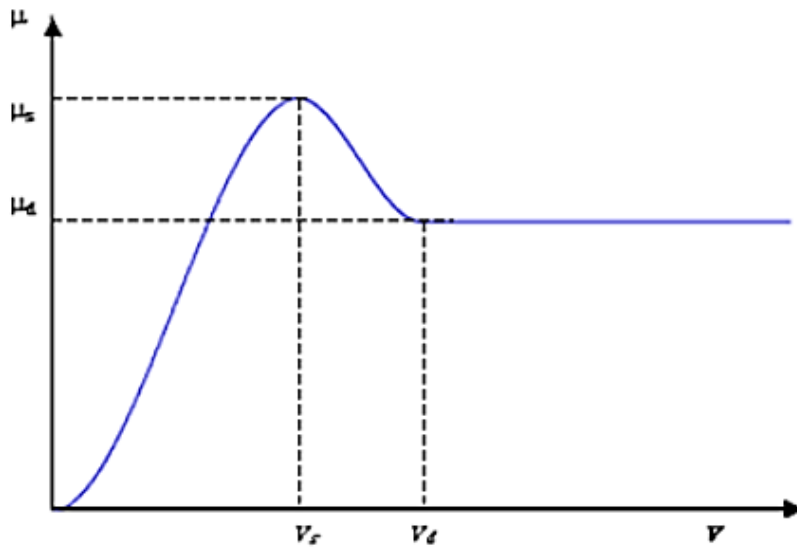


Figure 3.6 Relationship between μ and v [7]

3.1.3 Distance Measurement

In order to control a belt system actively, it is a must to collect the data from the system in order to know the state of the system. Generally, ultrasonic or laser sensors are used to collect data from the belt spans in real life. To simulate sensor behavior and collect the data needed, sensor tab is available in Recurdyn.

The center position, direction and the range of the sensor are given. Since the transverse vibration of the belt is interested in this thesis, the position of the sensor is adjusted so that its position in x-direction is at the middle of the longest span and the direction.

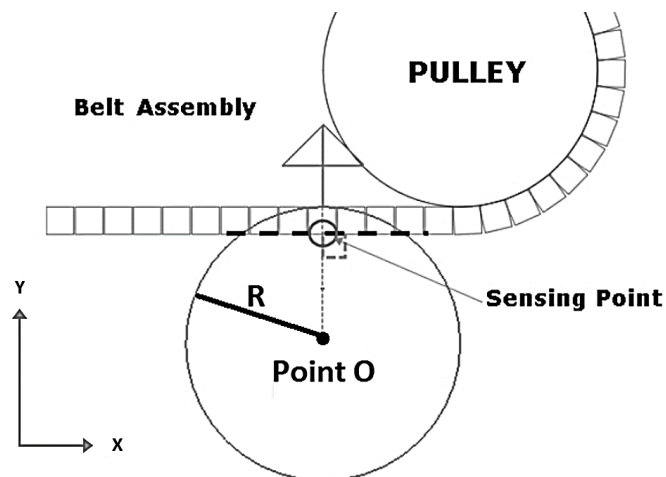


Figure 3.7 Sensing point of distance sensor

How a distance sensor works for a sheet body is explained in Recurdyn Help as mentioned below:

- The closest point in the range is found
- The minimum distance from sensor center to mid plane is calculated

$$d_{\min} = |r_s - r_p| \quad (8)$$

where

r_s is the position of sensor center

r_p is the position of the closest point of body on the direction and in the range of the sensor

- The angle, θ , between the direction of d_{\min} and the direction of the output desired is determined as

$$\theta = \cos^{-1} \left(g_s \cdot \frac{r_s - r_p}{|r_s - r_p|} \right) \quad (9)$$

- Finally, the output of the sensor is found as follows

$$d_o = \frac{d_{\min}}{|\cos \theta|} \quad (10)$$

3.2. Introduction to Artificial Neural Networks

The name Neural Network comes from the inspiration of brain cells which are called as neurons. Basically, dendrites, a cell body and the axon are the elements of a neuron. Electrical signals comes to dendrites and these elements carry the signals to the cell body. After signal processing in the cell body, the signal is sent to the other neuron's dendrites with the help of axon. Neuron arrangement (*interconnections*) and synapses, contact points between neurons, determines the networks' function. Similar to the natural or biological neural networks, artificial neural networks use parallel structure arrangement and transfer functions to understand the relations between inputs and outputs.

3.2.1 Network Architecture

There are two main concepts in Artificial Neural Networks which is base for the other concepts. These two basic concepts are cells (neurons) and architecture. The main idea of

Neural Networks is weight adjustment in the network by comparing the output of the network for the collected input and the target. This can be understood better from the below figure.

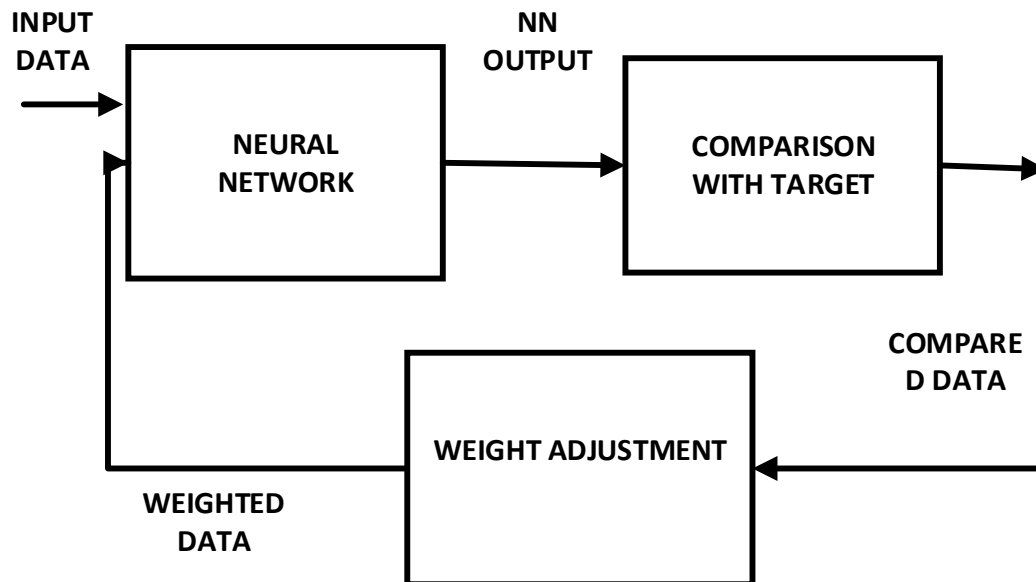


Figure 3.8 Schematic of neural network training process

3.2.1.1 Single Neuron

Though a Neural Network system may consist of several interconnections among the number of cells, their theory can be understood with the help of a simple neuron. The best way to address the neuron concept is to use figures. Below figure shows simple neurons without and with an offset. Generally, the offset is called as bias in Neural Networks. The input to the bias part is given as 1 for the simplicity. Using bias depends on the system behavior; that is, it is not a must for Neural Networks.

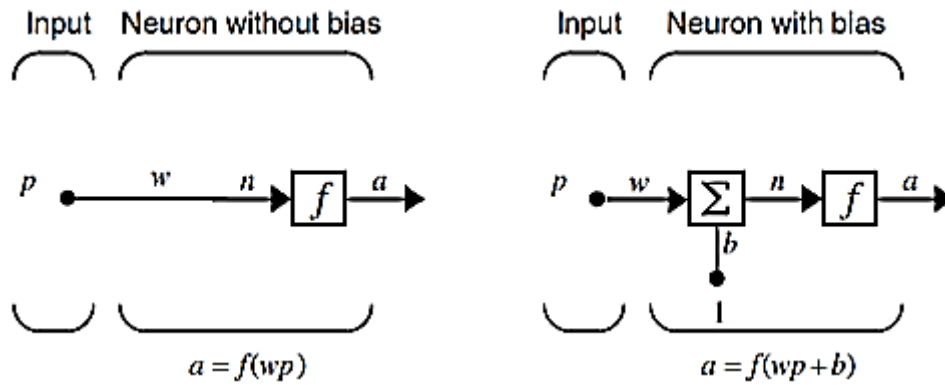


Figure 3.9 Schematic of simple neurons [9]

Explanations of the symbols of the above figure are as follows:

- p scalar input
- w scalar weight
- b bias or offset
- n output of sum or pw simply
- f transfer or activation function
- a scalar neuron output

As can be seen from the figure, there are two parameters to change. These parameters, weight and bias, are used to get desired value from the network.

3.2.1.2 Transfer Functions

The relation between sum of inputs n and network output a is defined with the help of transfer functions. There are several transfer functions used in neural networks. Linear Transfer Function (purelin) and Logarithmic sigmoid transfer function (logsig) are the most commonly used ones among the others. Thus, they will be explained briefly.

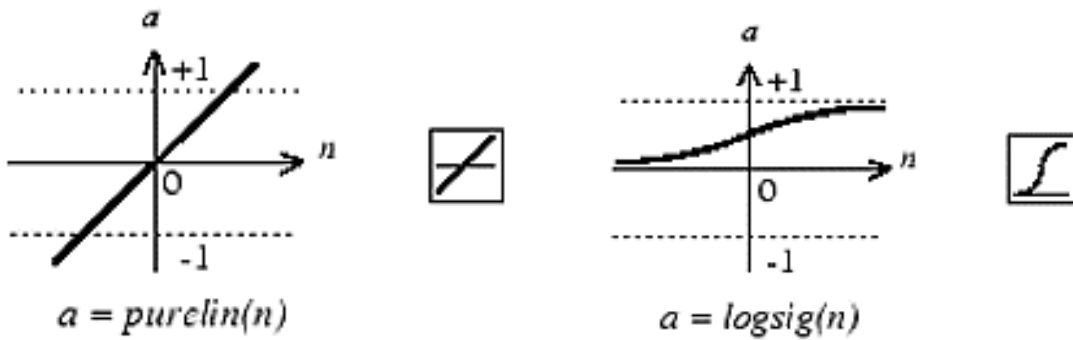


Figure 3.10 Linear and Log-sigmoid transfer functions [9]

Purelin is used for linear approximations. Log-sigmoid transfer function gives output between 0 and 1. It is generally used for backpropagation due to its being differentiable.

The boxes near each functions are the representations only.

3.2.1.3 Neuron with Vector Input

The inputs given to the neurons are not necessarily a single scalar input. In case of multiple inputs vectors are used for the neuron.

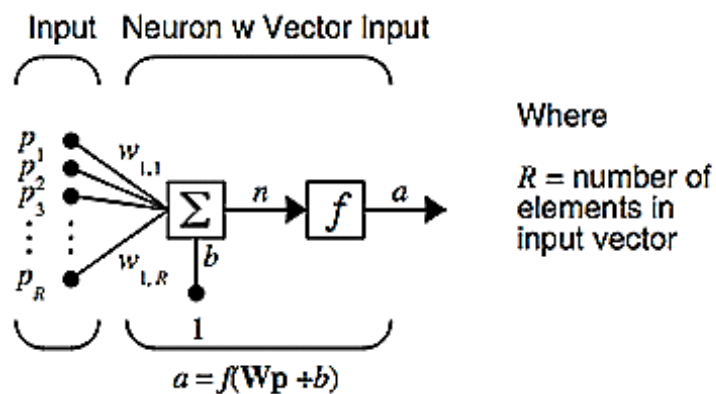


Figure 3.11 Neuron with vector input [9]

The symbols different from the previous symbols are

R number of individual inputs

$p_i (i=1,2,\dots,R)$ individual inputs

$w_{1,i}$ weights

As a result, the net input n can be described as

$$n = (w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R) + b \quad (11)$$

Another schematic demonstration of the above figure is done with abbreviated notation where matrix and vector representation is used and corresponding dimensions are given below the symbols and boxes.

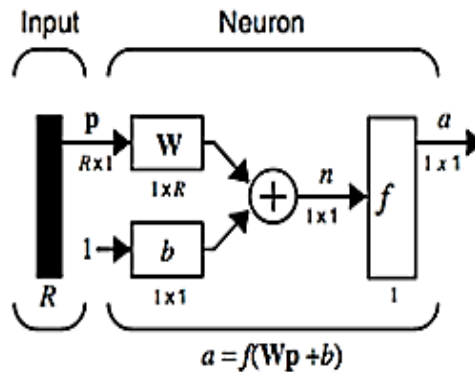


Figure 3.12 Neuron with vector input in abbreviated notation [9]

This demonstration ease the understanding of the architecture.

3.2.1.4 Single Layer

The part except for the inputs in the above figures can be considered as layer. A layer can have more than one neurons. This situation can be shown with the below figure.

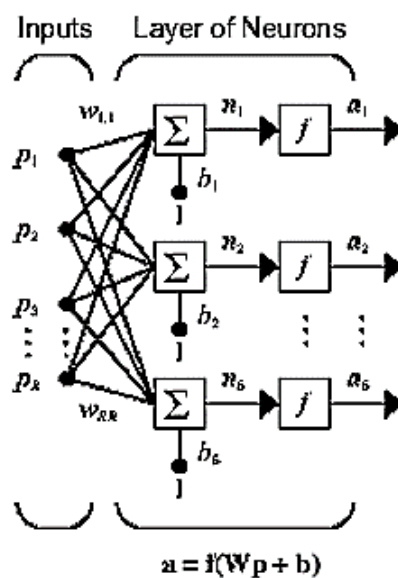


Figure 3.13 Single layer [9]

The terms used for the neural network is as follows:

R	number of elements in input vector
S	number of neurons in layer
$p_i (i=1,2, \dots R)$	individual inputs
p	input vector
$w_{ji} (j=1,2, \dots S)$	individual weights
W	weight matrix
$b_j (j=1,2, \dots S)$	bias for each neuron
b	bias vector
$n_j (j=1,2, \dots S)$	net input for each neuron
$a_j (j=1,2, \dots S)$	network outputs for each neuron
a	network output column vector
f	transfer function

The weight function in matrix form can be shown as

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad (12)$$

It should be noted that it is not necessary to put neither same number of neurons with inputs nor the same transfer functions for each neuron output.

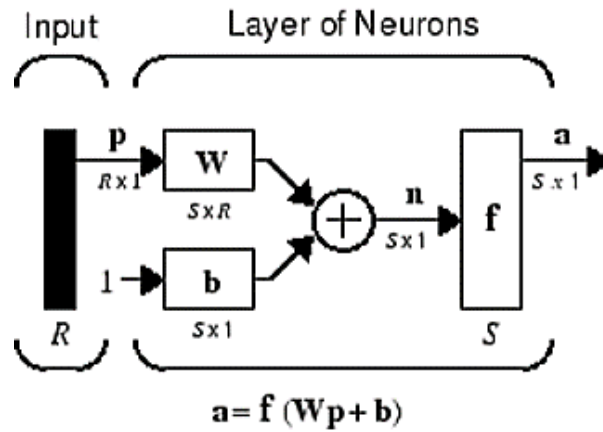


Figure 3.14 Single layer in abbreviated notation [9]

The above figure shows the same layer with abbreviated notation. Given matrix dimensions makes the drawing more clear.

3.2.1.5 Multiple Layers

As the name implies, more than one layer is used for this structure. Due to the existence of multiple layers, there is a need for weights between layers in addition to the weight between input and first layer sum.

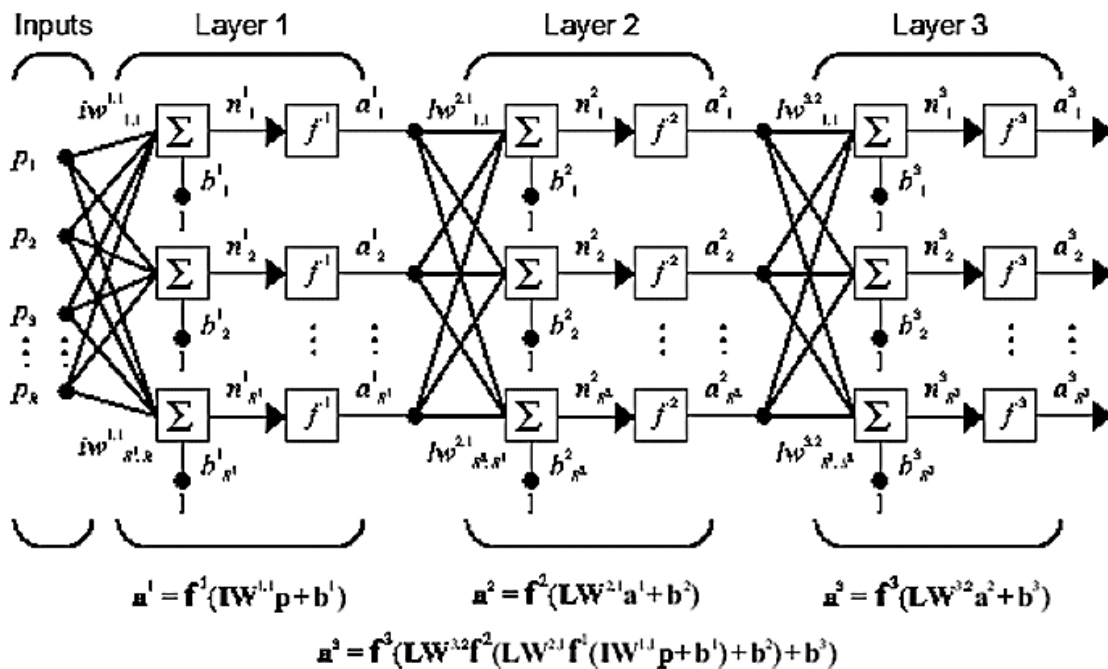


Figure 3.15 Multiple layer [9]

The abbreviated notation of the above structure can be shown as in below figure.

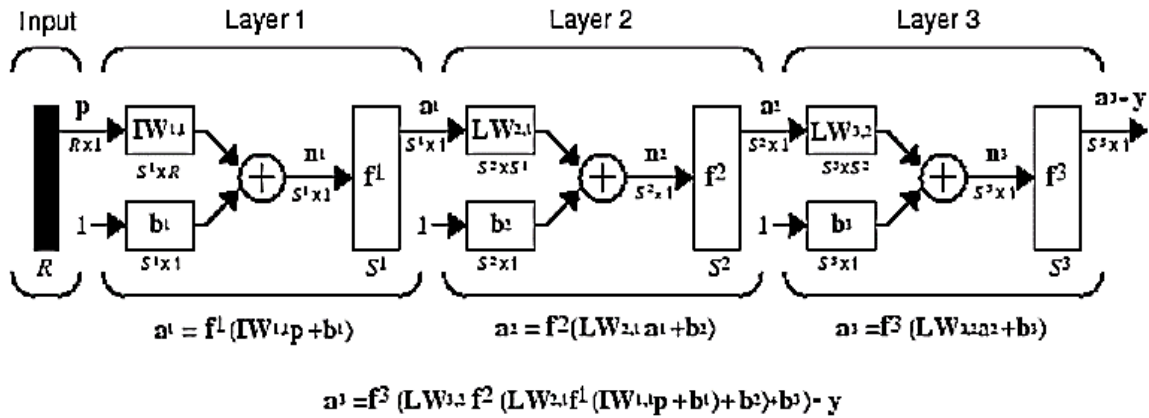


Figure 3.16 Multiple layer in abbreviated notation [9]

Net inputs, biases, weights and network outputs are shown with superscripts. Each number refers to the layers in order. As mentioned previously, the weights can be divided into two categories. They can be shown as

$IW^{i,j}$ *input weights*

$LW^{i,j}$ *layer weights*

where i shows the layer number and j refers to i-1 and j becomes i in case of i is equal to 1. That is the case for the first layer.

The crucial properties of the multiple layers are summarized below.

- Each layer may have different number of neurons.
- Output of a layer is an input for the next layer
- The first layer is called as Input Layer
- The last layer is called as Output Layer
- Layers other than the Input and Output Layer is called as Hidden Layer (*Input Layer can be count as a Hidden layer in the literature*)
- Any function can be approximated with finite number of discontinuities [10]

It should be also noted that the output of the last layer is the result of the network and it can be also shown with y . If the above multiple layer is considered, $y=a^3$.

3.2.2 Structures of Neural Network Data

There are 2 main type of networks,namely static and dynamic. In perspective of input data, there are two class of structure for the networks. The distinction between input data structures are done according to the importance of time sequence. Whereas the order is unimportant for concurrent inputs, it is crucial for sequential inputs.

3.2.2.1 Concurrent Inputs in a Static Network

If there is no feedback or delay in network which is called as static network, being sequential will be unimportant. That is why sequential inputs are not considered in this chapter. In other words, inputs can be thought as concurrent for this case. In fact, in case of sequential inputs for static network response will not alter but the way for training will change.

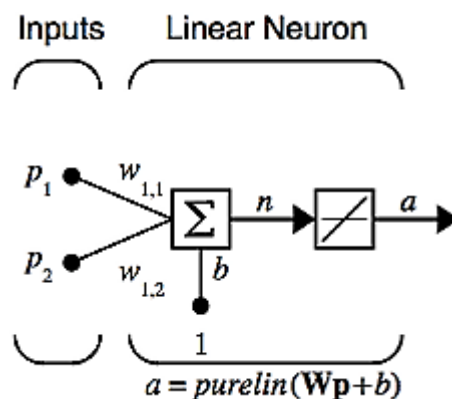


Figure 3.17 Static network [9]

As can be seen from the above figure, there is no interactions between the inputs and the output is also concurrent.

3.2.2.2 Sequential Inputs in a Dynamic Network

If there is a feedforward connection or a feedback, the network is called as dynamic network. This definition also classify the dynamic network as feedforward and recurrent dynamic network. Here, the term recurrent is used instead of feedback. If delay exist in the network, the case is a sequence of input usually.

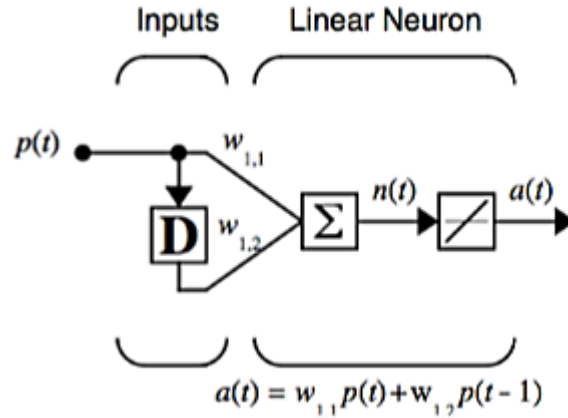


Figure 3.18 Dynamic network with a delay input [9]

An example to this case is given in the above figure in which there is only one delay for the sake of simplicity.

The inputs given to the network from the multibody dynamic simulation is divided into three categories as training, validation and test. The training data is used to teach the network how the system behave. Validation data is used to reach the peak of the generalization up to when training continues. Finally, how much the network works for different data sets is evaluated according to the test data.

3.2.2.3 Concurrent Inputs in a Dynamic Network

In some special cases, concurrent inputs can be used instead of sequence inputs for dynamic networks. If it is desired to see results for number of different sequences, concurrent sequences set can be used. These sequences behaves as individual sequences and does not affect the others.

3.2.3 Training

The simplest definition of the training is the adjustment of the network independent variables, weights and biases, so that network output is close enough to the target values. The measurement of closeness is determined by the difference between the target values and network outputs. That is, as the differences (errors) are small enough according to the algorithm used the convergence is accepted. Generally, mean square error is used for the neural networks and the formula of this error criteria is given below.

$$F = mse = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2 \quad (13)$$

where

F	mean square error
N	number of target values
$t_i (i=1,2,\dots,N)$	target values or outputs
$a_i (i=1,2,\dots,N)$	network output
$e_i (i=1,2,\dots,N)$	corresponding errors for each network output and target values

Although error information gives us the convergence for the system behavior, it is only a tool for training method which defines how to make updates for the independent network parameters and when to stop. There are several training algorithms; however, it is possible group them into two main branch which are incremental and batch mode. As their name implies, whereas all inputs send to the network before any update in batch training mode, computation of gradient and updates are done incrementally; that is, after each input in incremental training mode. Since making calculations after all inputs both shorten the processing time and gives a chance to evaluate the system behaviors from all inputs, batch mode does not only give faster results but also smaller errors [9].

The error can be calculated with the help of mean square error concept easily. However, in order to evaluate the performance depending on this error there are two ways of optimizations which are gradient and Jacobian method. There are many algorithms using gradient or Jacobian optimization methods like Levenberg-Marquardt, Bayesian Regularization, BFGS Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell/Beale Restarts, Fletcher-Powell Conjugate Gradient, Polak-Ribière Conjugate Gradient, One Step Secant, Variable Learning Rate Gradient Descent, Gradient Descent with Momentum and Gradient Descent. Since Levenberg-Marquardt algorithm is the fastest one among the mentioned ones, it is used for the belt drive system defined in this thesis.

3.2.3.1 Levenberg-Marquardt

Owing to the need for speed in the training algorithms, gradient descent methods are put a side and new methods developed such as Quasi-Newton, Levenberg-Marquardt and Conjugate gradient. Levenberg-Marquardt is chosen from the fast algorithms in this study

since it gives responses faster and its accuracy is better compared to the other methods for the belt drive model described.

The update in Levenberg-Marquardt is done by using both gradient and Hessian Matrix as follows

$$x_{k+1} = x_k - [H + \mu I]^{-1} g \quad (14)$$

where

x terms in algorithm

H Hessian matrix

μ scalar

g gradient

I identity matrix

Since Hessian Matrix is a second order partial derivatives of the errors, it is easier to use matrix of the first derivatives of error, Jacobian matrix, which can be calculated with the help of backpropagation. The matrix of second order partial derivatives, Hessian matrix is

$$H = J^T J \quad (15)$$

In addition to Hessian matrix, gradient can also be written in terms of Jacobian as follows

$$g = J^T e \quad (16)$$

where

J Jacobian matrix

e error vector

Finally, the update for the terms will become

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (17)$$

The performance function is reduced with decrease in scalar, μ . This is because if μ is large, the situation will be gradient descent with a small step size; on the other hand, if μ is zero the situation will be as in Newton's method which is faster and more accurate near an error minimum [9]

3.2.4 Dynamic Neural Network Concept and NARX

Since the belt drive model parameters are time-dependent, the network with memory, dynamic neural networks, should be used for this study. These networks are demonstrated in the form of Layered Digital Dynamic Network (LDDN) which consists of weights, bias, netprod or summing junction for net input and transfer function. The Nonlinear Auto Regressive with Exogenous Inputs (NARX) model is a kind of dynamic network with feedback. Here, the term "exogenous" refers to the independent inputs and these inputs are gathered from multibody dynamic simulation program for this study. The explanation of this model can be made step by step from linear models to the nonlinear models for inputs with time-series.

STEP 1: Linear Model for SISO (Single Input- Single Output)

The output $y(t)$ is calculated by using past inputs and outputs values as

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) = b_1 u(t) + b_2 u(t-1) + \dots + b_{nb} u(t-nb+1) + e(t) \quad (18)$$

or by leaving the output alone

$$y(t) = b_1 u(t) + b_2 u(t-1) + \dots + b_{nb} u(t-nb+1) + e(t) - a_1 y(t-1) - a_2 y(t-2) - \dots - a_{na} y(t-na) \quad (19)$$

Now, it is possible to write the equation in the vectoral form so that weights can be seen separately.

$$y(t) = [-a_1, -a_2, \dots, -a_{na}, b_1, b_2, \dots, b_{nb}] [y(t-1), y(t-2), \dots, y(t-na), u(t), u(t-1), \dots, u(t-nb-1)]^T \quad (20)$$

The first vector is a weight vector in the form of row vector and the second vector is the regressor which includes delayed inputs and outputs.

STEP 2: Nonlinear Model Extension

If weight vector is deleted and regressor are used in a non-linear function f , the nonlinear extension will be completed. This time output will be

$$y(t)=f(y(t-1),y(t-2),y(t-3),\dots,u(t),u(t-1),u(t-2),\dots u(t-nu)) \quad (21)$$

In fact, it is not a must to use the delayed inputs and outputs. For instance, $y(t-1)^2$, $u(t-1)y(t-2)$, $\tan(u(t-1))$, and $u(t-1)y(t-3)$ can be used in regressor [9].

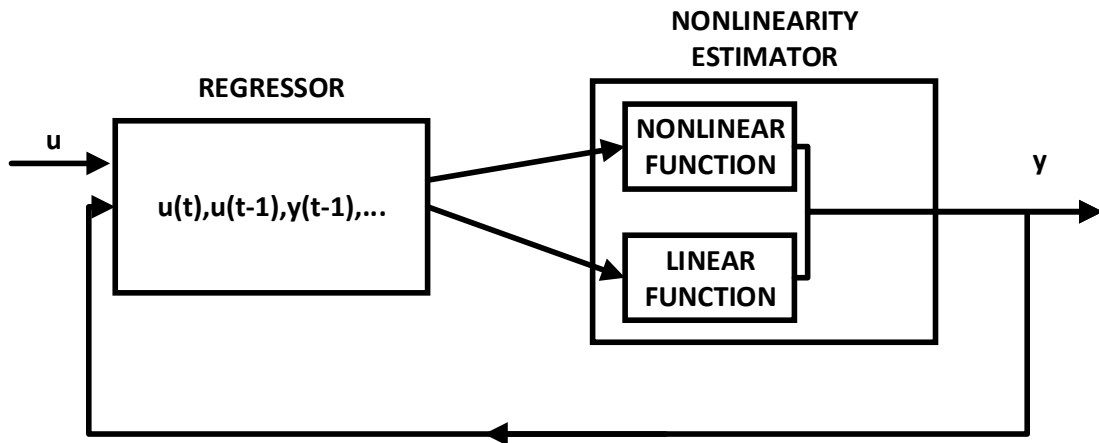


Figure 3.20 Block diagram of NARX

As can be seen from the figure NARX structure composed of two blocks which are regressors and Nonlinearity Estimator. Neural Network will be used as Nonlinearity Estimator for this study. In other words, the nonlinear function f will be represented by neural network.

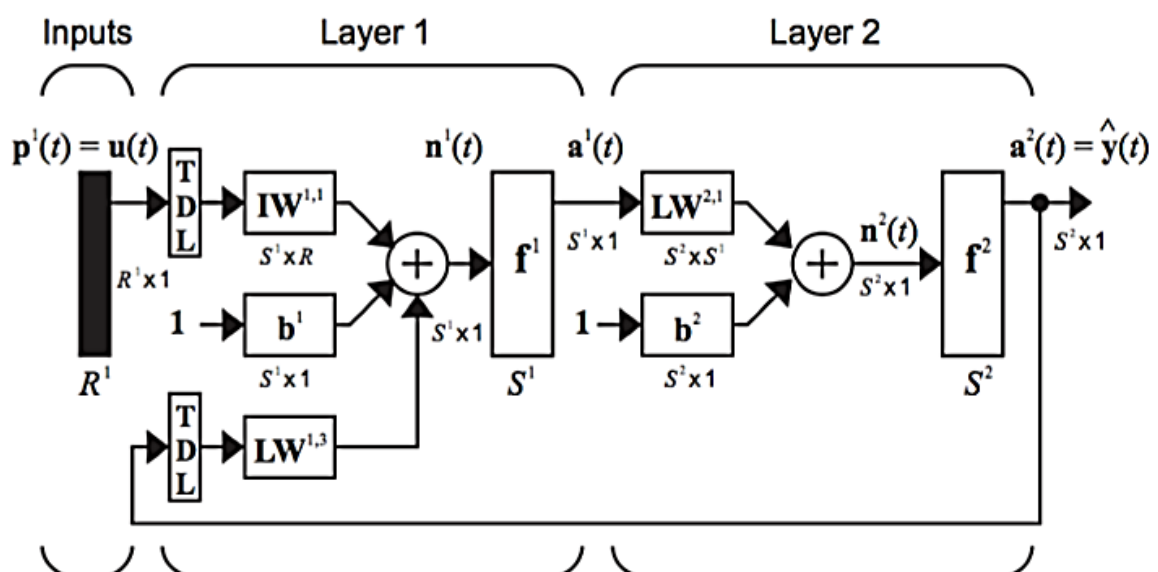


Figure 3.21 NARX architecture with two layer [9]

The above figure shows an example to NARX network structure. The only figure reference not mentioned up to now is Tapped Delay Line (TDL) which is used for the networks with time series.

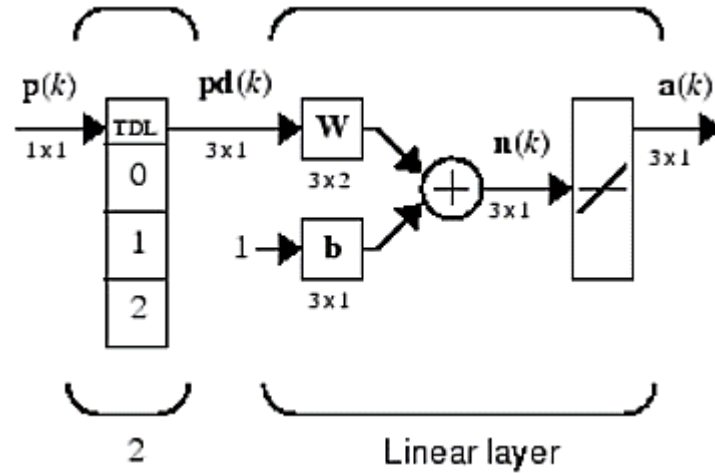


Figure 3.22 Tapped delay line diagram [9]

The tapped delay line with three time zone is given in the above diagram for illustration. As can be seen from the figure the input values are given for 3 different time which are shown with the numbers 0,1 and 2 and these numbers reveals the current signal, the previous signal the signal delayed before that respectively. Since it is required to know the previous terms of input and outputs for NARX formula mentioned previously, delays are used as system's memory which enables to use previous input and output values.

3.3 Model Predictive Control, MPC

Model Predictive Control Theory is divided into five categories as introduction, controller horizons, cost function, constraints and optimizer.

3.3.1 MPC Introduction

Model Predictive Control approach is similar to the human prediction. The steps of a prediction depends on a model which is learned with experience or from the description of an external source. Since how system behaves, the model output, for a limited horizon can be guessed, the control of the system become easier. With the help of future anticipation, the precautions can be taken to be in the safe zone or to be near the desired values.

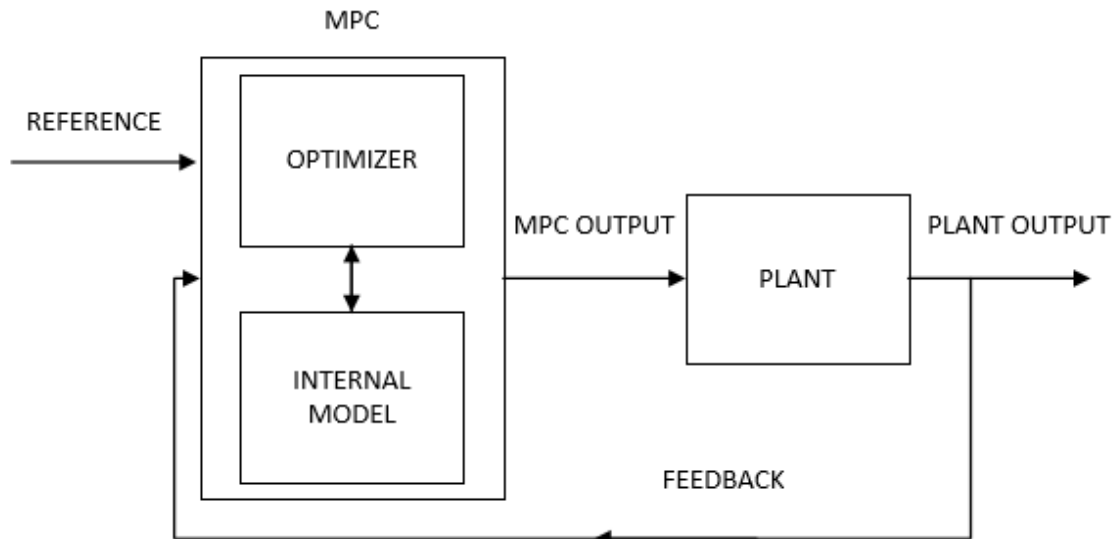


Figure 3.23 MPC block diagram

As can be seen from the above figure, MPC is a closed loop controller like PID; nevertheless, it uses a Plant Model for prediction with an optimizer. Depending on model's being either linear or non-linear, MPC is named as Linear Model Predictive Controller (LMPC) or Non-Linear Model Predictive Controller (NMPC) in some sources of the literature. The distinction between these two classes of MPC is done by the consideration of whether cost functions and constraints are linear or non-linear in some other sources [11]. According to the second definition, MPC is a linear one if only if there is no non-linearity in either cost functions or constraints. Since control input strongly depends on the prediction from the plant model, the model should be as close as possible to the original plant. There are several ways to represent a model such as state-space, transfer function, Hammerstein-Wiener and so on. Despite of the high degree of model closeness to the plant, there is a need for corrections and this is achieved by using feedbacks from plant output to the controller optimizer. In other words, feedback is used to minimize the error with the help of optimizer.

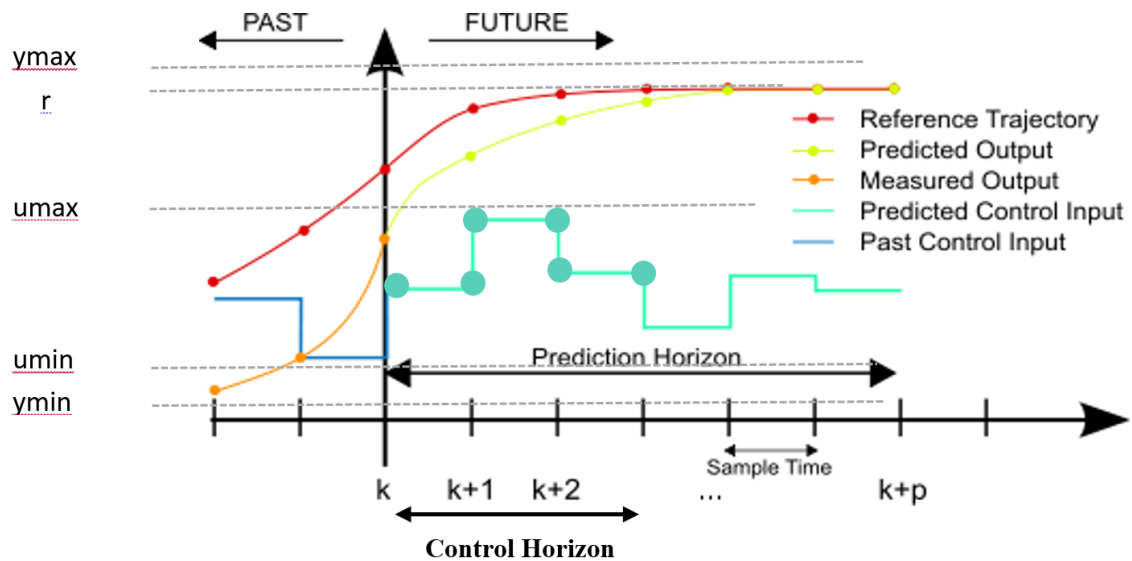


Figure 3.24 MPC working principle

The above figure is a summary of how a model predictive controller works. The left side of the figure shown with arrow below the title 'past' is the measurement step. That is, this part uses the measured inputs and outputs which are shown with blue and orange colors respectively. The right side of the figure shown with arrow below the title 'future' is used to reveal the estimation for outputs and planned movement of the inputs which are shown with ochard (mustard yellow) and green lines respectively. The red line shows the reference trajectory to be followed. The terms mentioned on the left side of the figure can be explained as follows:

y_{max}	maximum value for the predicted output
y_{min}	minimum value for the predicted output
u_{max}	maximum value for the predicted input
u_{min}	minimum value for the predicted input
r	steady state set point for the reference trajectory

Sample time determines the frequency of the output control. Measured values are controlled for a limited time called as control horizon by taking the prediction time or prediction horizon into account. As time goes, the prediction horizon goes forward; that is why, receding horizon controller is the other name of MPC in some literature [12].

3.3.2 Controller Horizon

There are two basic horizon concepts as prediction and control. As their name implies, prediction horizon is used for the purpose of future anticipation. Although plant model enable the controller to anticipate the future, controller not necessarily use the full range of predicted horizon to be in safe zone. Thus, control horizon concept, control action horizon, is arised. There are two basic criteria for the determination of the horizons. The initial criteria for horizon selection should be the objective of the controller according to the plant behavior. The general idea is that as the need for reaction time from controller is desired to be small, control horizon should be shortened. Another criteria is related to computation steps and time. Increasing the horizons give a chance for further anticipation but yields increase in the number of computed variables; therefore, increase in computation time. All in all, horizons should be chosen according to both the specific control purpose of each plant and the idea which tells to escape from the unnecessary computation.

3.3.3 Cost Function

Model Predictive Control is a closed loop control system. It is a closed loop because it uses the feedbacks coming from the plant output. These feedbacks are used to get the predicted values from the internal model of the controller. Then, differences between the reference values and the predicted outputs are minimized for the predicted horizon by penalizing control increments. A cost function is used for this penalizing procedure [13]. The degree of controller performance is measured with cost values. Smaller cost values means better performance. Although cost has a great effect on the optimization algorithm selection, its effect on closed- loop performance is little [12]. Thus, simpler cost functions are preferred mostly in optimization procedure. MATLAB offers a standard cost function as follows:

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_q(z_k) \quad (22)$$

where the output reference tracking term, manipulated variable tracking, manipulated variable move suppression, constraint violation terms can be expressed as follows:

$$J_y(z_k) = \sum_{j=1}^{n_y} \sum_{i=1}^p \left\{ \frac{w_{i,j}^y}{s_j^y} [r_j(k+i|k) - y_j(k+i|k)] \right\}^2 \quad (23)$$

$$J_u(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^u}{s_j^u} [u_j(k+i|k) - u_{j,target}(k+i|k)] \right\}^2 \quad (24)$$

$$J_{\Delta u}(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^{\Delta u}}{s_j^u} [u_j(k+i|k) - u_j(k+i-1|k)] \right\}^2 \quad (25)$$

$$J_q(z_k) = \rho_\epsilon \epsilon_k^2 \quad (26)$$

The control input can z_k can be expressed as

$$z_k^T = [u(k|k)^T \ u(k+1|k)^T \ \dots \ u(k+p-1|k)^T \ \epsilon_k] \quad (27)$$

Meaning of the each symbols mentioned in the above formulas can be expressed as follows:

$J_y(z_k)$	output reference tracking term
$J_u(z_k)$	manipulated variable tracking term
$J_{\Delta u}(z_k)$	manipulated variable move suppression term
$J_q(z_k)$	constraint violation term
k is	current control interval
p	prediction horizon
n_y	number of plant output variables
$y_j(k+i k)$	predicted value of jth plant output at ith prediction horizon step
$r_j(k+i k)$	reference value for jth plant output at ith prediction horizon step
s_j^y	scale factor for jth plant output
$w_{i,j}^y$	tuning weight for jth plant output at ith prediction horizon step
n_u	number of manipulated variables
s_j^u	scale factor for jth manipulated variable
$w_{i,j}^u$	tuning weight for jth manipulated variable at ith prediction horizon step

$w_{i,j}^{\Delta u}$	tuning weight for jth manipulated variable movement at ith prediction horizon step
ϵ_k	slack variable at control interval k
ρ_ϵ	constraint violation penalty weight

As can be seen from the above formula, there are 4 terms with weights for balance. All of the terms are using control input as function variable. Output reference tracking term is used to evaluate the closeness of the result to the desired value; i.e. reference value. If the number of control input or manipulated variable is more than the outputs of the plant, it is necessary to control inputs such that they are either at the target values or near these values. This time Manipulated Variable Tracking term is used for input control. Manipulated variable move suppression is also used for controlling the inputs but for small changes in the input. The last term is Constraint violation term which is used as a performance value to be inside the limits.

3.3.4 Optimizer

The main purpose of the optimizer is to ensure the convergence or closeness of the cost value near zero. In order to achieve this idea, a quadratic program(QP) is used for solution at each control interval in this study.

Figure 3.26 shows how the quadratic program in the model predictive control environment works. It takes differences between the predicted outputs coming from the internal model and reference values and by using this difference it decide what a future input should be according to the constraints and the cost function. The first elements of the future inputs go to the plant as control inputs and the plant outputs are sent to the model which also uses past data and future inputs. The cycle is repeated as soon as simulation continues.

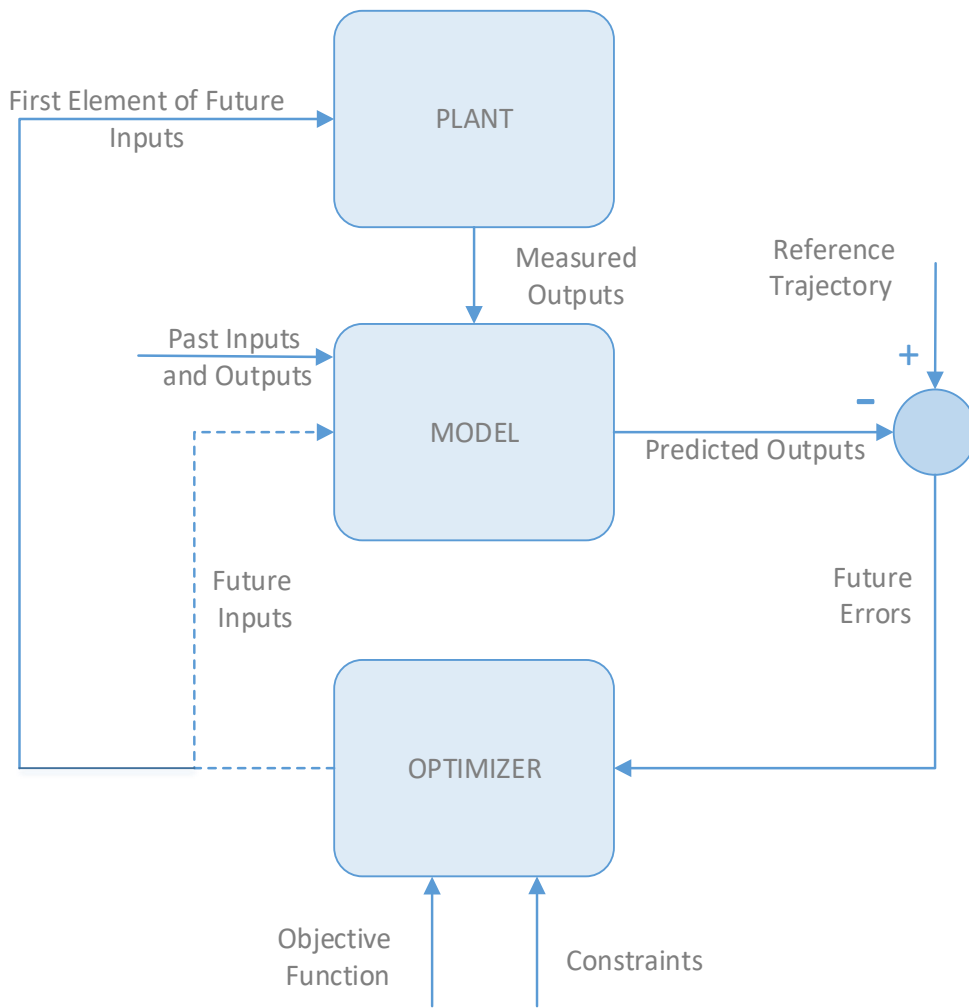


Figure 3.25 MPC block diagram [8]

CHAPTER 4: METHODOLOGY

The methods used to control the belt drive system is introduced in this chapter. This chapter includes six main steps. Initially, multibody dynamic system for the drive system considered in this thesis is built and control system parameters are given with communicator parameters. Then co-simulation steps come and plant block is introduced to simulink. By using the data gathered from co-simulation, an artificial neural network is used for system identification and MPC is designed according to this network. After that, co-simulation is repeated with the designed MPC to control the belt drive system. The block diagram shown in figure 4.2 explains this procedure.

4.1 Belt Drive System with MBD

The simulation system composed of the all fundamentals parts of a serpentine belt system. It includes a driver pulley, a driven pulley (*accessory pulley*), a tensioner and the belt. The idler pulley is not the fundamental part of the serpentine belt system. Thus, it does not exist in the prototype model. In addition to this reason, the increase in number of pulley means increase in the length of the belt and calculation procedure which result in great cpu consumption and longer simulation time for this study.

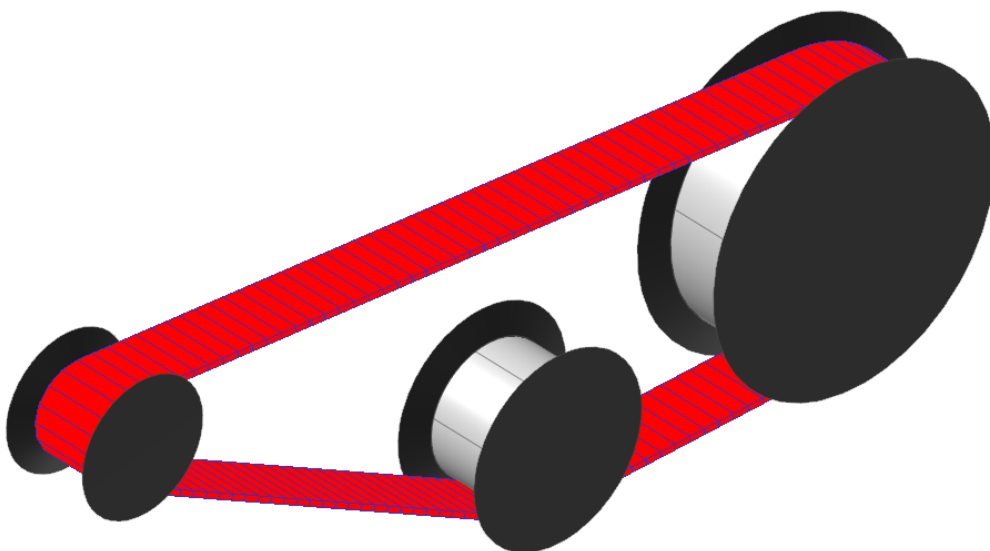


Figure 4.1 The belt-pulley system used as prototype in the thesis

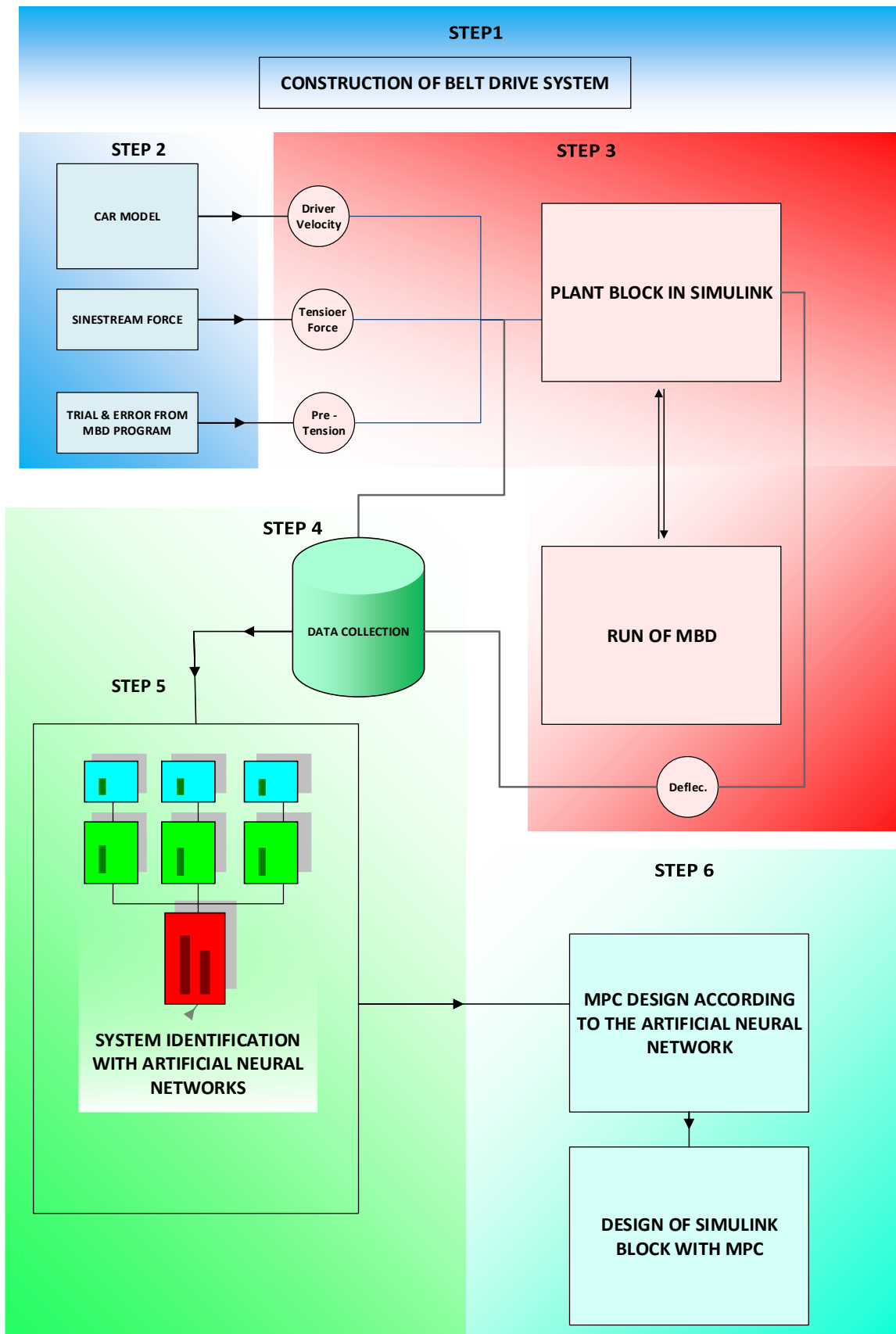


Figure 4.2. The Procedure Followed in the Thesis Methodology

The biggest pulley is the driven pulley, the pulley in the middle is tensioner pulley and the other one is the driver pulley in Figure 4.1.

The properties of materials and boundaries between the parts are established in this section of chapter 4.

The most critical part of the belt drive system is the belt design. The belt used in this study shown in the below figure.

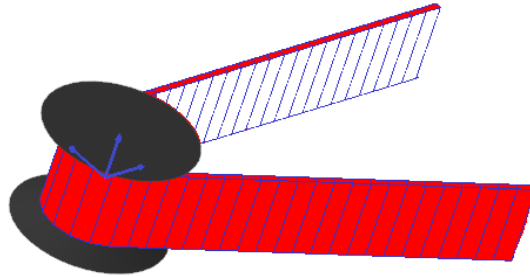


Figure 4.3. The belt system used in the simulation

As mentioned in the theory, belt composed of several beam elements and there can be a measure for the number of elements used. There are two measures for the system in this thesis. The first one is visual inspection from the results. The results here refers to the mid-point deflections of the upper span of the prototype. The results are plotted for number of cases with a step input $2000 * \text{STEP}(\text{time}, 0, 0, 0.5, 1)$ and -1000 N pre-force for 20 seconds in Figure 4.5. The result are shown for the cases from 50 element to 300 element. For 25 elements not shown in the figure the results are not reasonable. As the number of elements increases, the results converge to the ideal value. However, after a certain value, improvement amount decrease. This situation can be understood if the differences between the results of 50 and 75 elements are compared with the differences between the results of 75 and 90 beam element cases.

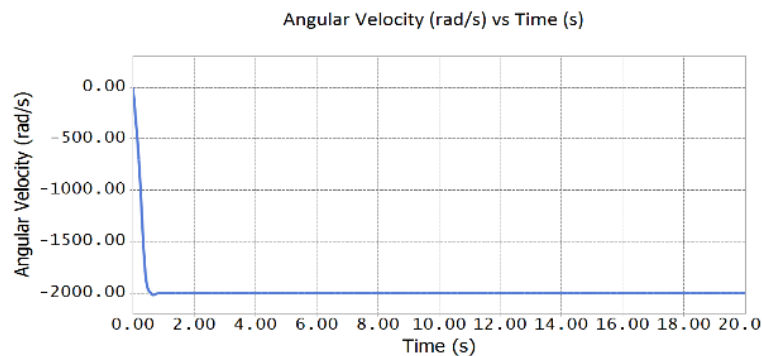


Figure 4.4 Step input used for the determination of beam element numbe

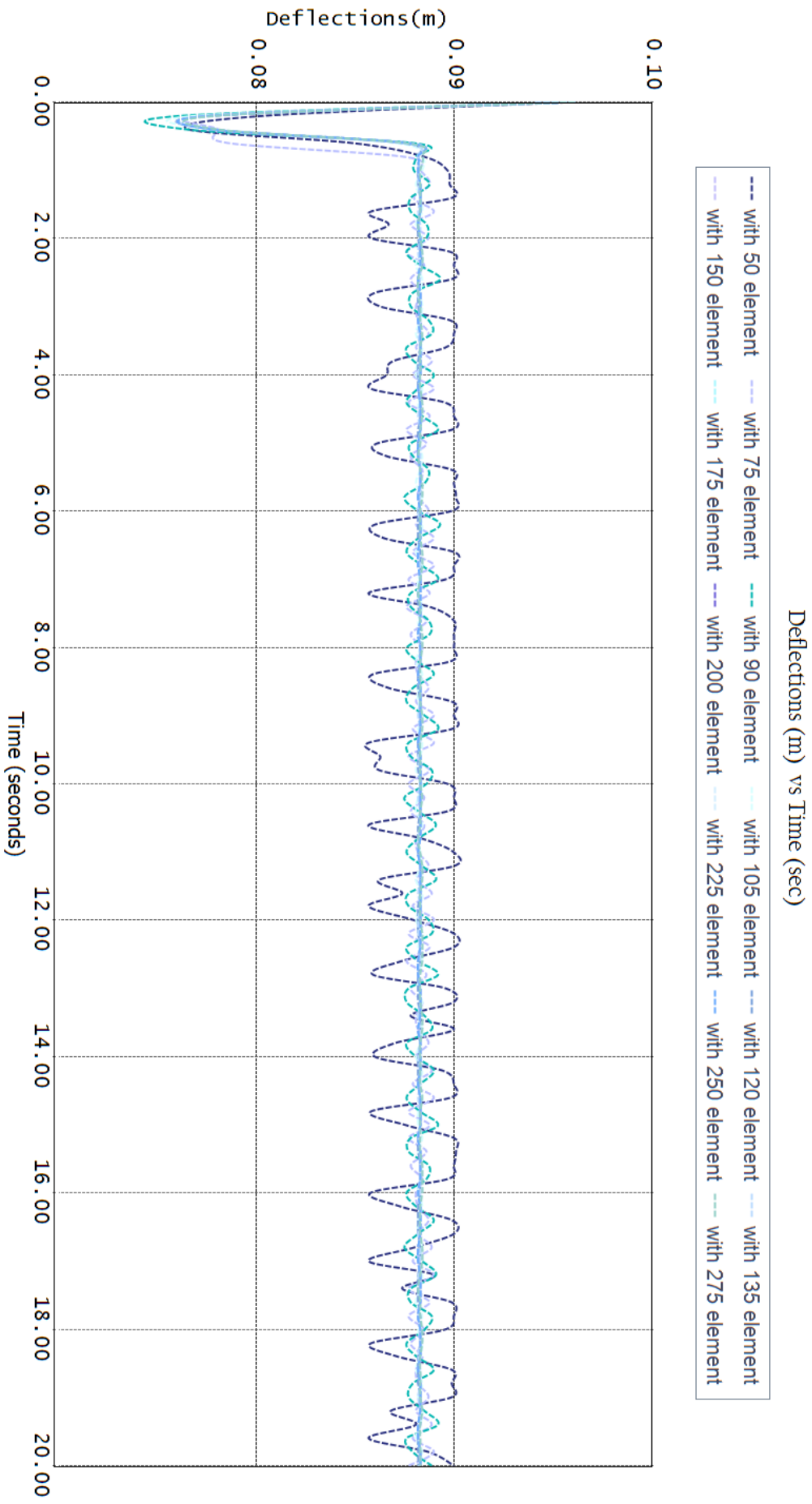


Figure 4.5 Belt Deflections for the belt models with 50, 75, 90, 105, 120, 135, 150, 175, 200, 225, 250, 275, 300 elements

The second measure of stopping criteria is error evaluation. Although the ideal value for the number of beam element is + infinity, 300 element is used as the stopped point for the maximum value. The errors are calculated according to this assumed maximum number. The term error refers to the mean square error (MSE) in this point.

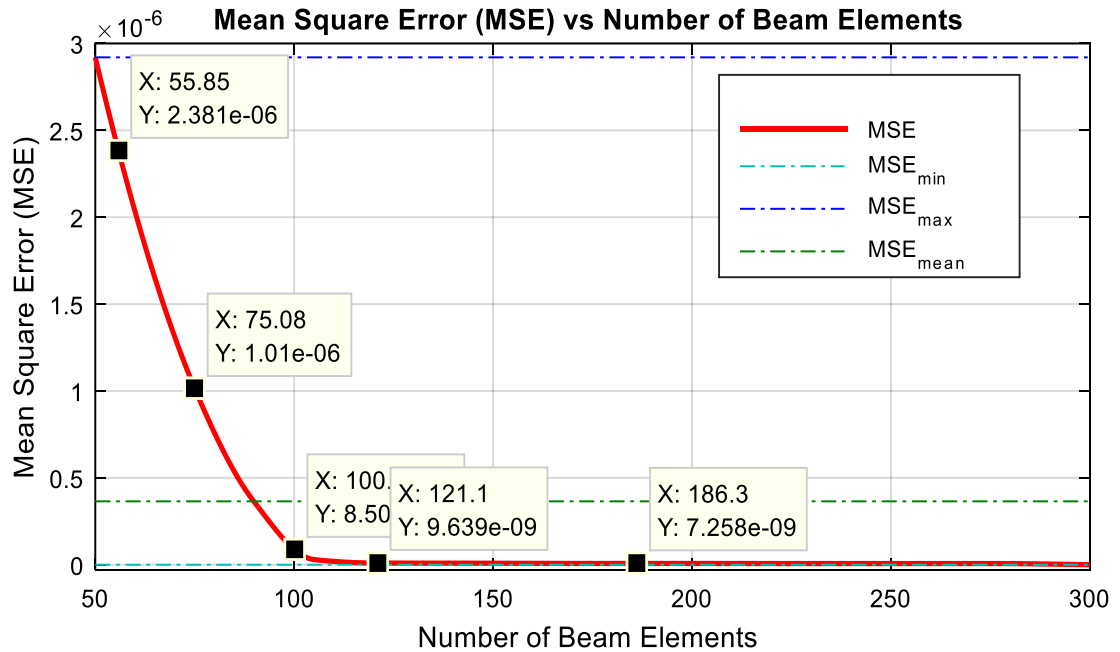


Figure 4.6. Convergence analysis for belt segment

Similar to the Figure 4.5, the improvement decreases with the increase in the number of elements in Figure 4.6. As shown in the above figure, after 100 beam elements the improvement is not at the level of the improvement before 100 elements. Thus, 100 beam elements can be considered as the threshold value for convergence. The change in the inputs can change the deflection amount but the same threshold value is acceptable for the other inputs. In order to be in safe zone 120 beam elements will be used for the belt drive system introduced in this study.

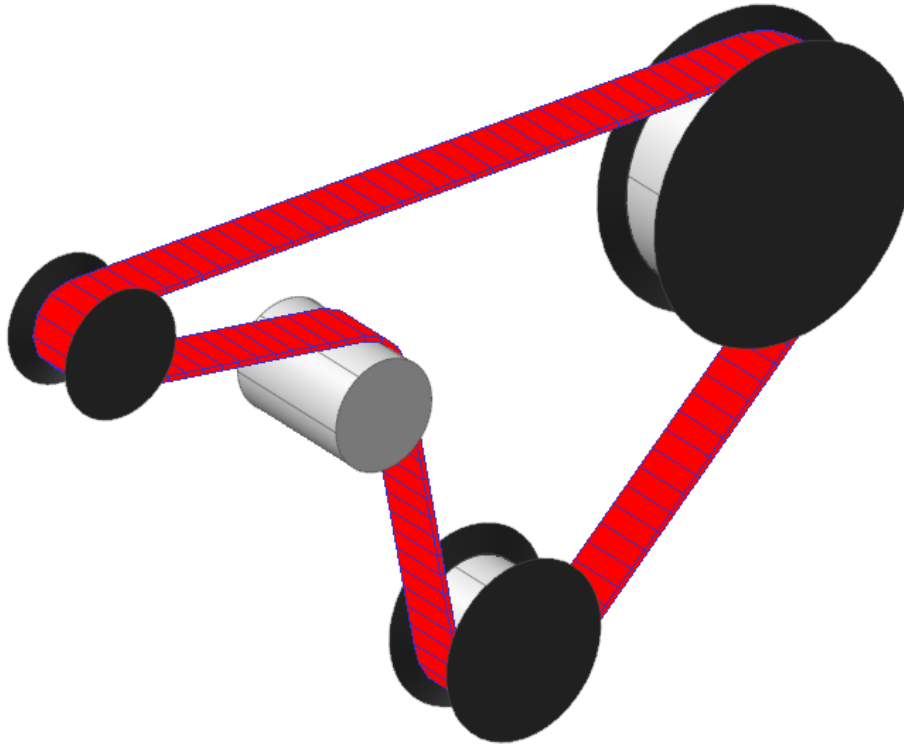


Figure 4.7. The belt-pulley system used in the intermediate level case study

The geometric and material properties of the components of the belt drive system is given in the following tables.

Table 4.1 Beam belt properties

Type/ The cross section	Flat/Rectangular
Lower Thickness: Defines the lower thickness of belt (m)	0.002
Upper Thickness: Defines the upper thickness of belt (m)	0.002
Width: Defines the width of belt (m)	0.050
Number of Elements: Defines the number of force elements to connect nodes(m)	120.0
Element Length: Defines the length of element (m)	0.016
Initial Velocity: Is the initial velocity in the longitudinal direction of belt (m)	0

Table 4.1 and Table 4.2 reveals the belt properties. Whereas material Properties like density, Young' s modulus are given in Figure 4.2 , beam properties such as thickness, width are given in Table 4.1

Table 4.2 Material properties of the belt

Property	Value
Density (kg/m³)	1317
Damping Ratio:	0.003e-1
Young's Modulus: Defines the young's modulus of belt (Pa)	1000e+04
Shear Modulus: Defines the shear modulus of belt (Pa)	4000e5
Moment of area (Ixx) (m⁴)	4.580e-09
Moment of area (Iyy) (m⁴)	6.670e-09
Moment of area (Izz) (m⁴)	1.670e-09
Cross section area (m²)	2.000e-04

Table 4.3 Geometric and material properties of pulleys

Property	Driver Pulley	Driven Pulley	Tensioner Pulley
Radius (m)	2.697e-002	8.890e-002	4.520e-002
Width (m)	5.100e-002	5.100e-002	5.100e-002
Density(kg/m3)	7850	7850	7850
Volume(m3)	1.165e-004	1.266e-003	3.273e-004
Mass(kg)	0.914	9.940	2.569
Young's Modulus(Pa)	2000e+8	2000e+8	2000e+8
Poisson' s Ratio	0.285	0.285	0.285
Ixx(m4)	3.646e-004	2.179e-002	1.869e-003
Iyy(m4)	3.646e-004	2.179e-002	1.869e-003
Izz(m4)	3.327e-004	3.927e-002	2.624e-003

Table 4.4 Geometric and material properties of flanges

Geometric and Material Properties of Flanges			
	Driver' s Flanges	Driven' s Flanges	Tensioner's Flanges
Radius (m)	2.697e-002	8.89e-002	4.52e-002
Width (m)	0.007	0.007	0.007
Angle (degree)	70.00	70.00	70.00
Density(kg/m³)	7850	7850	7850
Volume(m³)	3.011e-005	2.140e-004	6.675e-005
Mass(kg)	0.236	1.680	0.523
Young's Modulus(Pa)	2000e+8	2000e+8	2000e+8
Poisson' s Ratio	0.285	0.285	0.285
Ixx(m⁴)	8.889e-005	4.145e-003	4.153e-004
Iyy(m⁴)	8.905e-005	4.153e-003	4.161e-004
Izz(m⁴)	1.761e-004	8.284e-003	8.273e-004
Ixy(m⁴)	2.757e-008	1.886e-006	1.951e-007
Iyz(m⁴)	-4.791e-011	-1.552e-010	3.249e-011
Izx(m⁴)	1.614e-010	-.246e-009	-.784e-010

Table 4.5 Position of pulleys and flanges for prototype model

Components	x (m)	y (m)	z(m)
Driver Pulley	0	0	0
Driver' s Flange Front	0	0	0.026
Driver' s Flange Back	0	0	-0.026
Driven Pulley	0.552	-6.043e-002	0
Driven' s Flange Front	0.552	-6.043e-002	0.026
Driven' s Flange Back	0.552	-6.043e-002	-0.026
Tensioner Pulley	0.328	-0.124	0
Tensioner's Flange Front	0.328	-0.124	0.026
Tensioner's Flange Back	0.328	-0.124	-0.026

Table 4.6 Position of pulleys and flanges for four pulley system

Center of Bodies			
	x (m)	y (m)	z(m)
Driver Pulley	0	0	0
Driver' s Flange Front	0	0	0.026
Driver' s Flange Back	0	0	-0.026
Driven Pulleys 1	0.552	-0.060	0
Driven Pulleys 2	0.200	-0.100	0
Driven' s Flange Front	0.552	-0.060	0.026
Driven' s Flange Back	0.552	-0.060	-0.026
Tensioner Pulley	0.328	-0.324	0
Tensioner' s Flange Front	0.328	-0.324	0.026
Tensioner' s Flange Back	0.328	-0.324	-0.026

Table 4.7 Contact properties of pulleys and flanges

Property	Pulleys	Flanges
Spring Coefficient (N/m)	1000000.	1000.
Damping Coefficient	1000	1
Dynamic Friction Coefficient	0.6	0.6
Stiffness Exponent	1.3	1.3
Damping Exponent	1	1
Indentation Exponent	2	2

Table 4.8 Properties of connecting force

Property	Values
Translational Stiffness (N/m)	1000000000.
Translational Damping Ratio	1.e-002
Rotational Stiffness(N/m)	100000000000.
Rotational Damping Ratio	1.e-002

4.2 Plant Variables

The most important step of the system estimation is to decide what to control and on which system elements will be used to trigger the system. They are strongly related to the purpose of the control. Since the interest of this study is related to control of the transverse vibration of the serpentine belt system, control variables are chosen accordingly. The system variables will be detailed in this section.

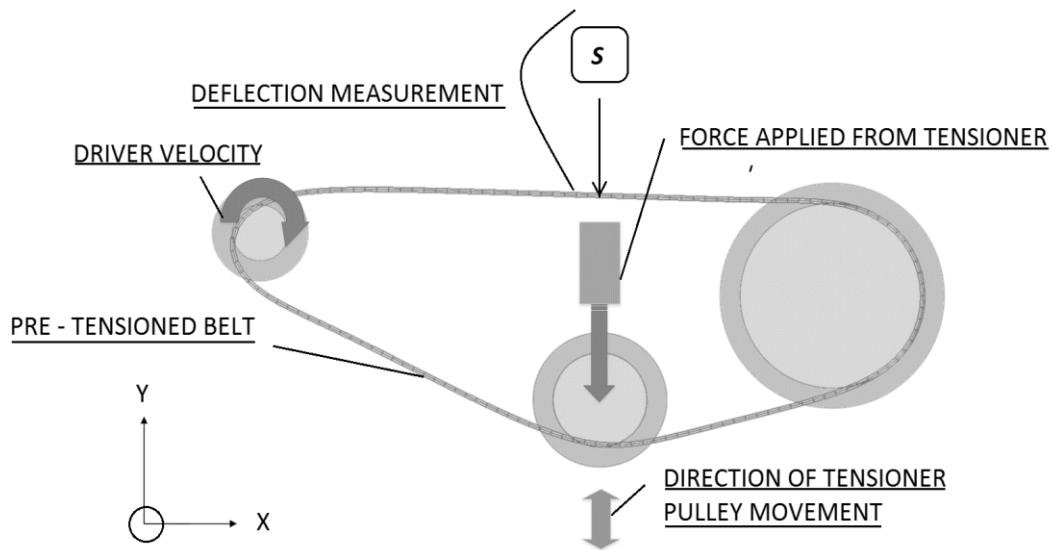


Figure 4.8 Serpentine belt system components

The study in this thesis depends on the fundamental prototype model shown above. This system begins its movement with the motion of the driver pulley. In order to guarantee the contact between the pulleys and belt, pre-tension is applied. This pre-tension value is decided after several trial and error. Its value is adjusted such that it both satisfy the minimum requirement for the contact and allow the belt to oscillate to some extent. If this pre-tension is applied so much, then it will bring about undesired consequences as mentioned in the literature survey. That is why tensioner pulley is controlled with predictive control in this study. Another input to the system is the displacement of the tensioner pulley in the transverse direction. By putting these inputs into the system, the transverse displacement of the belt will be tried to control. The deflection are gathered as the value at the middle of the span which is between the driver and driven pulleys. By controlling of this specific point not only the inspected point deflection will be under control but also the whole belt motion will be controlled since each point on the belt affects other points as well. Once deciding on the working region, the inputs can be given to the system. The rest is collecting the data, estimating the system behavior and implementation of the controller designed according to the identified system behavior.

To sum up, there are three inputs for belt drive plant totally

- *Pre-Tension*
- *Driver Velocity*

- *Tensioner Force*

The only manipulated input to control the oscillation in transverse direction is the tensioner force.

4.2.1 Pre-Tension

As mentioned in the previous section, there are three inputs for the belt drive system. The pre-tension is a fixed value and decided according to the trial and error method so that it both satisfy the minimum requirement for the contact and allow the belt to oscillate. It is achieved by adjusting the force according to the minimum value which satisfies the contact between the belt and pulleys.

Pre- Tension value is chosen as -580 N for the prototype model. However, it does not mean that the minimum value for the contact satisfaction is -580N. Although the minimum value is below -580 N, this value is chosen to be in a safe zone. How pre-tension is adjusted is shown in the following figure. Below figures shows the plant input to determine pre-tension for illustration. Driver velocity is given as will be mentioned in the next section.

Figure 4.9 shows the results of the belt drive prototype for the inputs given in the previous figure. This figure consists of two parts. Whereas the lower portion shows the deflection amount of the belt drive for whole simulation time, the upper portion shows the position of the belt drive system at 3rd second for illustration purpose. As can be seen from the figure, the belt contact is not lost for -500 N, yet the position of the driver pulley is not desired level in terms of contact satisfaction. That is why pre-tension is adjusted somehow bigger than the minimum value. One another important inference is that the deflection amount is directly affected from the velocity profile. Since there is no tensioner force and pre- tension value is constant, the effect of driver pulley velocity profile on the deflection is reasonable.

Time = 2.99999997 Second

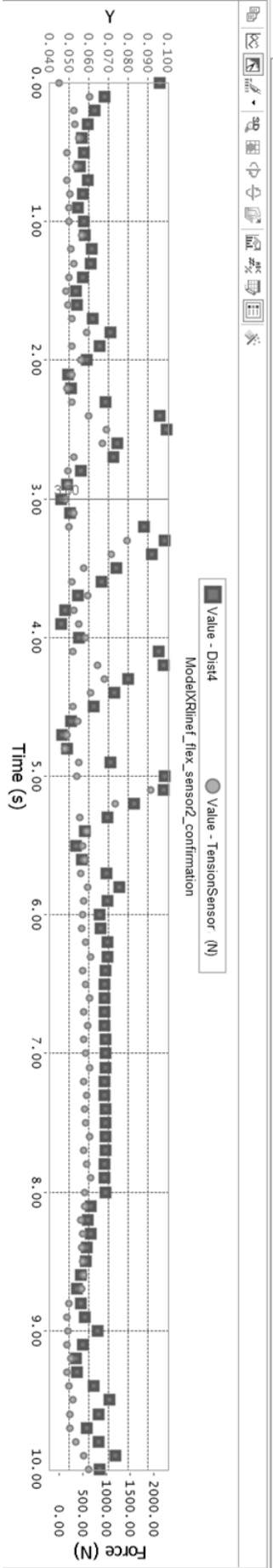
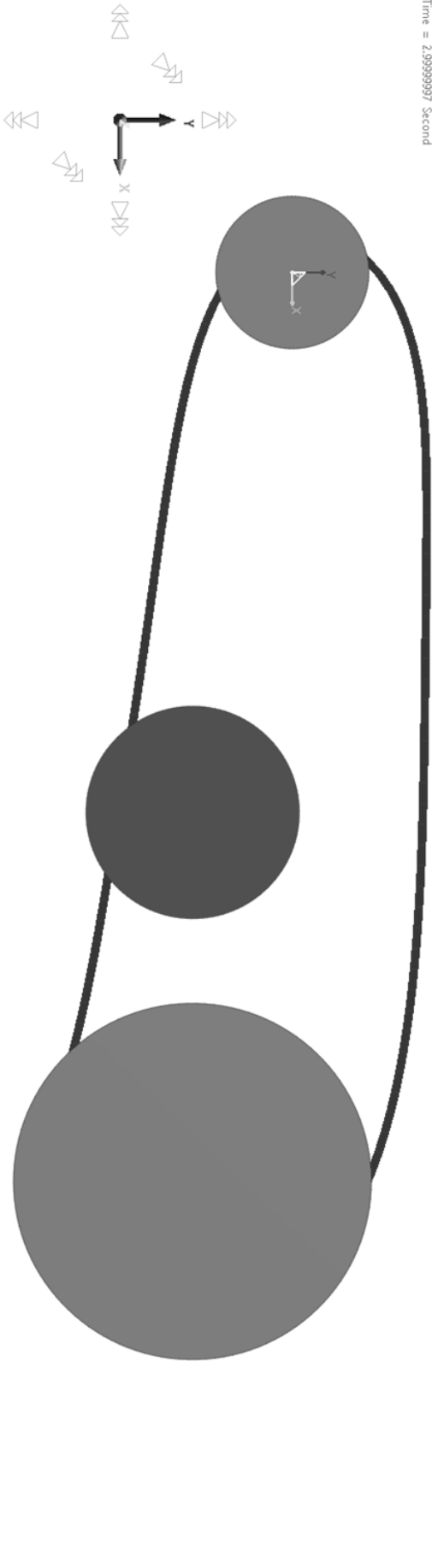


Figure 4.9 A screenshot from the simulation used for pre-tension determination

4.2.2 Driver Velocity

Instead of given random values for the driver pulley velocity, a full car model is used in Simulink environment which will be detailed in this section. However, before gathering the inputs from full car model; step, ramp, sinusoidal and impulse inputs are used as will be mentioned in chapter 5. The details can be found both in section 4.4.2 and chapter 5.

4.2.3 Tensioner Force

The only manipulated input, force applied from the tensioner to the belt, is selected so that the overall system behavior under the effect of different forces can be analyzed. In order to better understand how the system behavior can be learned, the below explanations are given.

Frequency response of the system reveals the characteristics of the system and estimation of the frequency response is done by either sinestream or chirp input signals.

The swept-frequency sine (chirp) excites the system at a range of frequencies which changes instantaneously as shown on the below figure.

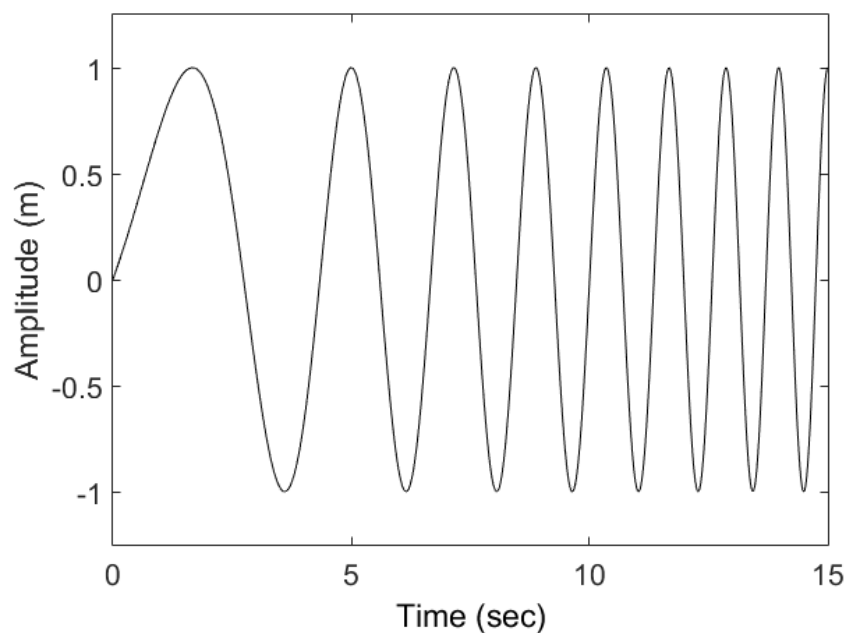


Figure 4.10. A chirp wave

Even though chirp wave can be a good way for nearly linear models if it is desired to quickly obtain a response for many frequency points, it is not suitable for the systems with strong nonlinearities because there is a need for time to estimate the system behavior in case of nonlinear models. Therefore, many adjacent sine waves of varying frequencies which excites the system for a period of time will be the solution. This is called as sine- stream signal.

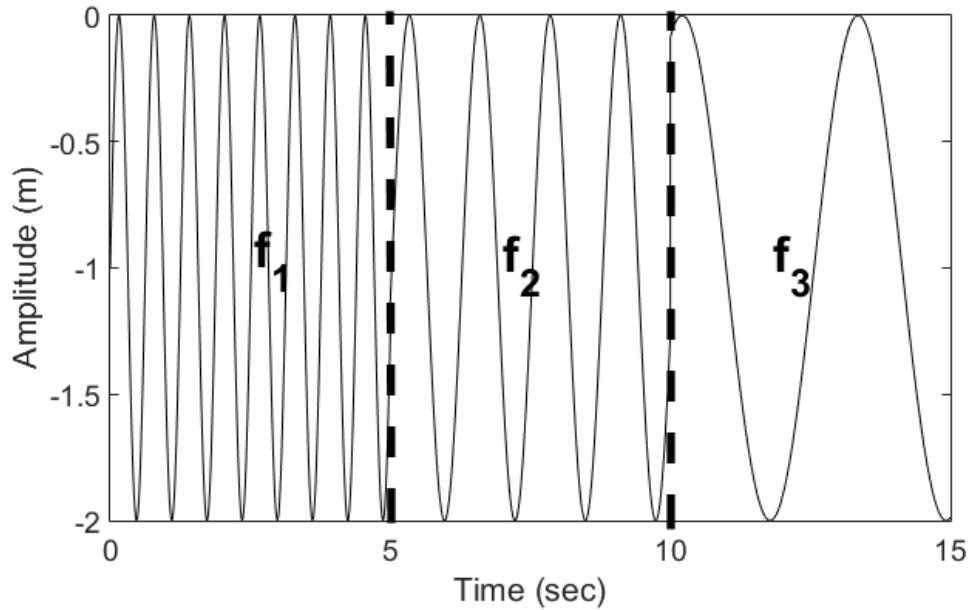


Figure 4.11 A sine- stream wave

As may be seen in chapter 5, sinusoidal functions at varying frequencies are used for system identification of the plant which will be controlled with a predictive algorithm.

4.3 Co- Simulation

In this section, multi- body dynamic simulation program and Simulink are tried to be worked together. Initially, input and outputs are defined in MBD program. Then, how MBD program will work with Simulink is defined in MBD program by giving information related to host program, control time step (sampling period or time), Plant Block M-File, Simulink run M-file. After preparing the program for co-simulation, plant block is established with proper queries. The next step is just designing the blocks in Simulink environment and using the data obtained from this co-simulation.

Figure 4.12 gives a general idea of the co-simulation. As can be seen from this figure, there is a shared memory between co-simulated programs and with the help of data transfer from one to another co-simulation is accomplished. The following figures summarize the all steps covered in this section.

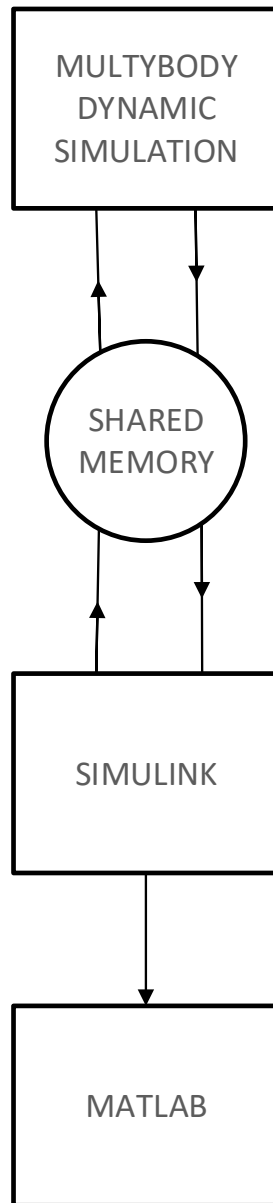


Figure 4.12 Co-simulation fundamentals

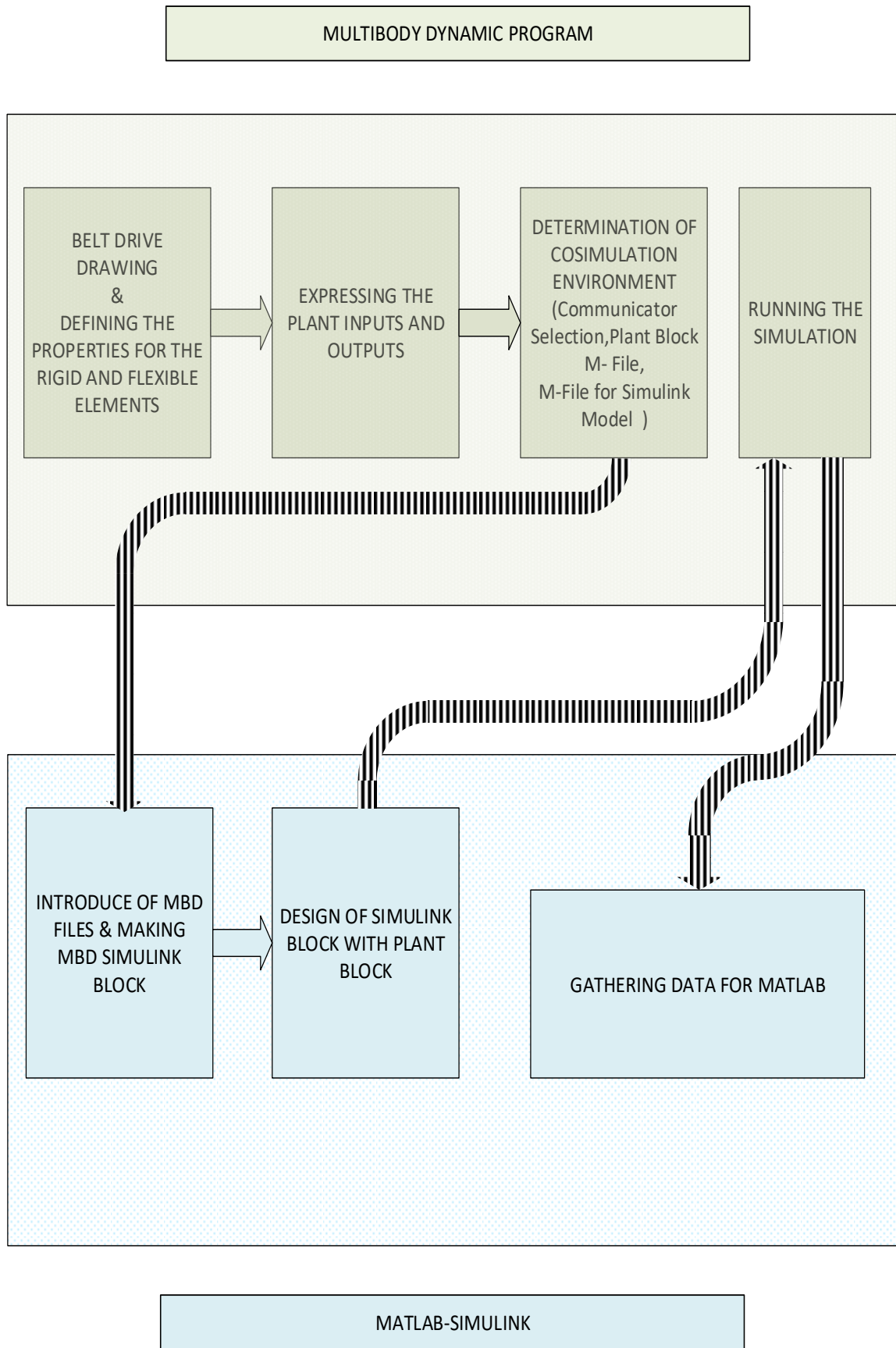


Figure 4.13 Co-simulation steps

4.4 Case Studies and Velocity Profiles

Up to this section the plant, plant inputs, co-simulation procedure are introduced. This section is the last section before plant identification. Therefore, the multi-body dynamic plant models and the plant-inputs used for this study are summarized conceptually. Collected data for both input and the output will be given in the fifth chapter together. This section is divided into three main categories. Data collection for different geometric constructions are introduced in the first part. The second part includes scenarios for several velocity profiles. The third part explains the situations for different forces.

4.4.1 Case Studies

This category is divided into two depending on the components' geometric configuration of the belt drive system.

4.4.1.1 Elementary Level Cases

The elementary level case configuration is simply the prototype used in the belt drive system design.

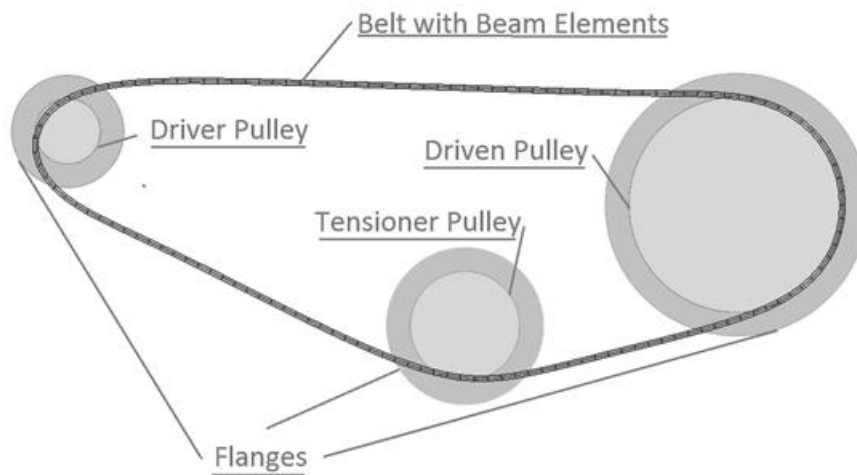


Figure 4.14 Elementary level configuration

4.4.1.2 Intermediate Level Cases

In addition to the belt drive system introduced in elementary level case, an extra pulley is added to the system in order to increase the system complexity.

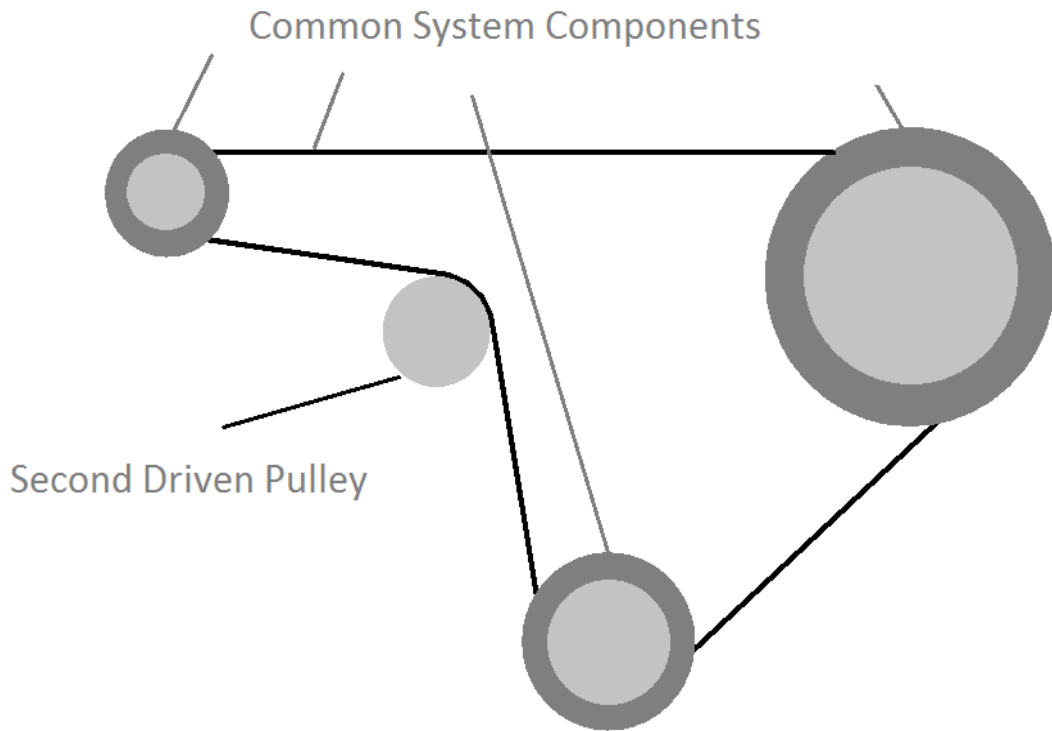


Figure 4.15 The belt-pulley system used in the intermediate level case study

4.4.1.3 Advanced Level Cases

In order to control a more complicated case, the rotation center is transferred to 3 mm above the geometric center for driver pulley.

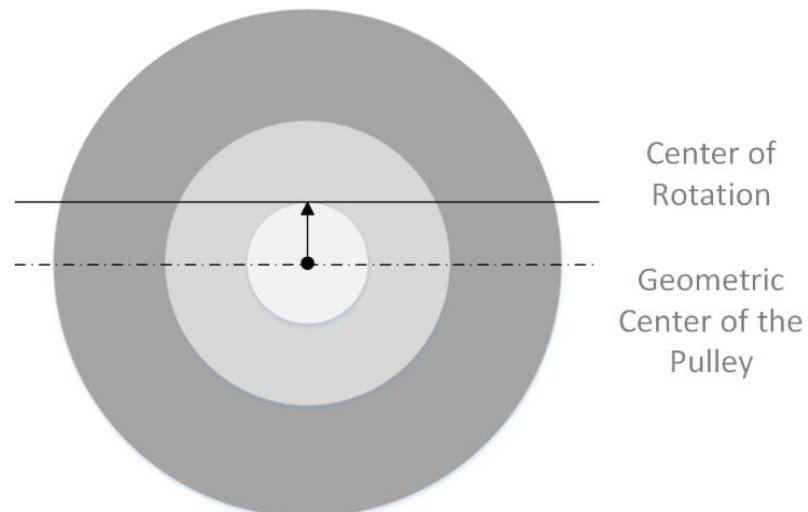


Figure 4.16 Driver pulley for advanced case

4.4.2 Velocity Profiles

Oscillation of the belt span will be observed for the transient states where there can be considerable amount of belt deflection compared to the steady state belt deflection. The velocity profiles gathered from Simulink car model are collected according to the four different situations as follows:

- *Sudden Increase in Vehicle Speed*
- *Sudden Decrease in Vehicle Speed*
- *Combination of Case 1 and Case 2.*
- *Repetition of Case3 for Narrower Brake- Throttle Range*

The behavior of the belt drive system will be tested for the above- mentioned situations, tried to be identified and controlled accordingly.

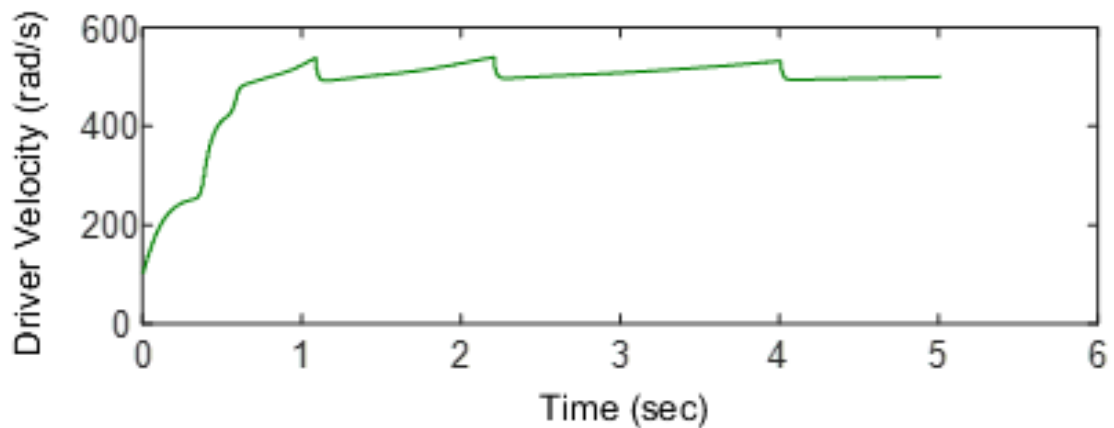


Figure 4.17 Driver velocity for case 1

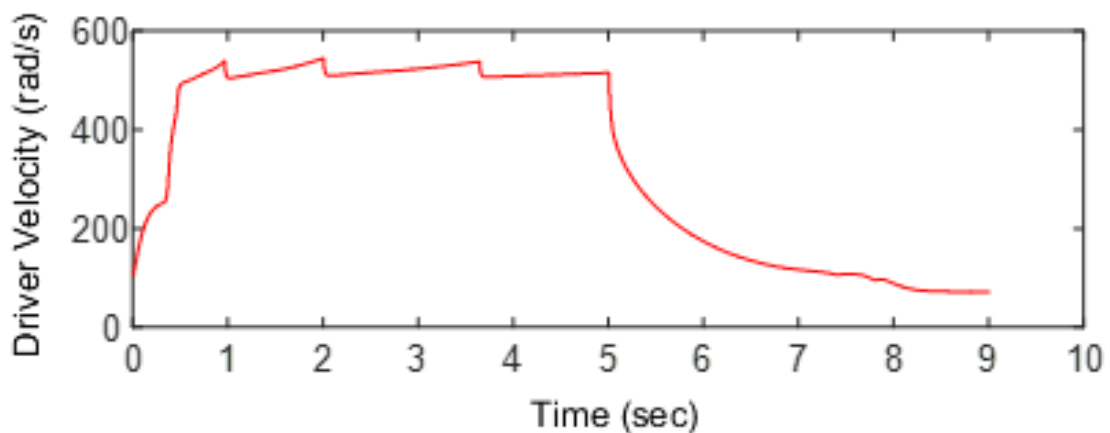


Figure 4.18 Driver velocity for case 2

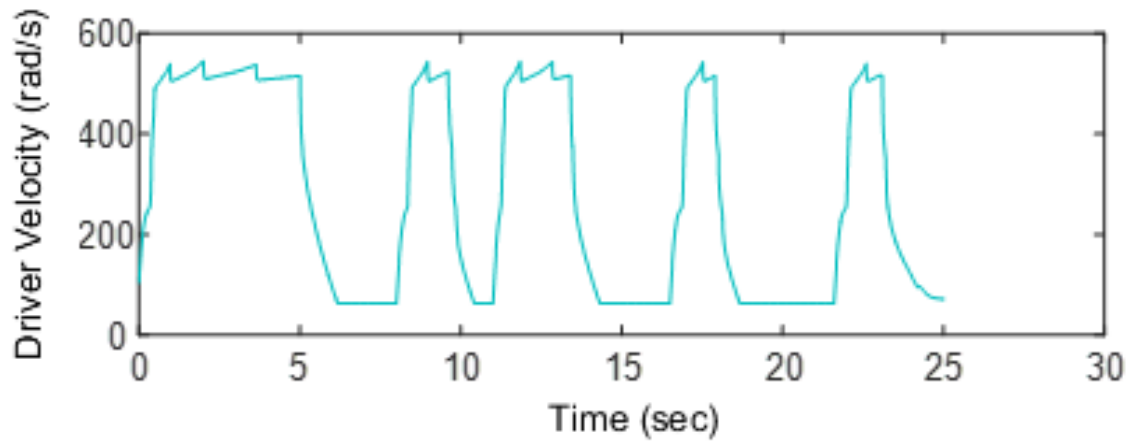


Figure 4.19 Driver velocity for case 3

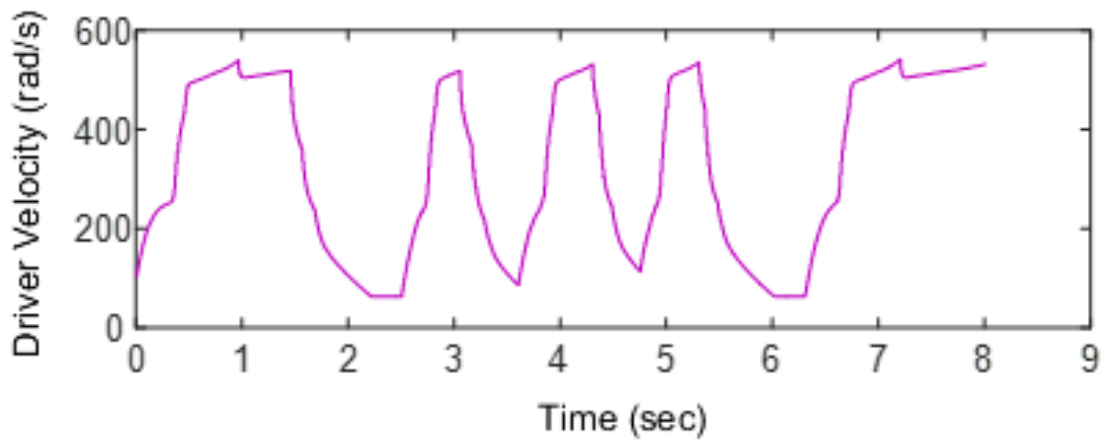


Figure 4.20 Driver velocity for case 4

The sudden increase in the driver velocity can be seen from all of the figures. Although the second case is named as sudden decrease in vehicle speed, the velocity is increased suddenly to have a velocity whose velocity will be decreased. The increase in the velocity is done suddenly because it is not only shorten the simulation time but also transient behavior belt span motion is achieved which is the necessity of this study.

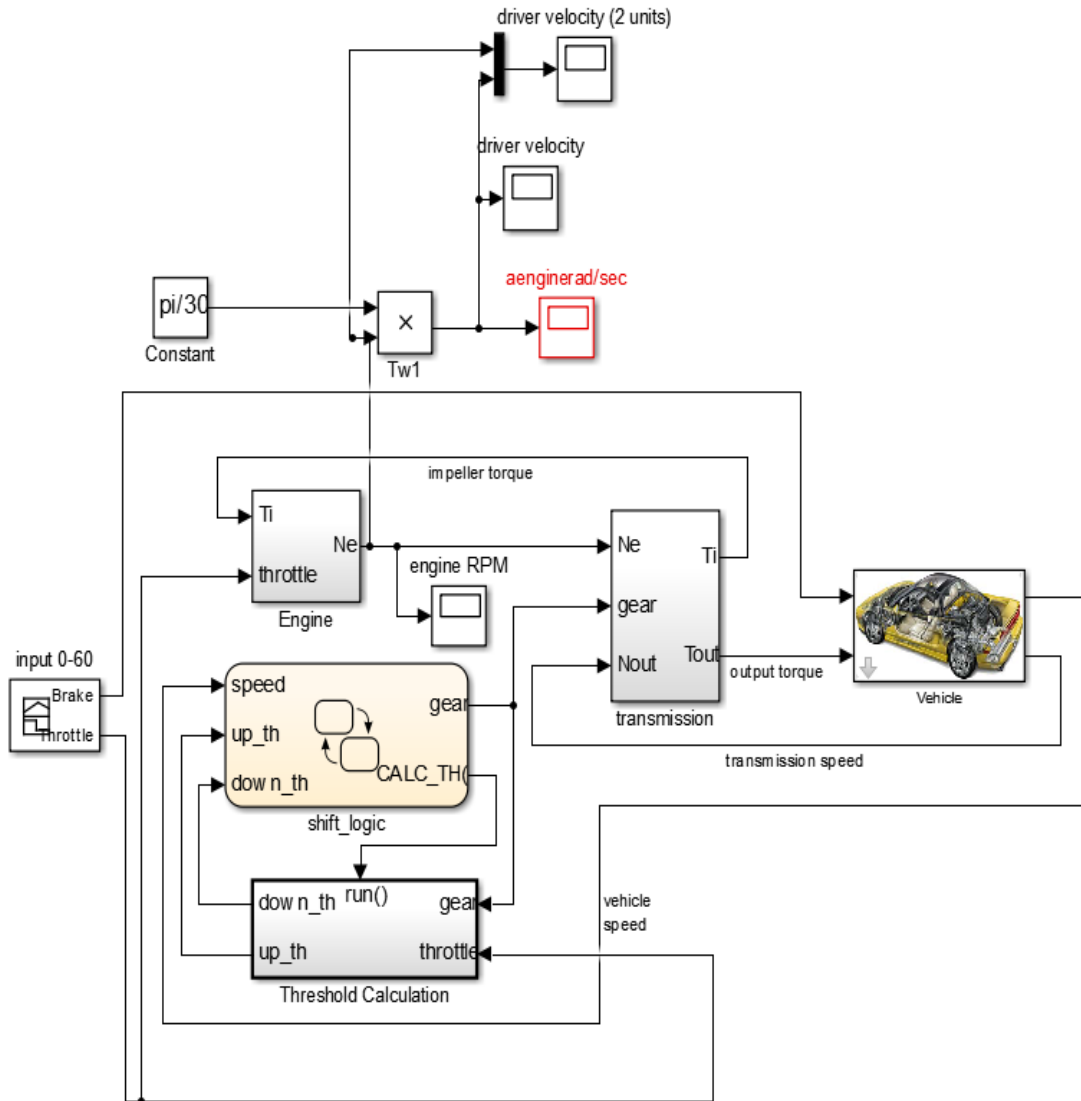


Figure 4.21 Full-car model block diagram in simulink

Above figure demonstrates the typical full car model supported by MATLAB for automotive related simulations. The next figure is the detailed view of the full car model. It can be beneficial to note that the torque converter and transmission ratio are both sub-classes of transmission part shown in Figure 3.27. In addition to the inputs gathered from the full car model, random step, impulse, sinusoidal and ramp inputs are employed as can be seen in chapter 5.

4.5 Building Artificial Neural Network for Plant Model

The only knowledge about the system is the inputs and outputs; that is, the system is a black box. In order to make a proper identification for the belt drive system, artificial neural networks are preferred.

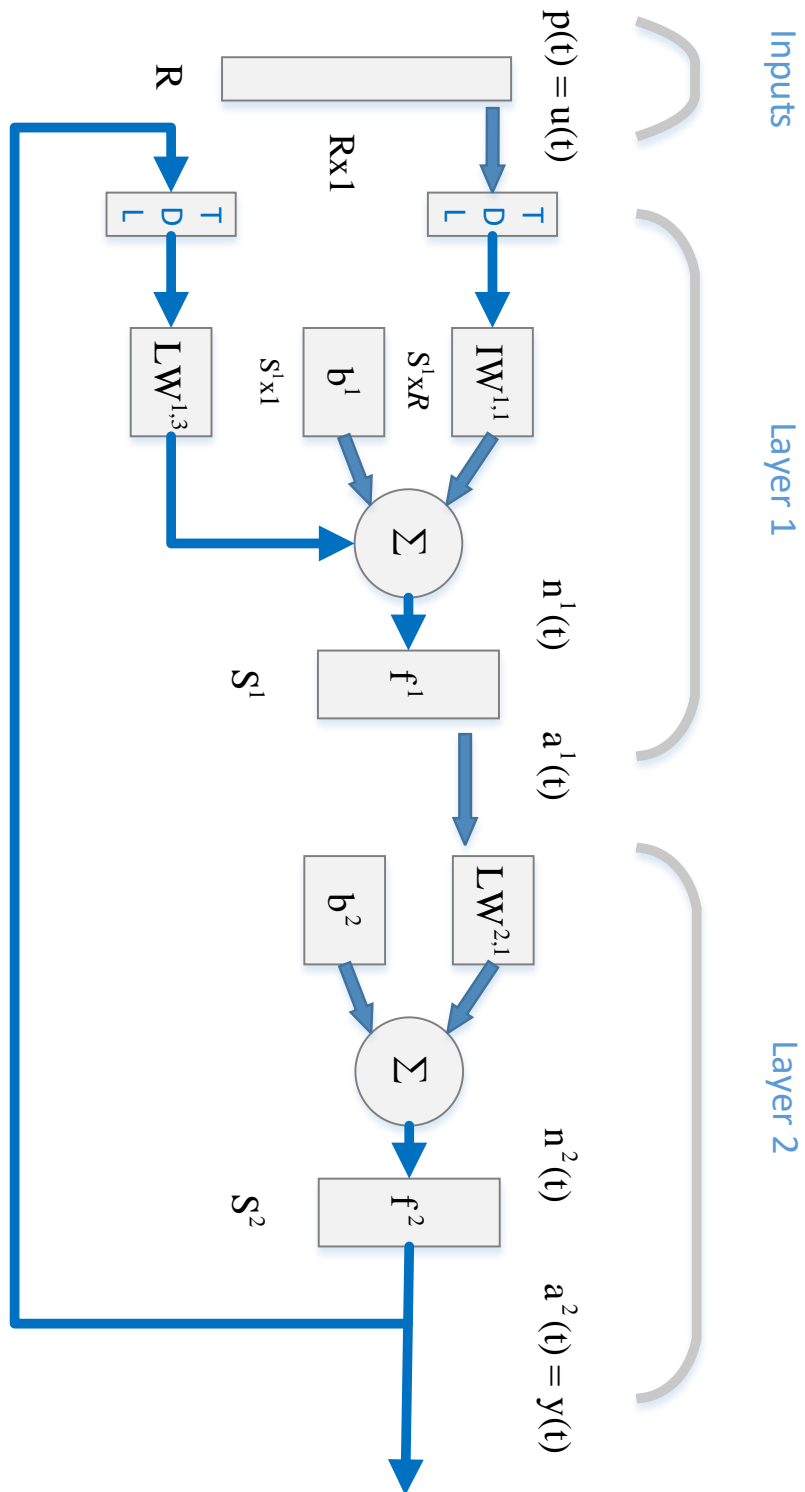


Figure 4.23 Narx structure for 2 hidden layers

In order to build a neural network, several criteria should be in consideration. After several trial and error the network is shaped as demonstrated in the tables below. The same properties used for the all configurations used in this thesis.

Table 4.9 Properties of artificial neural network used for the belt drive system identification

Network Type	NARX
Training Function	trainlm
Hidden Layer Transfer Function	tansig
Output Layer Transfer Function	purelin
Input Delays	4
Feedback Delays	4
Number of Hidden Neurons	20
Data Preparation Function	preparets
Data Division Function	dividerand
Training Ratio	70
Validation Ratio	15
Testing Ratio	15
Performance Function	mse

The explanations of the above properties can be summarized as follows:

<i>NARX</i>	nonlinear autoregressive network with exogenous inputs
<i>trainlm</i>	Levenberg-Marquardt backpropagation
<i>tansig</i>	hyperbolic tangent sigmoid transfer function
<i>purelin</i>	linear transfer function
<i>preparets</i>	a function used for preparation of input and target time series data to train the network

dividerand used to divide the data into training, validation and test sets

mse mean square error

The transfer functions mentioned in the table are drawn in the following figures. The input values here refers to the net input in the neuron structure and the output is the network output.

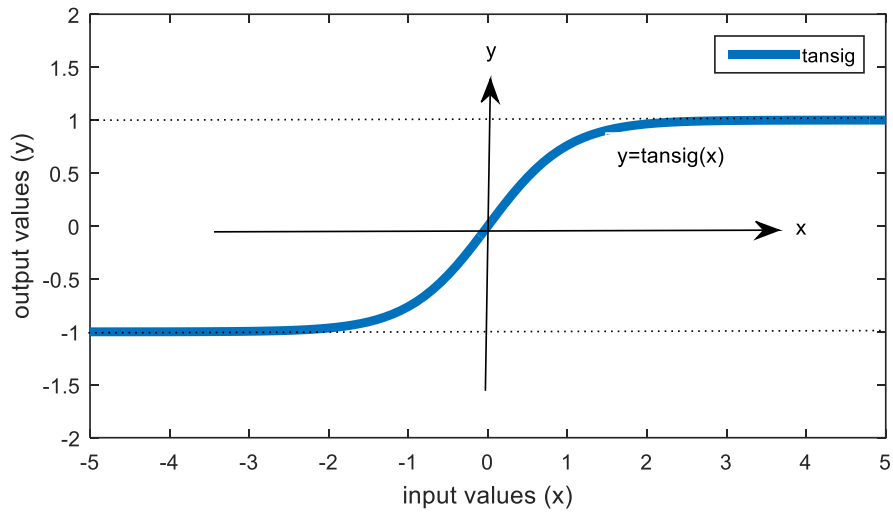


Figure 4.24 Hidden layer transfer function

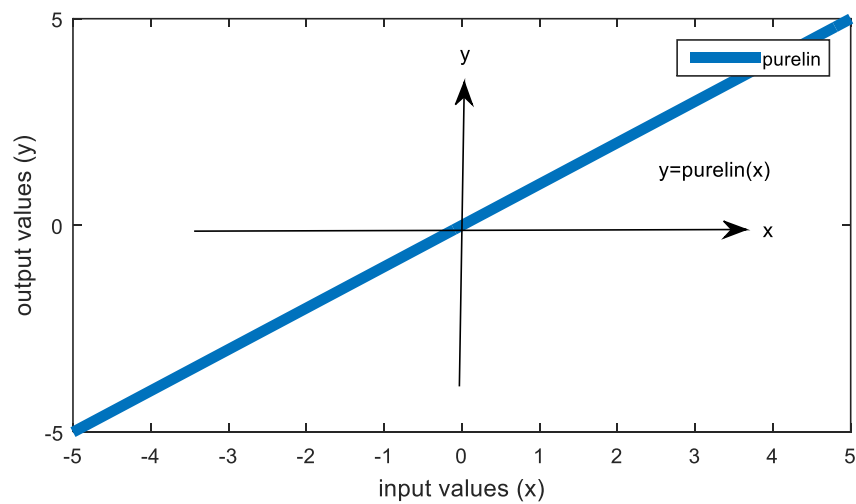


Figure 4.25 Output layer transfer function

The following figure covers the all critical features of the artificial neural network introduced in this section.

ur 4.49 Co-Simulain Blocks with MPC

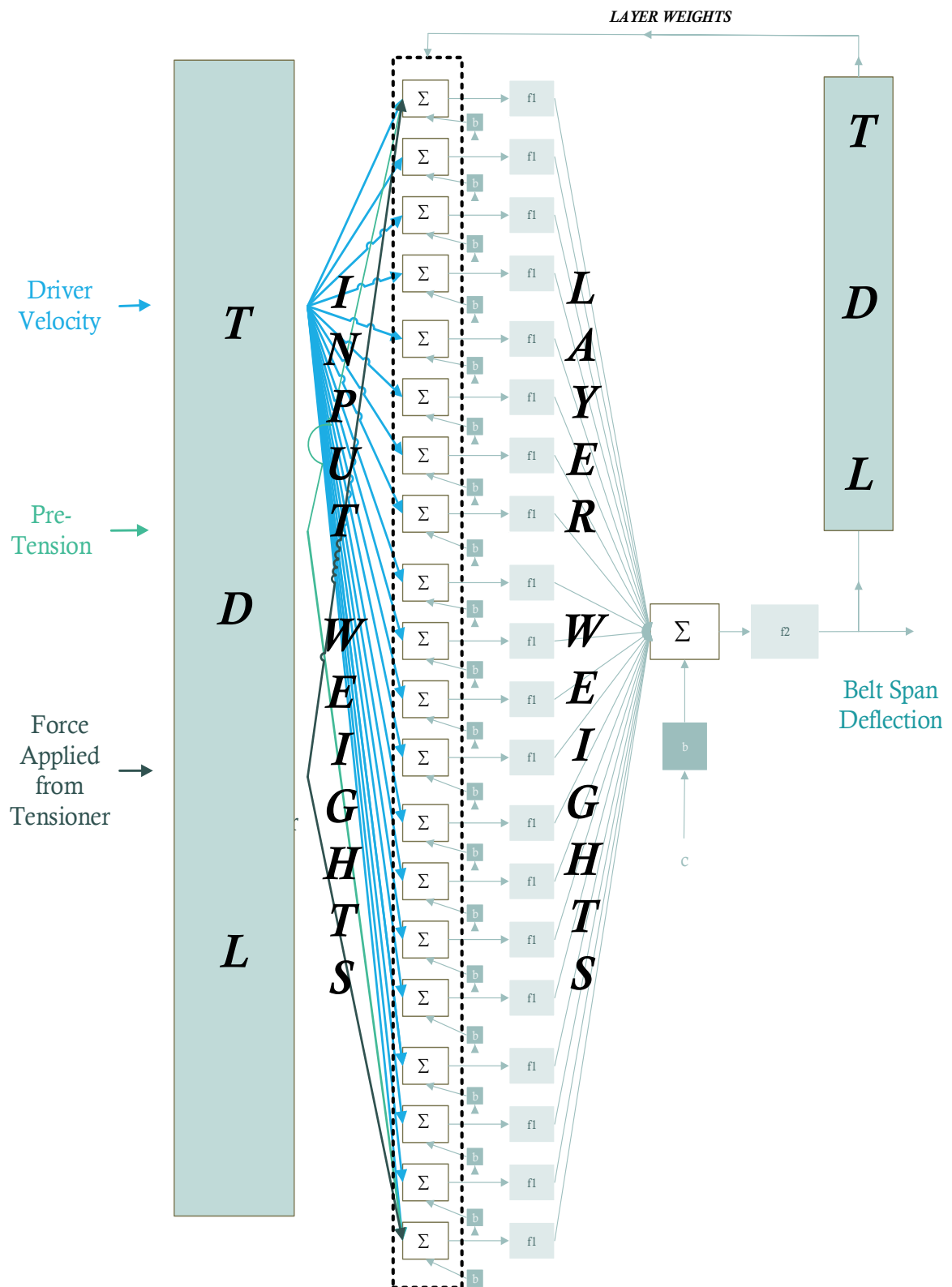


Figure 4.26 Artificial neural network used in this thesis

4.6 MPC Design with NN Approximation

As mentioned in the literature review part, there are two kinds of model predictive controller to be used. Due to the limitations such as horizons, restriction on the number of input and output variable to be supplied and so forth, the response of the neural network predictive controller is not at desired level. Thus, a model predictive with linearized model is designed for this study. A model predictive controller is established based on the model obtained with the neural network plant model. The summary of the neural network is shown by below graph.

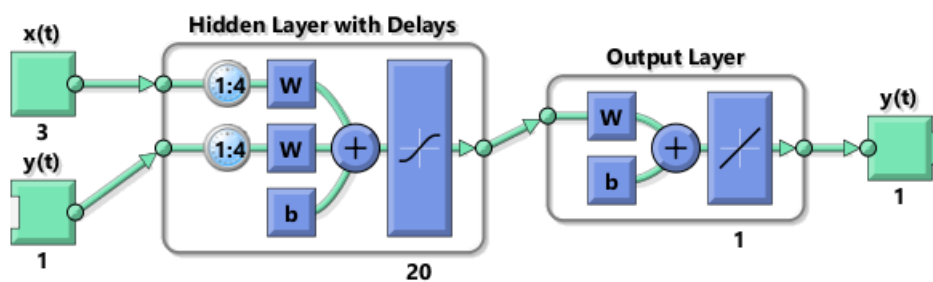


Figure 4.27 Graphical diagram representation of neural network

After obtaining the identified system for the model, the model is linearized. Linearization depends on the tangent linearization of nonlinear black-box model by which a first-order Taylor series approximation is obtained. The codes written to use the model predictive controller with the neural network is in Appendix.

Simulink blocks of the model predictive controller with neural network is demonstrated in Figure 4.22. There are three parts in the block diagram. The upper part is used for the model predictive controller design, the left bottom part is used for data supply and the right bottom part is the neural network with two layers.

The next step after MPC design is to use this MPC in simulink model of co-simulation plant. Figure 4.23 shows the co-simulation block diagram.

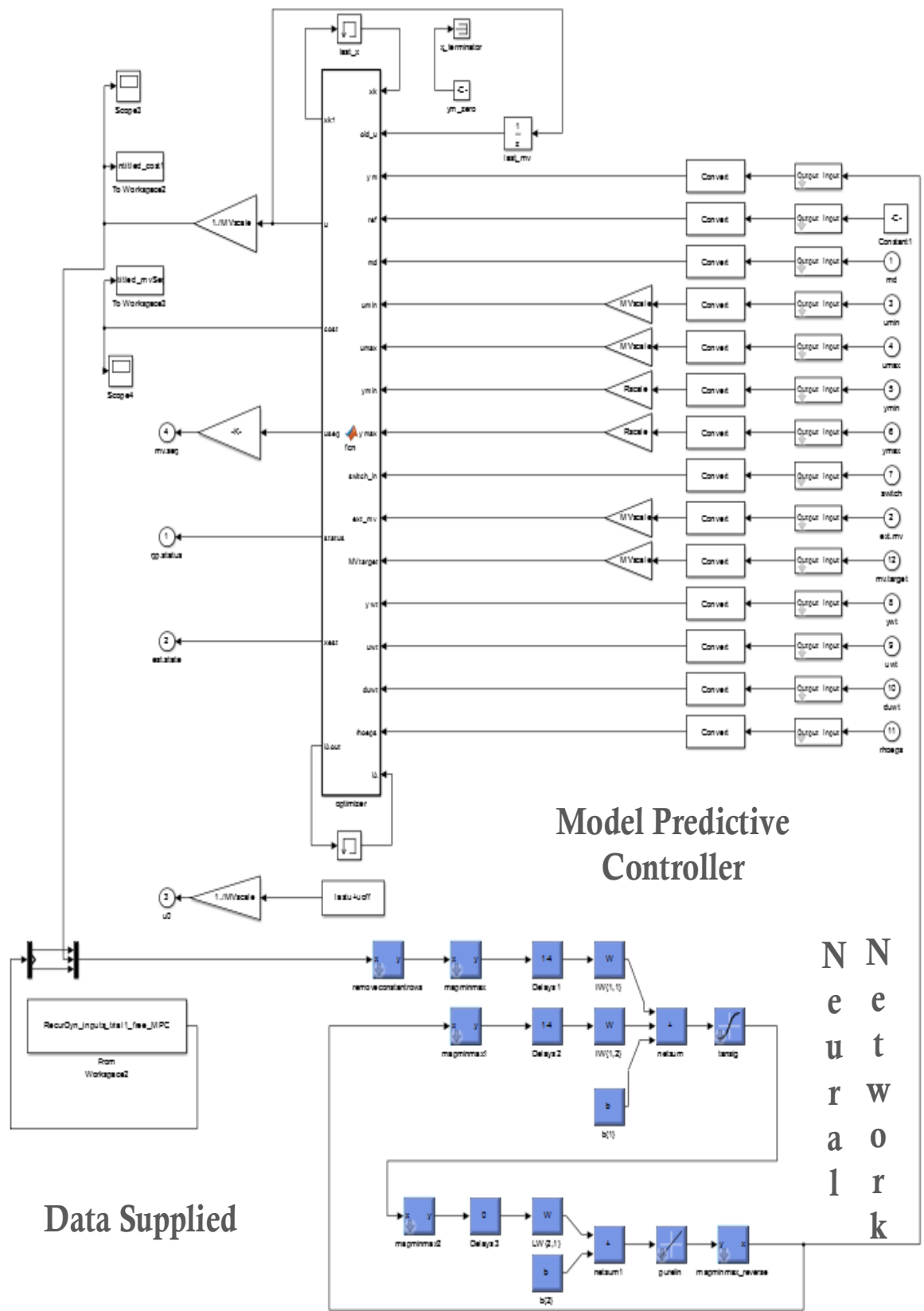


Figure 4.28 Simulink block for MPC design

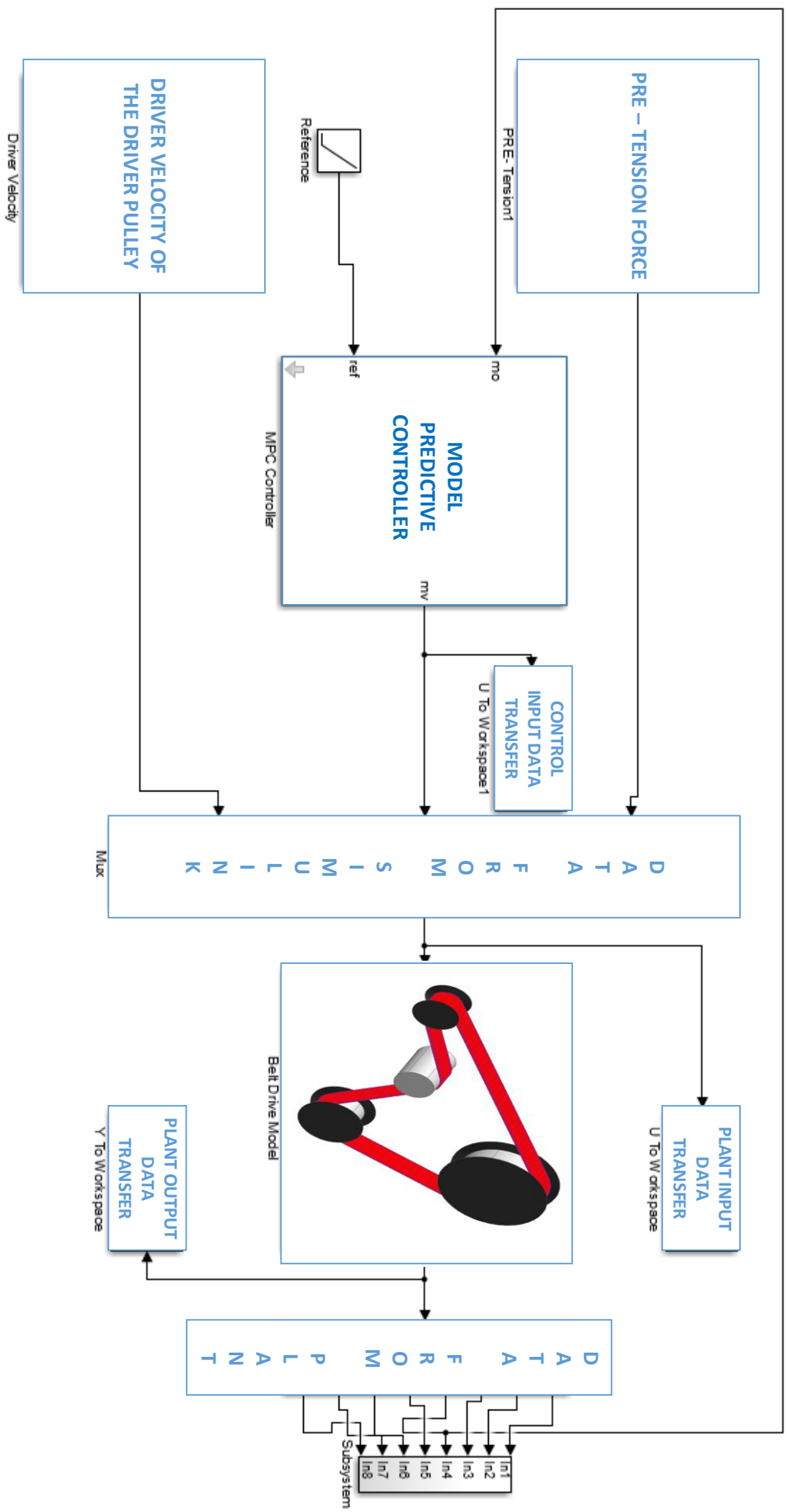


Figure 4.29 Co-simulation block diagram

CHAPTER 5: RESULTS and DISCUSSION

This chapter is the last chapter before the conclusion chapter and divided into four main categories.

5.1 Belt Drive System Behavior under Different Driver Velocity Inputs

A prototype belt drive system model is tested for impulse, ramp, sinusoidal and step inputs in this section. System is modeled with NARX neural networks initially, and then this model is used for an MPC. The aim of this section is to show the power of NARX networks rather than real life approximation.

5.1.1 Impulse Input

The first input used to test the artificial neural network is the impulse input. It is crucial to note that sinusoidal training tensioner force input is used in order to teach the system behavior to NARX artificial neural network in all driver velocity inputs. Sinusoidal waves' scanning the all predefined points of tensioner force range enable easy system identification of belt drive system.

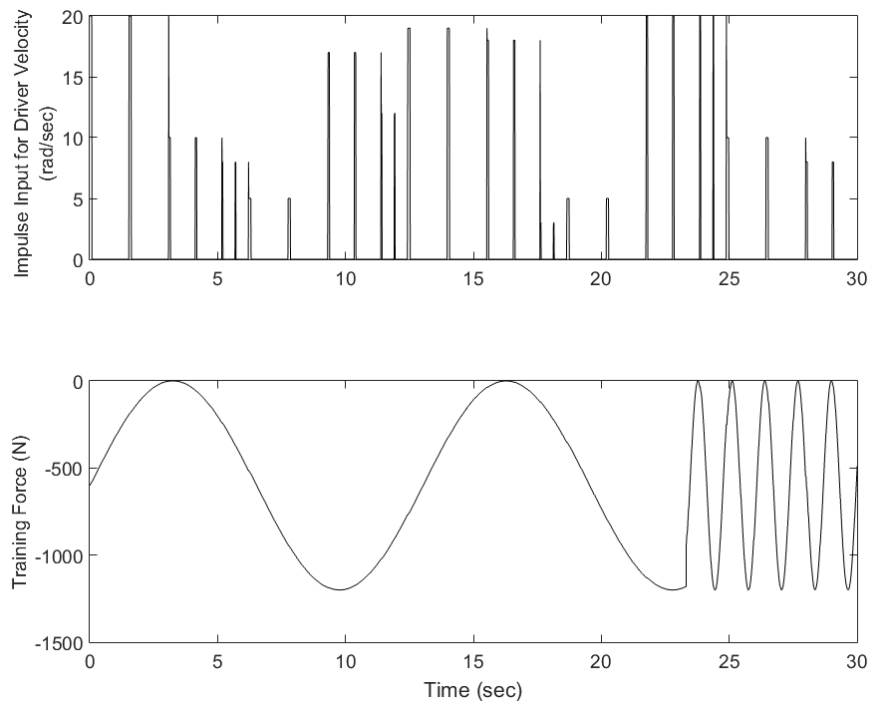


Figure 5.1 The training inputs for impulse driver velocity

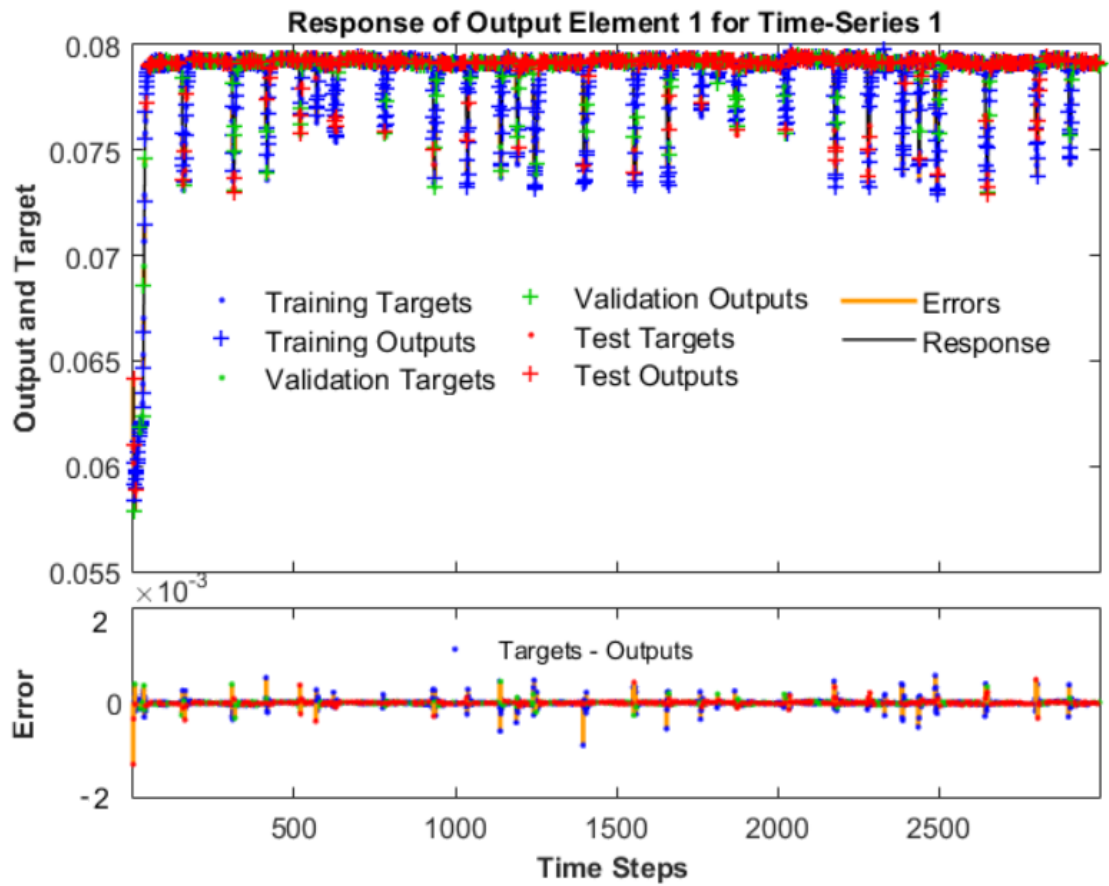


Figure 5.2. Narx responses for the impulse training inputs

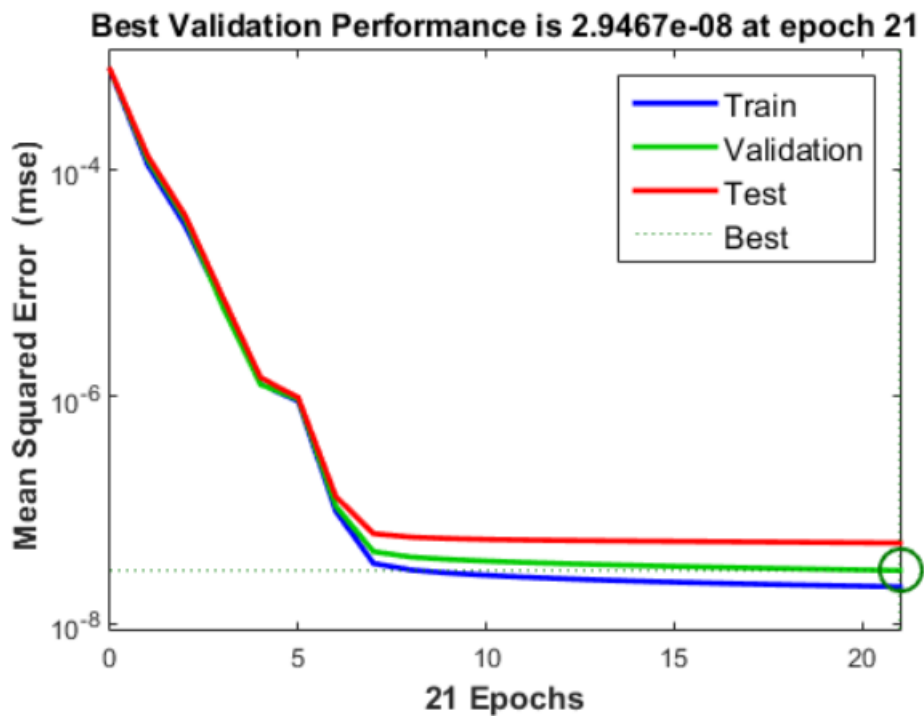


Figure 5.3. Narx performance for the impulse training inputs

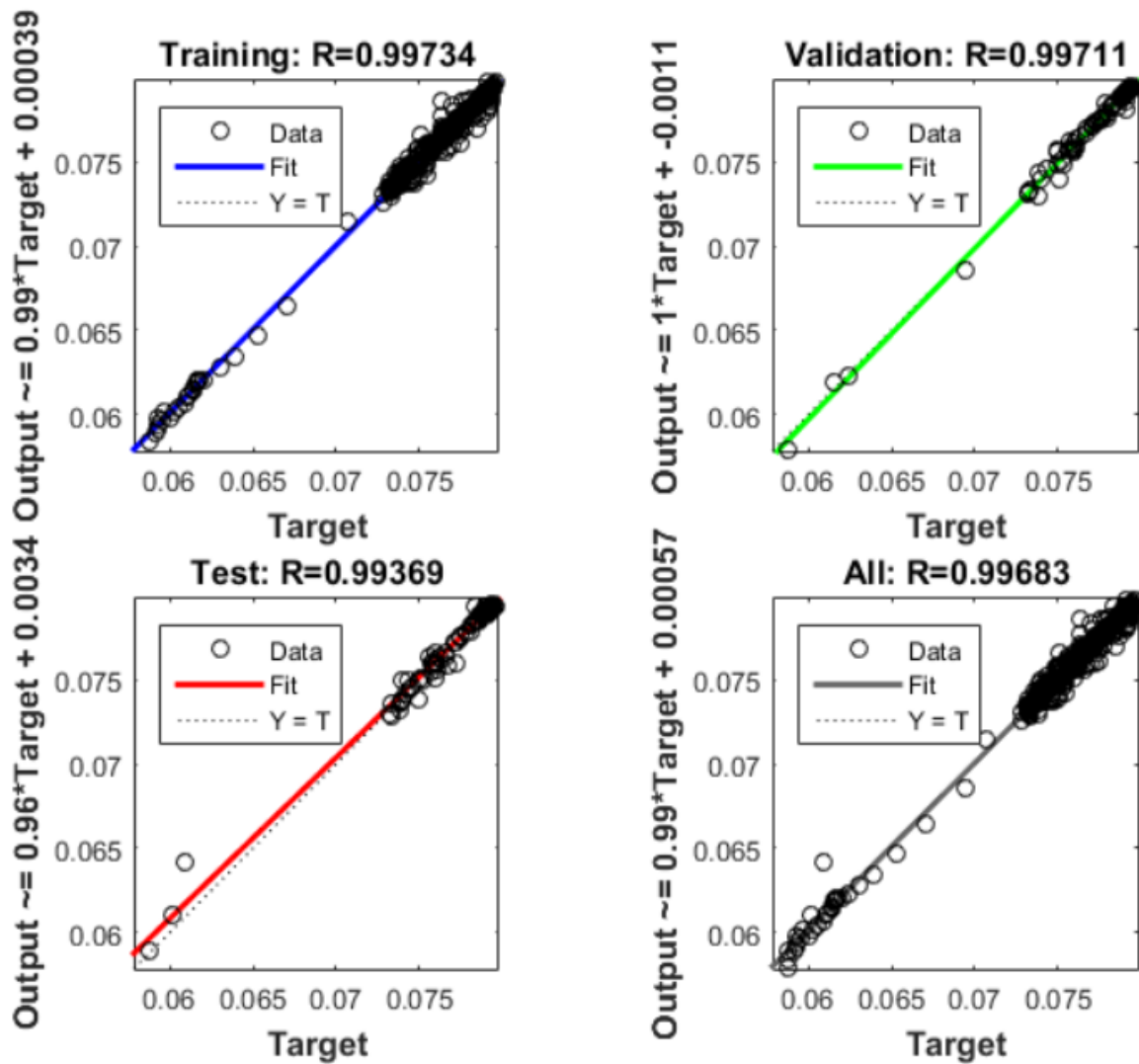


Figure 5.4. Narx regression for the impulse training inputs

Figure 5.2, Figure 5.3 and Figure 5.4 show the NARX related results for the impulse driver velocity. 70% percent of the data is used for training, 15% of data is used for test and 15% of data is used for validation data. As can be seen from Figure 5.2, NARX results are close enough to the system response to make a model. Figure 5.2 and 5.3 supports this idea with small mean square error which is about 10^{-8} and data's being around the fitting line respectively. Figure 5.5 shows the success of the NARX neural network for different inputs. Two layers are used with 20 neurons and 4 delay for this artificial neural network.

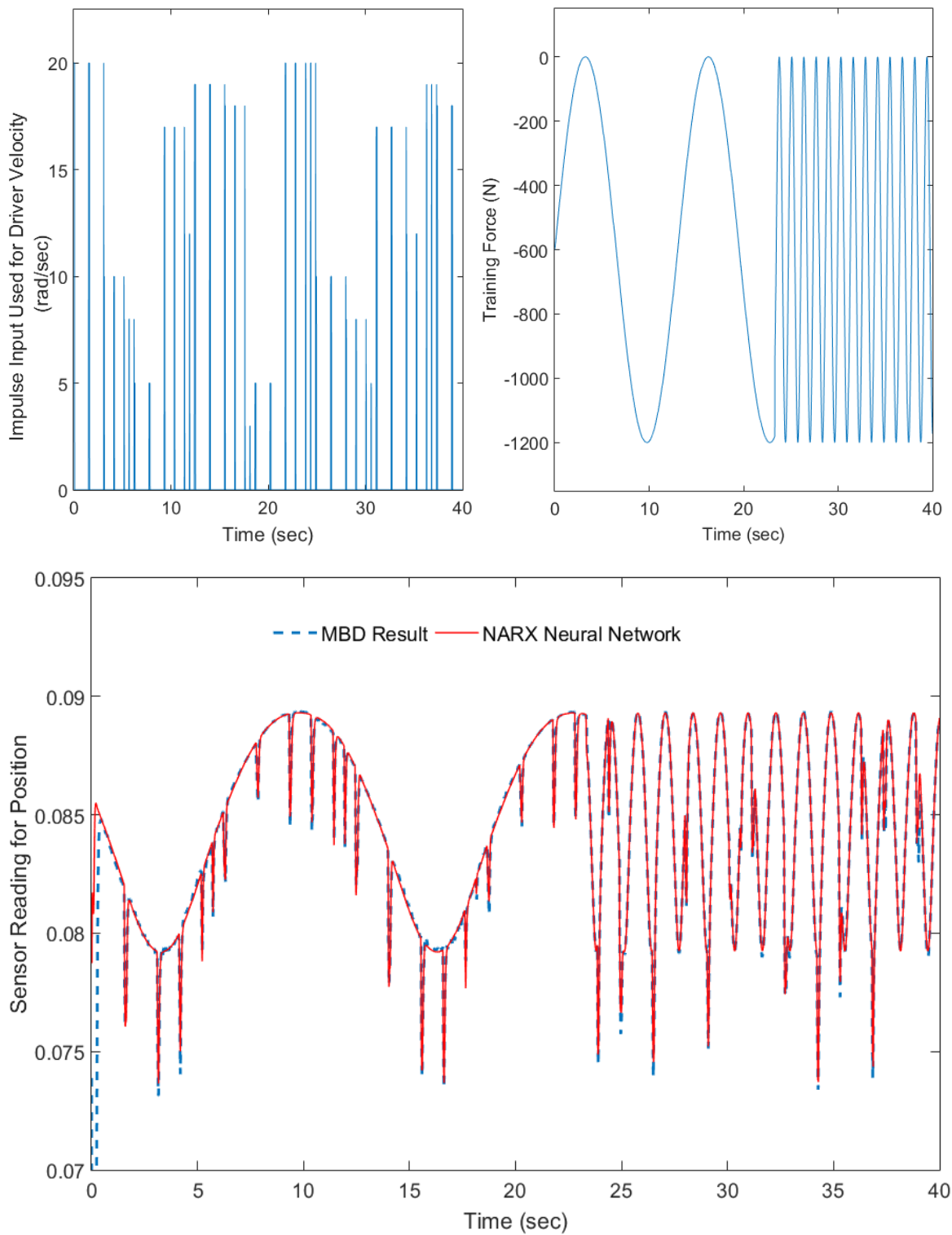


Figure 5.5 Narx confirmation for the extended training inputs

The above Figure is used to test the system behaviour in case of extended impulse driver velocity and tensioner force. The closeness of the NARX result is good enough for system approximation satisfaction. The above result is used to show the success of artificial neural network for extended training input. However, the system will be controlled in case of zero

tensioner force and the tensioner force will be applied according to the plant model. In other words, it is expected that the artificial neural network should work well enough in case of no tensioner force is applied because it can be desired to get a response such that no tensioner force is required. That is why the established NARX model is tested for the same but without tensioner force data in Figure 5.6.

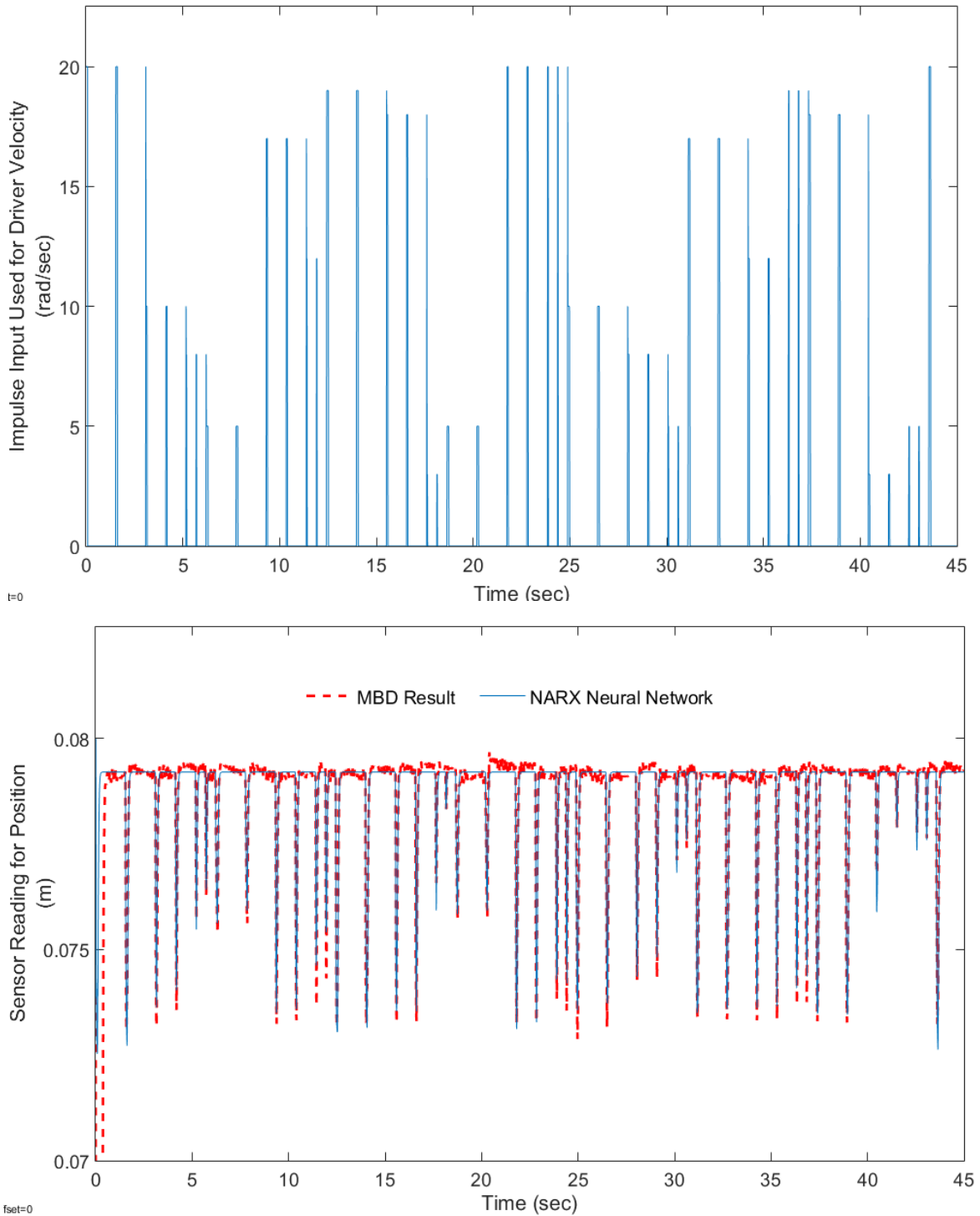


Figure 5.6 Narx model confirmation trained with impulse input in case of no tensioner force is applied

Impulse inputs different from the previous data set is used to test how the NARX model approximate the different impulse inputs in the same range.

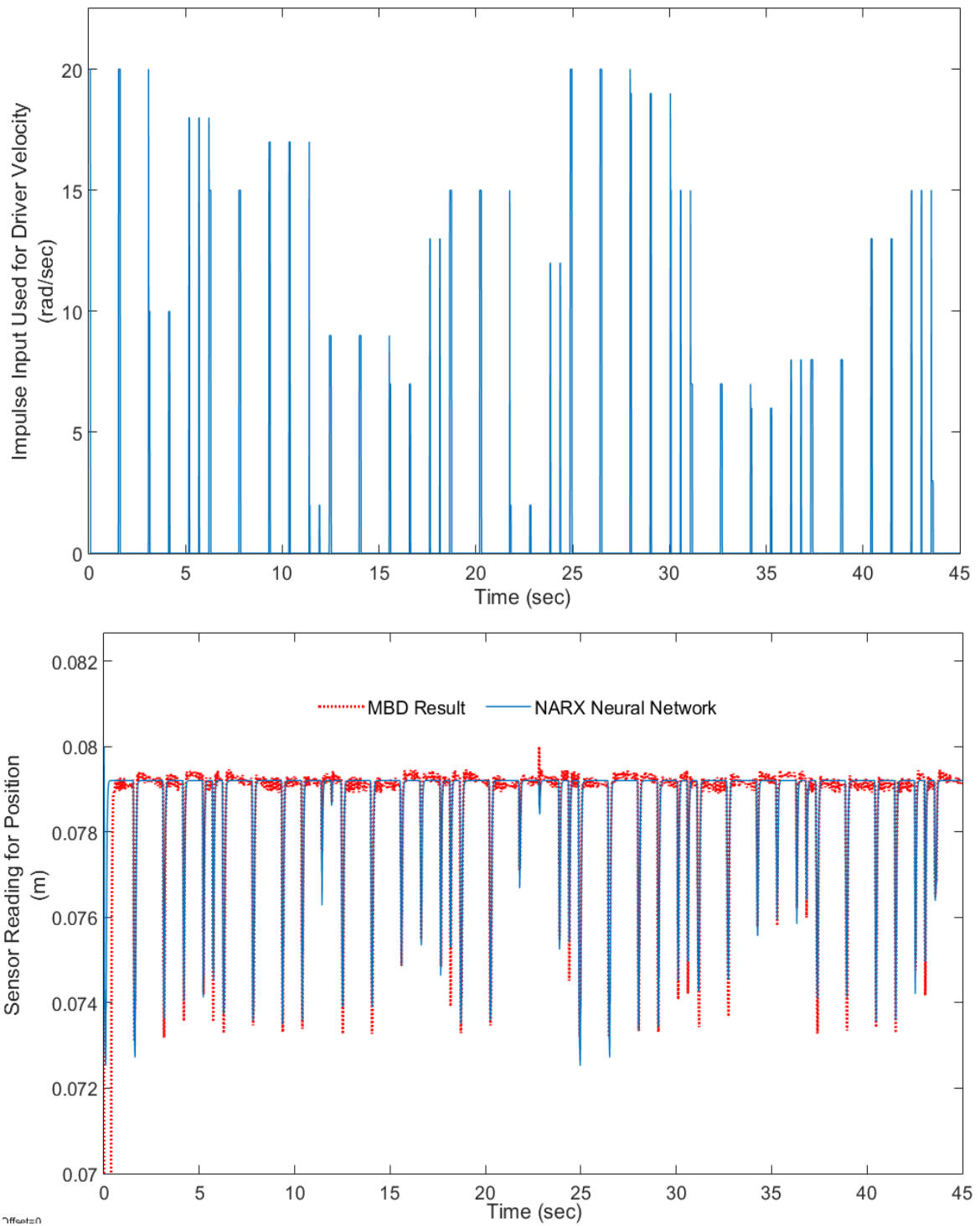


Figure 5.7 Narx model confirmation trained with impulse input for different driver velocity in case of no tensioner force is applied

5.1.2 Ramp Input

The second input type is ramp input. Figure 5.6 demonstrates the training inputs. The same procedure is applied to this type of input.

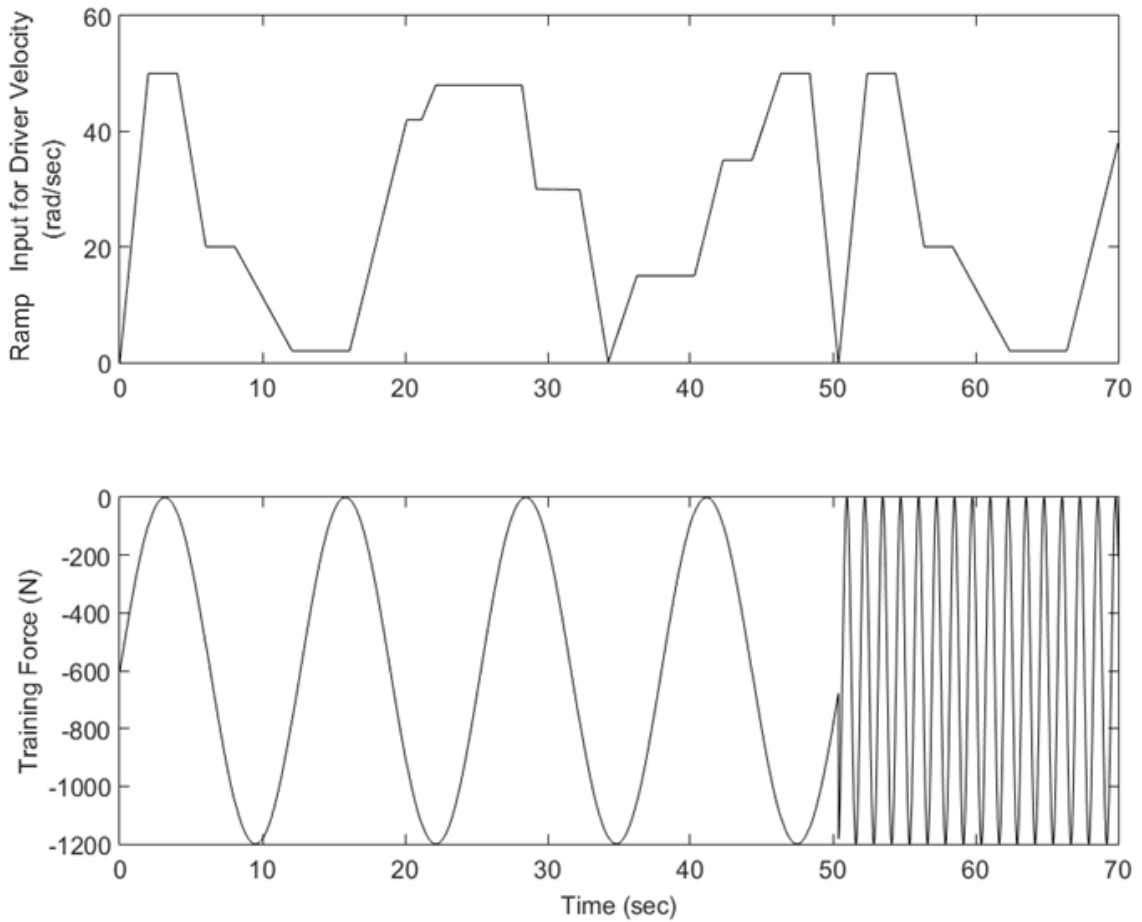


Figure 5.8 The training inputs for ramp driver velocity

Figure 5.7 shows the NARX responses for the above training data set. The same percentages mentioned in the impulse driver velocity is used for this and the remaining inputs. In addition, NARX delay, number of neurons and the transfer functions are same.

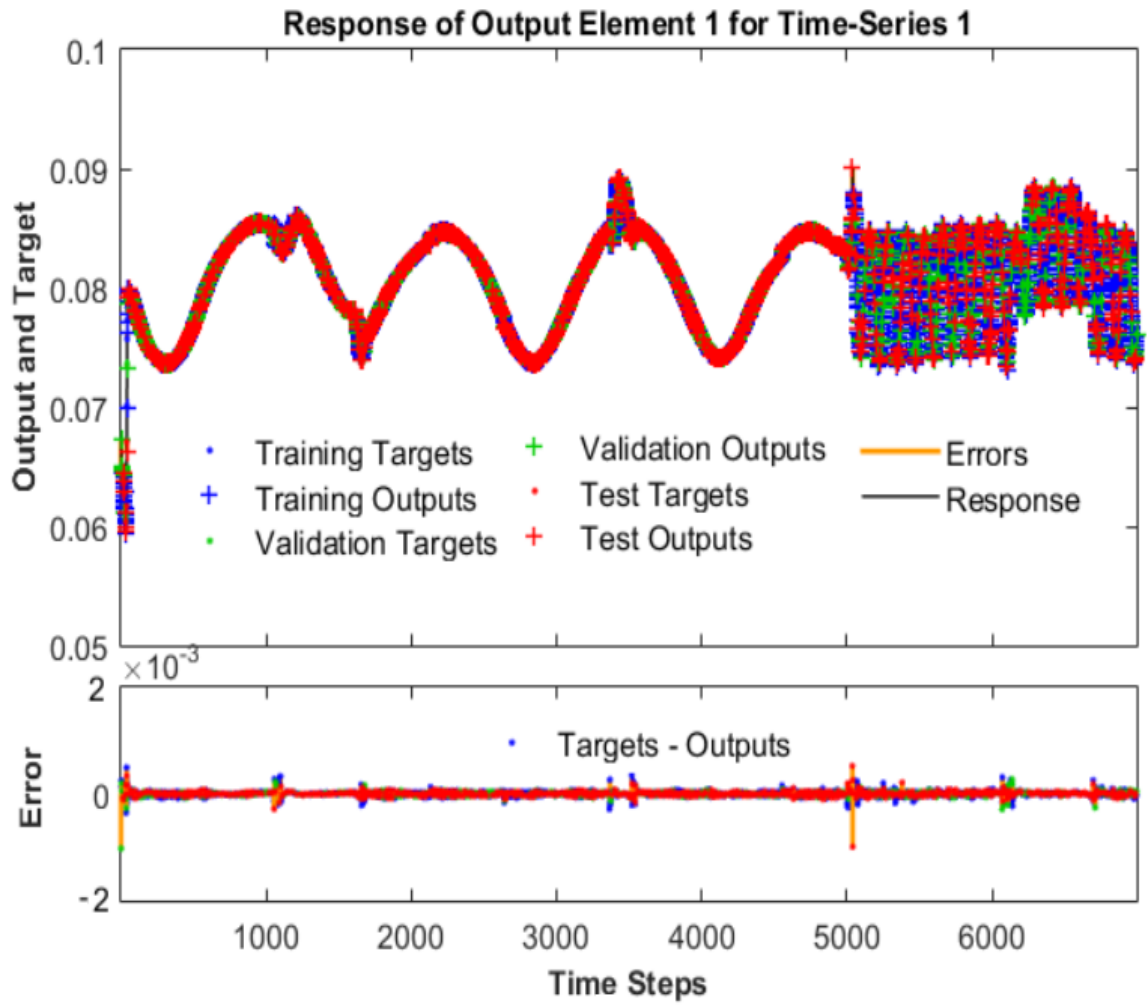


Figure 5.9 Narx responses for the ramp training inputs

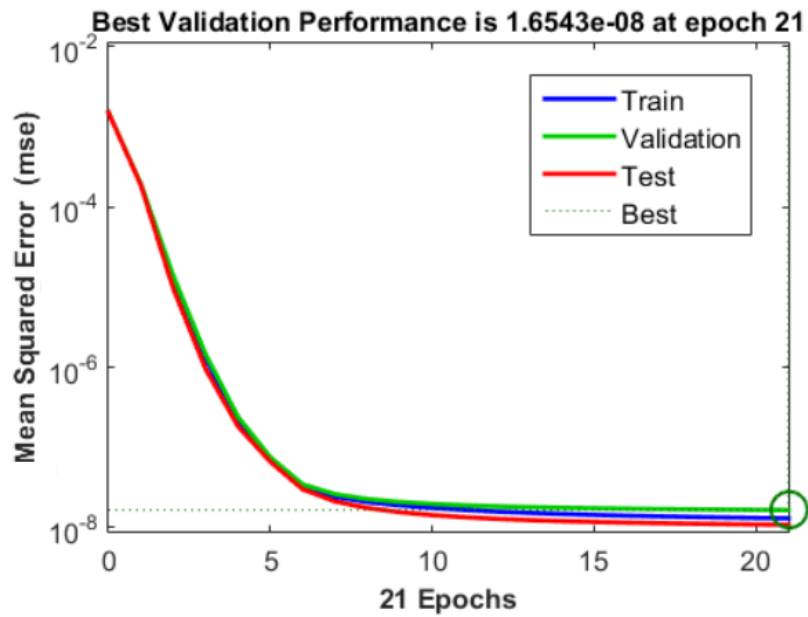


Figure 5.10 Narx performance results for the ramp training inputs

The NARX model results for training, test and validation data are shown in Figure 5.9. Obtaining a perfect fit is not possible; therefore, the error shown below the responses are accepted to model satisfaction. The mean square errors for training, validation and test data are as shown in Figure 5.10. The best validation performance value decreases to 1.6534×10^{-8} . The convergence of the error just a little above $y=10^{-8}$ line is a good indication for the belt drive system model fit.

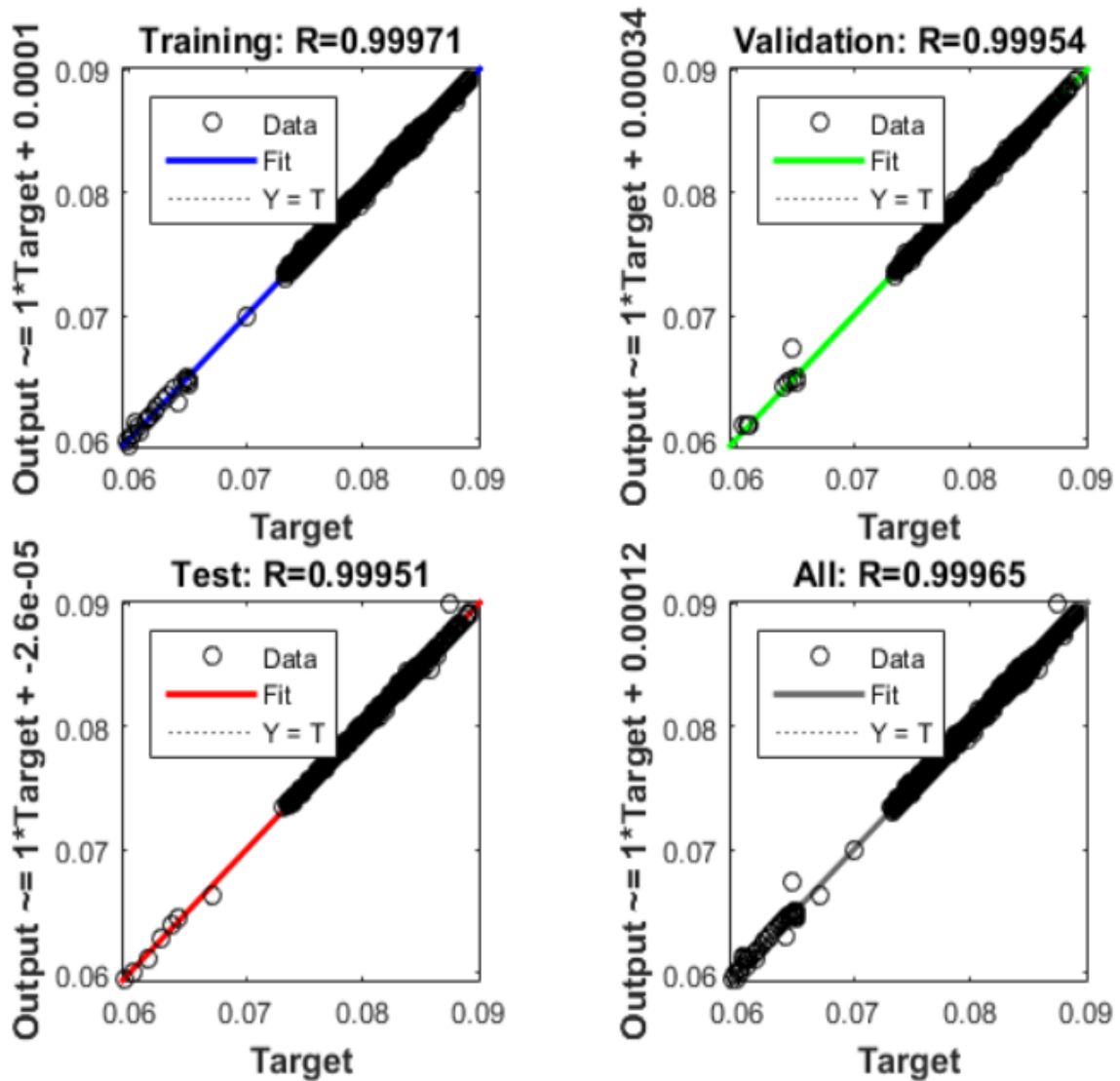


Figure 5.11. Narx regression results for the ramp training inputs

The above figure shows the regression plots for training, validation, test data and combination of all. The closeness of R value to zero and data to fit line are the two critical indicators for the success of regression and it is obvious that the NARX model is a good approximation for the ramp input.

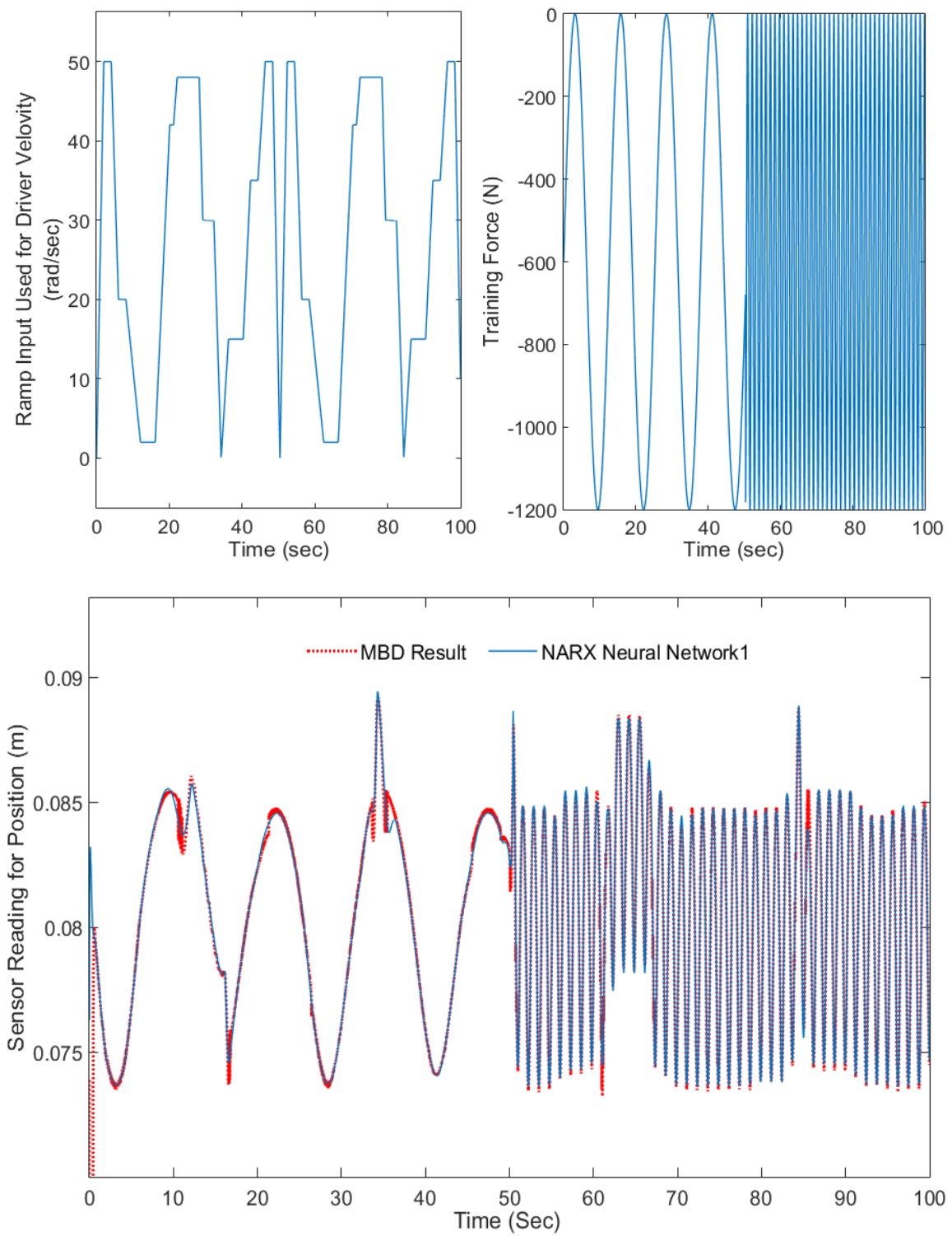


Figure 5.12 Narx confirmation for the extended training inputs

The above figure shows how this NARX model behaves under extended training data. The same data except for the tensioner training force tested and as can be from the Figure 5.13 the results of MBD and NARX are close. A different driver velocity data is tested for the same purpose in the below figure.

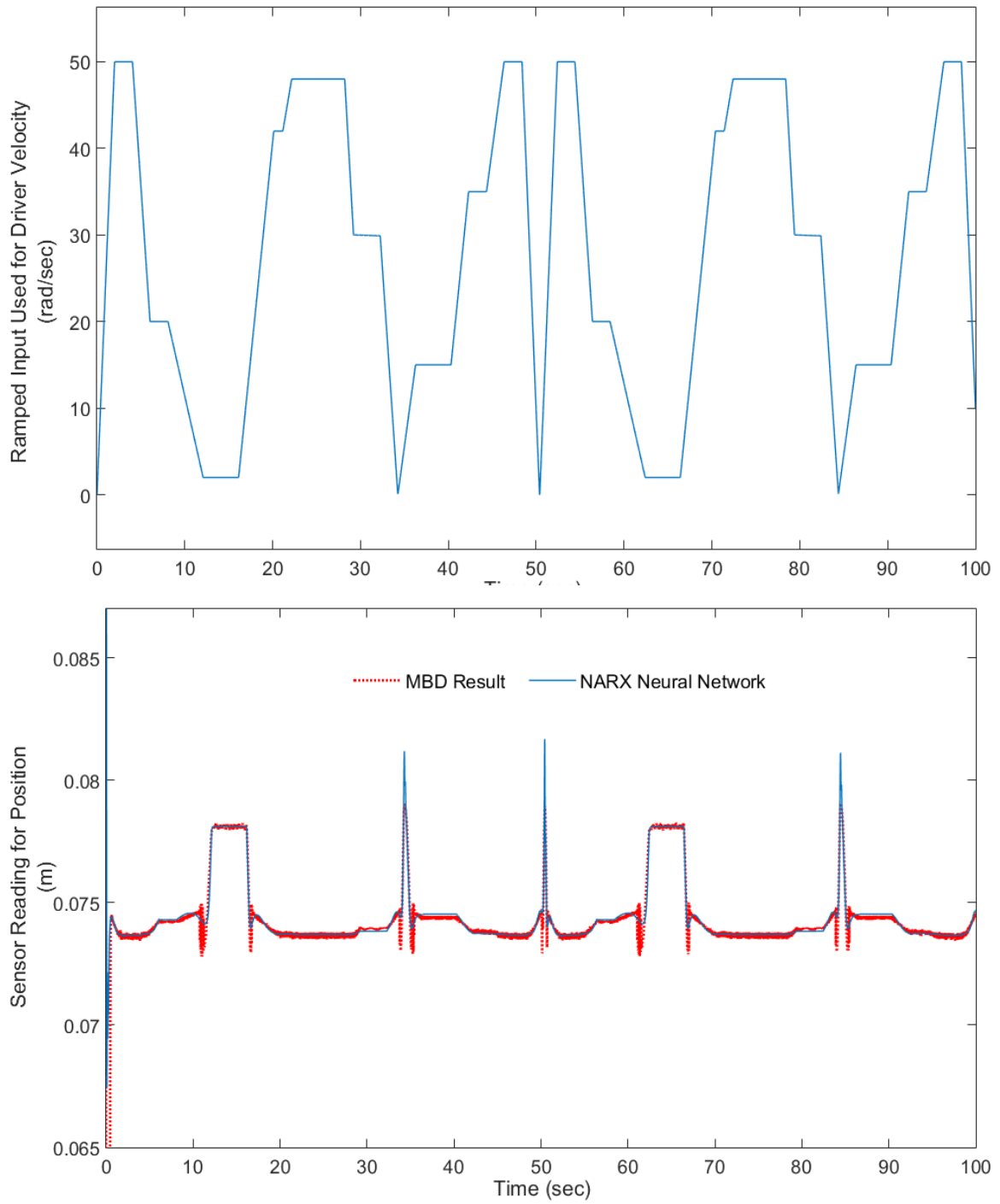


Figure 5.13 Narx model confirmation trained with ramp input in case of no tensioner force is applied

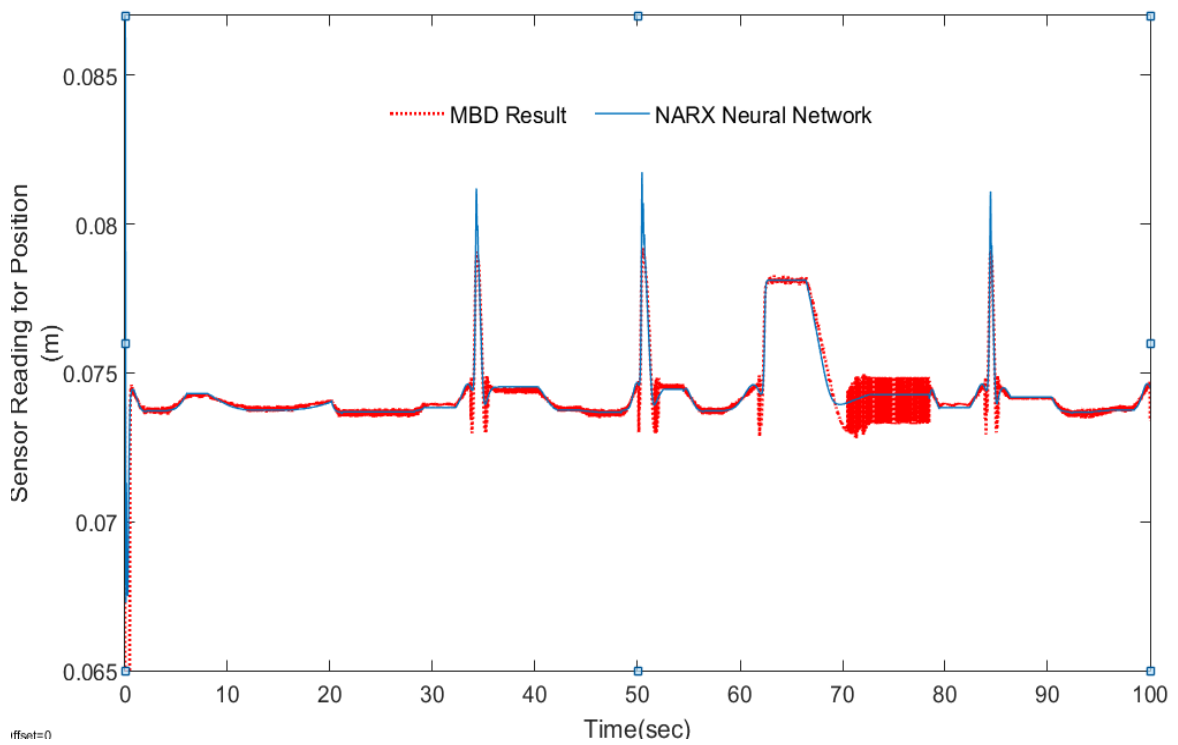
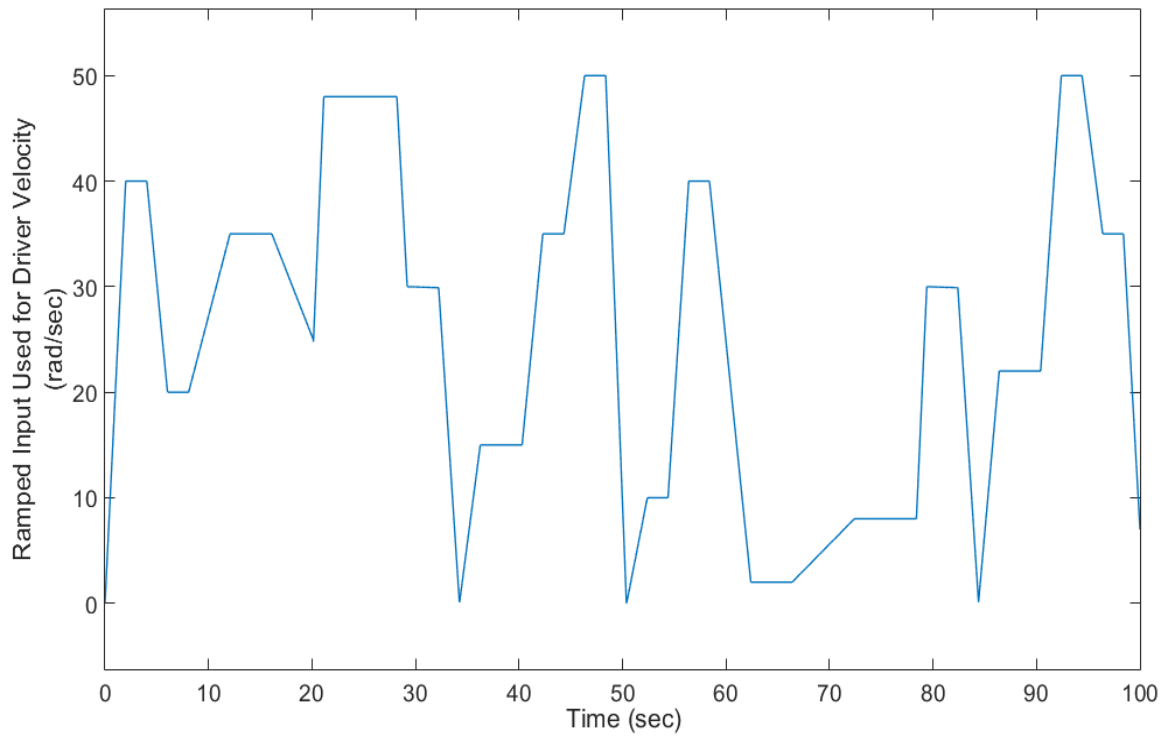


Figure 5.14 Narx model confirmation trained with ramp input for different driver velocity in case of no tensioner force is applied

5.1.3 Sinusoidal Input

The third input data type is sinusoidal input. Sinusoidal input is tested for different frequencies as can be seen from the below figure.

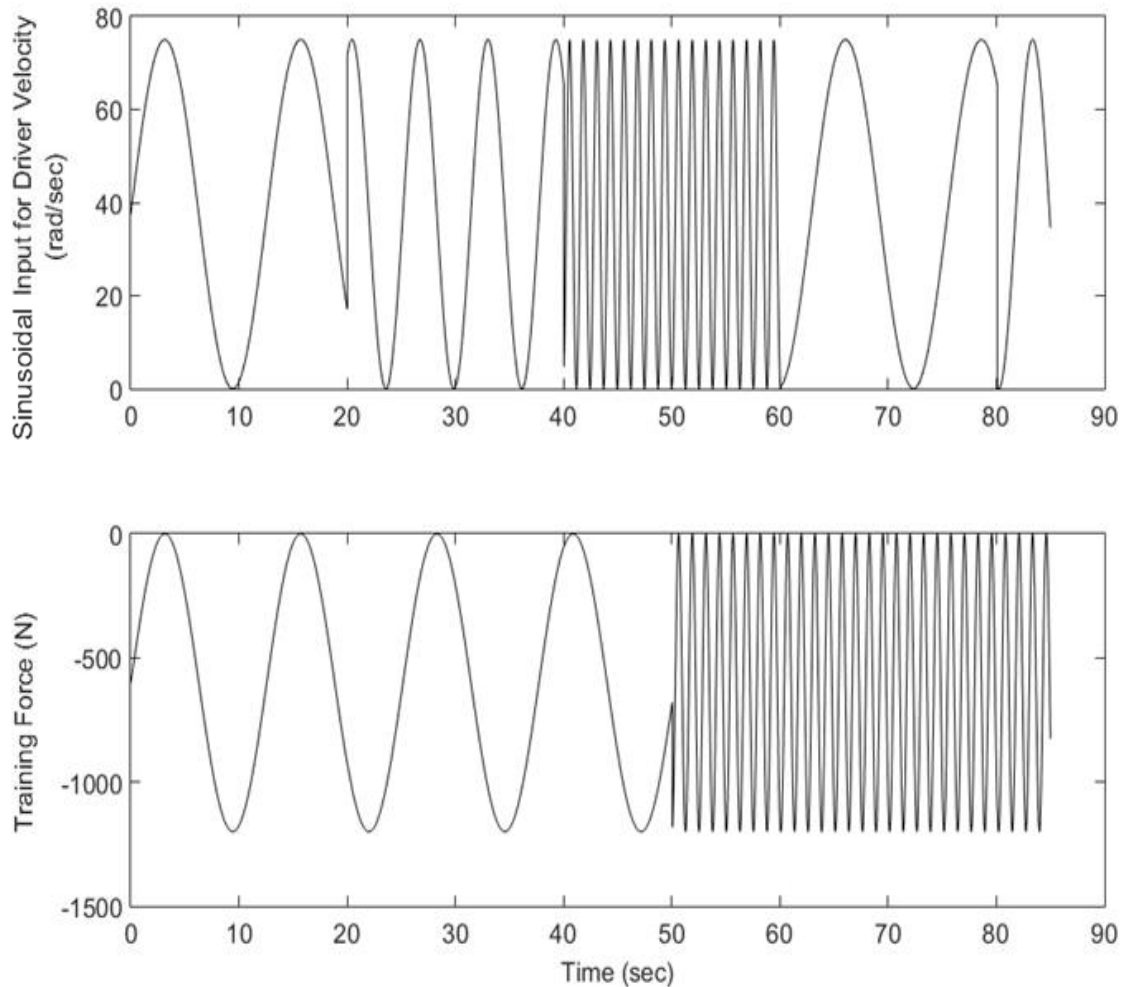


Figure 5.15 The training inputs for sinusoidal driver velocity

The dominance of the tensioner force over the driver velocity input is seen from the figure below. However, the effect of the driver velocity on the system response is more clear in this figure compared to the ramp input. This effect is smaller compared to the impulse input. The reason behind this behaviour is that whereas the sudden changes in the driver velocity has strong effect on the belt behaviour due to its strong tensioning effect, the soft transitions in the driver velocity has not that much effect. Although it is possible to obtain better results by changing the parameters used in this artificial neural network, the closeness of NARX model to MBD model in Figure 5.16 is accepted as satisfactory.

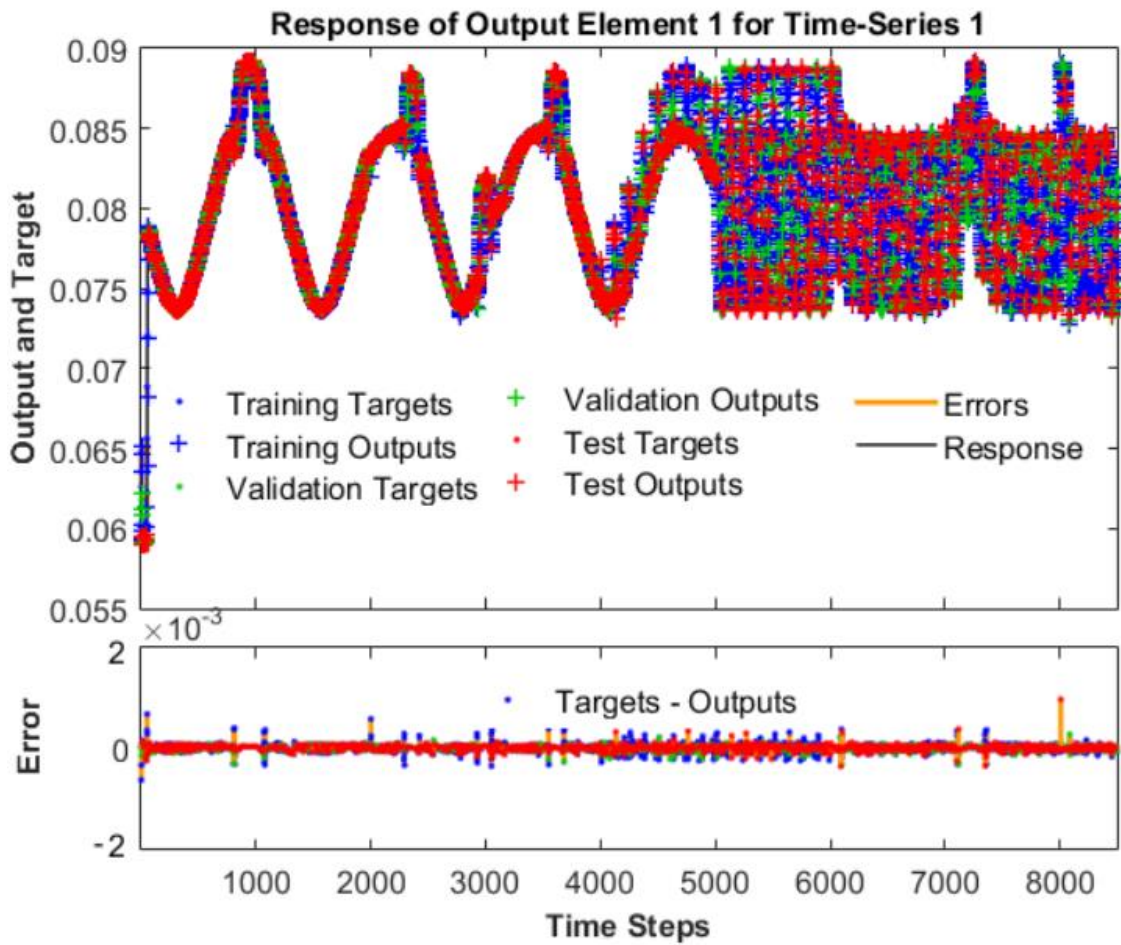


Figure 5.16 Narx responses for the sinusoidal training inputs

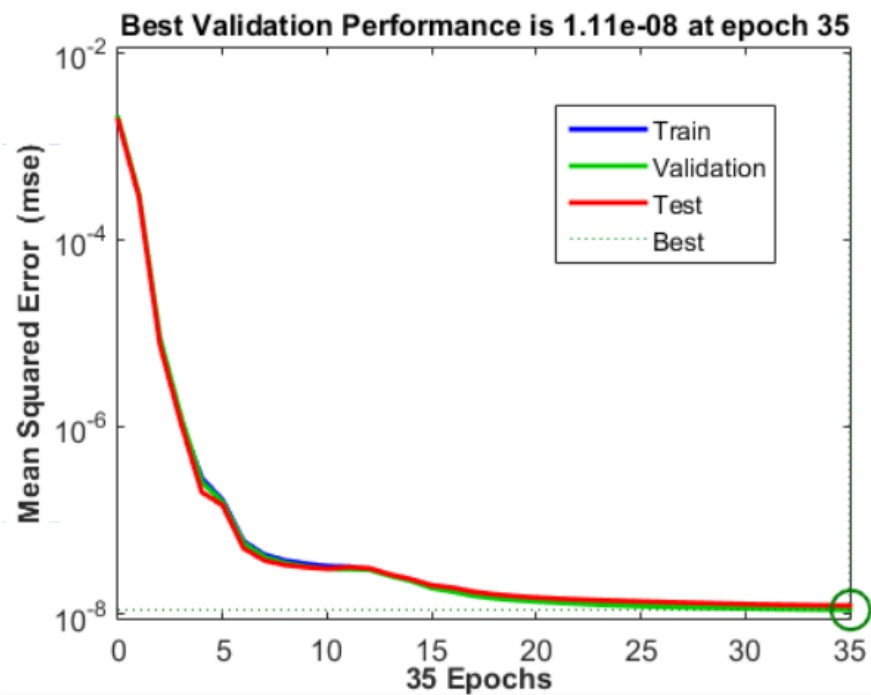


Figure 5.17 Narx performance for the sinusoidal training inputs

The performance values of the NARX model is plotted in Figure 5.17. Training, validation and test data mean square errors converges near to $y=10^{-8}$ line.

Moreover, regression plots in Figure 5.18 supports this idea as fit line pass from almost all of the data circles centers. The R value for all data sets; i.e., training, validation, test and combination sets, are all bigger than 0.999 value.

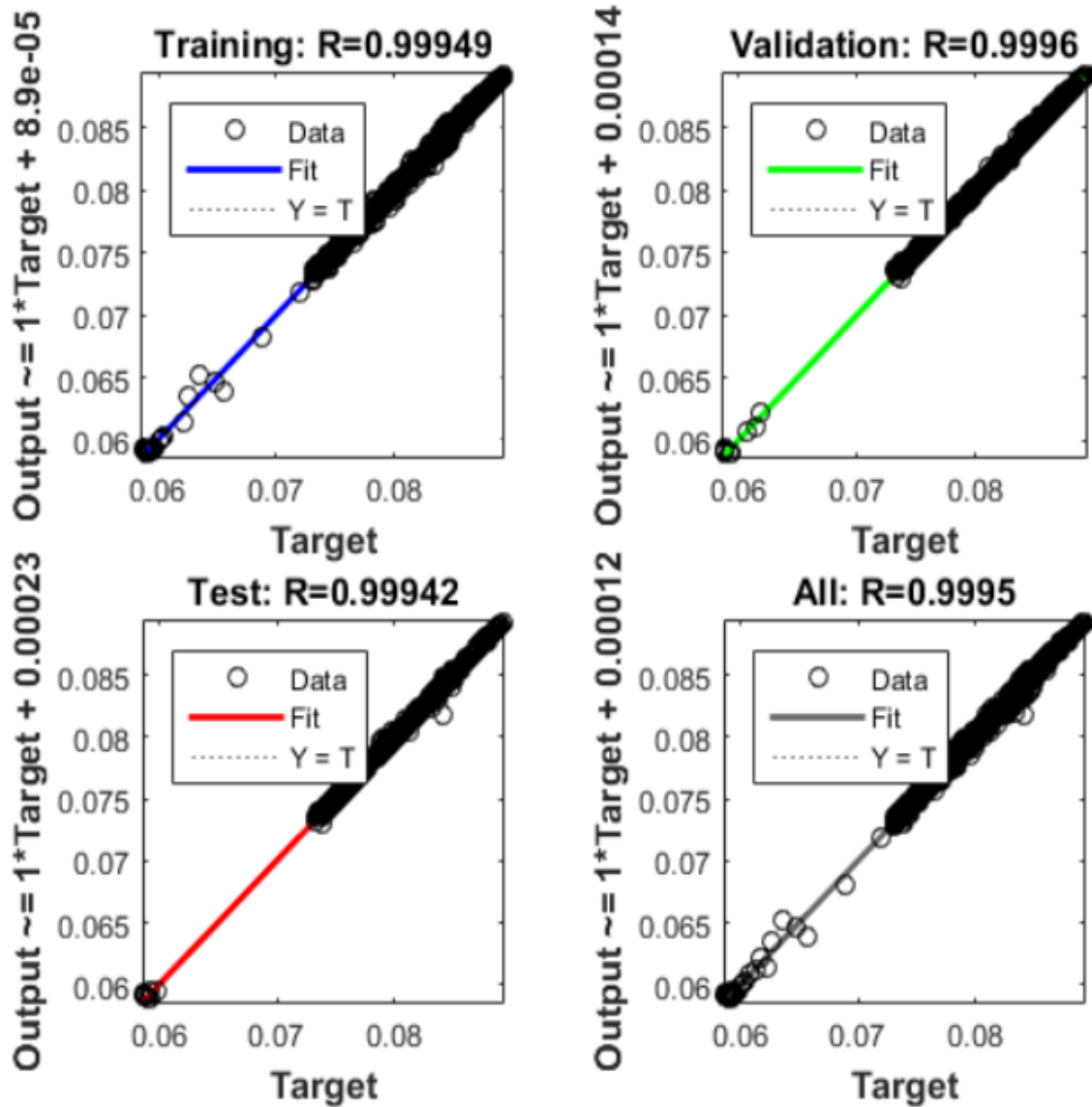


Figure 5.18 Narx regression for the sinusoidal training inputs

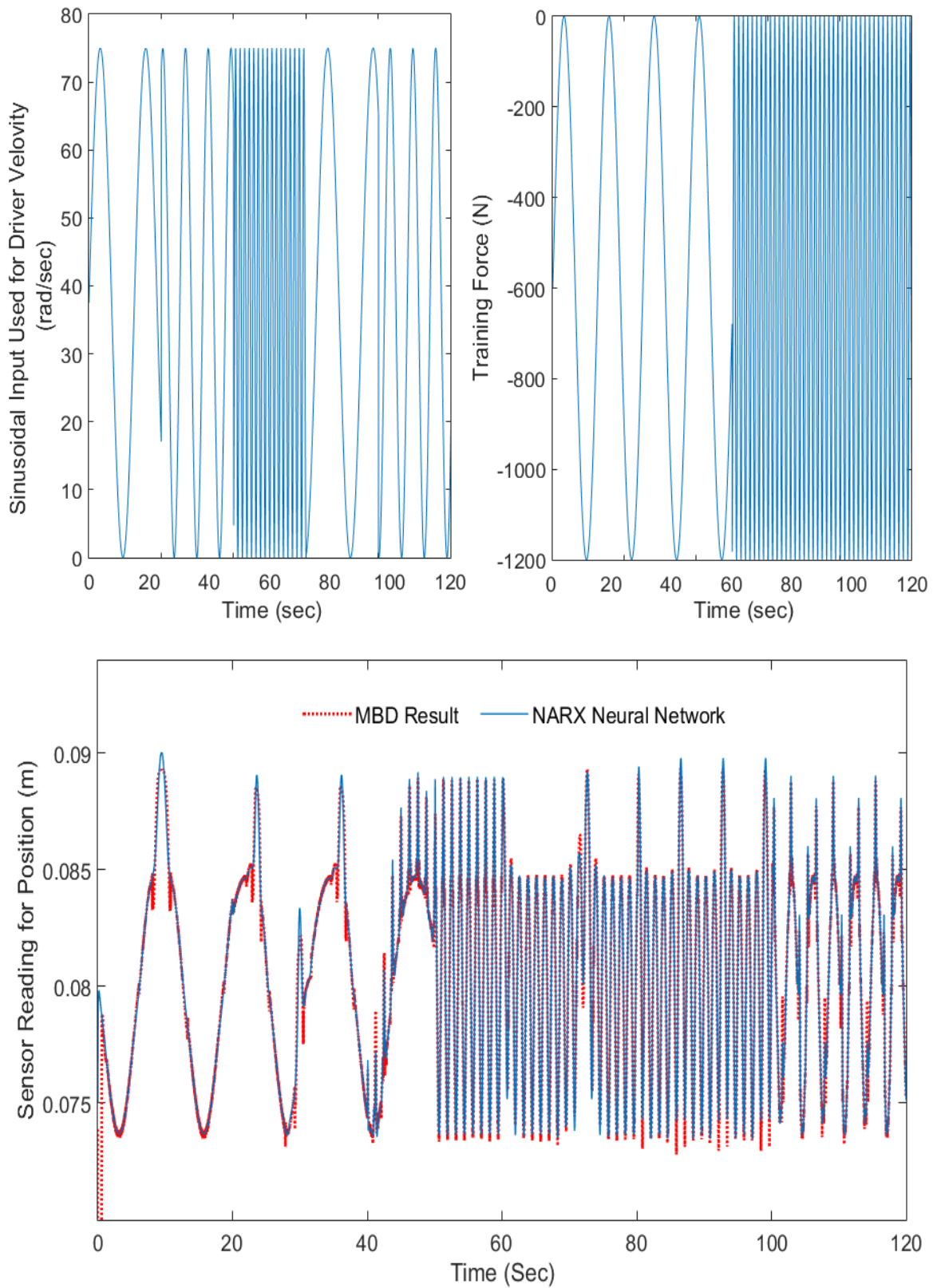


Figure 5.19 Narx confirmation for the extended training inputs

Above figure reveals how the NARX model approximate the MBD result in case of extended training inputs are used. Since the training tensioner force has three different frequencies,

the results are changing accordingly. The effect of driver velocity becomes more obvious when the tensioner force reaches its peak values especially. The weaker effect of the driver velocity compared to the tensioner force is expected because the tensioner force will be the control input.

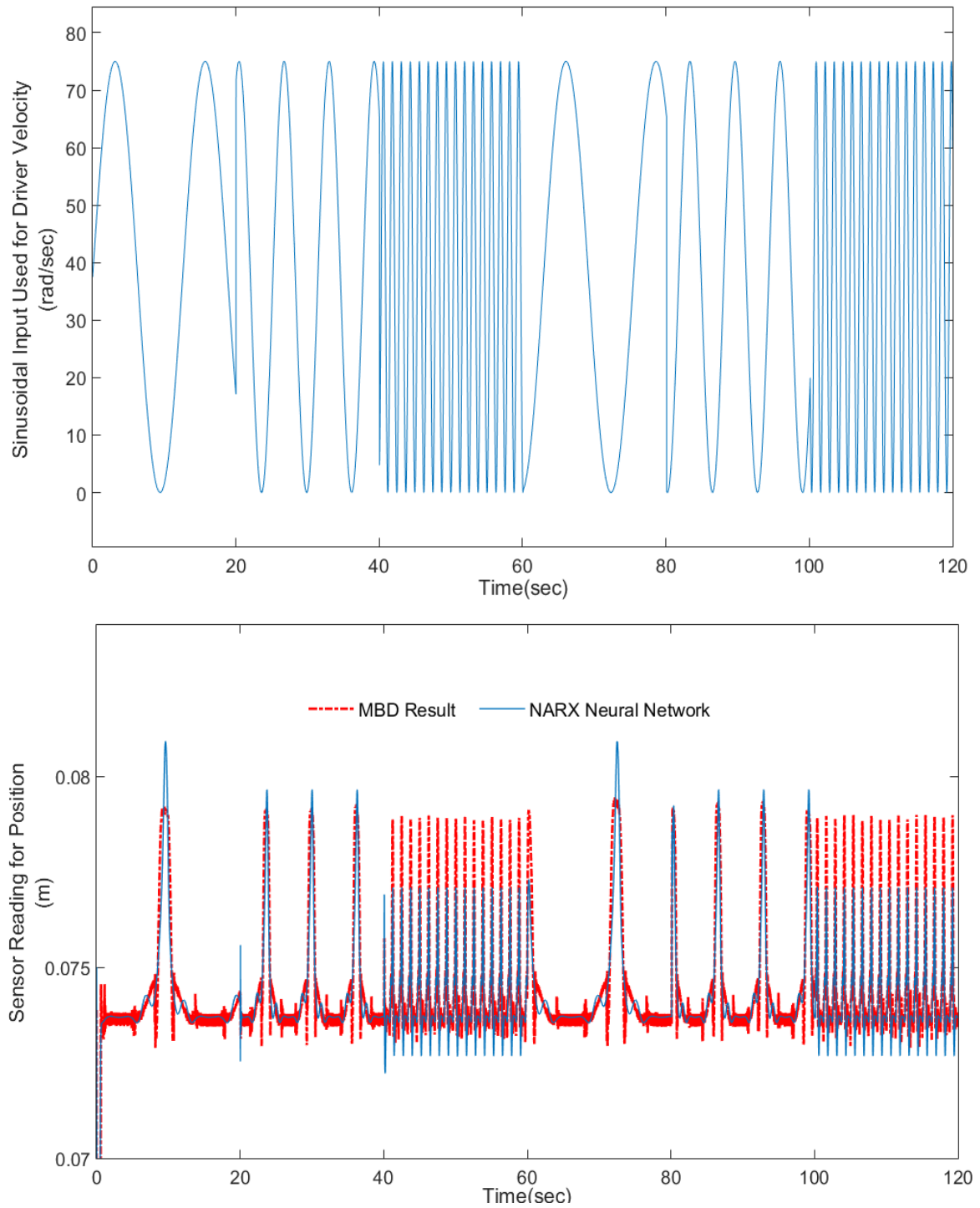


Figure 5.20 Narx model confirmation trained with sinusoidal input in case of no tensioner force is applied

Figure 5.20 and 5.21 are used to observe whether the similar effects occur at different sinusoidal driver velocities when system is free of tensioner force.

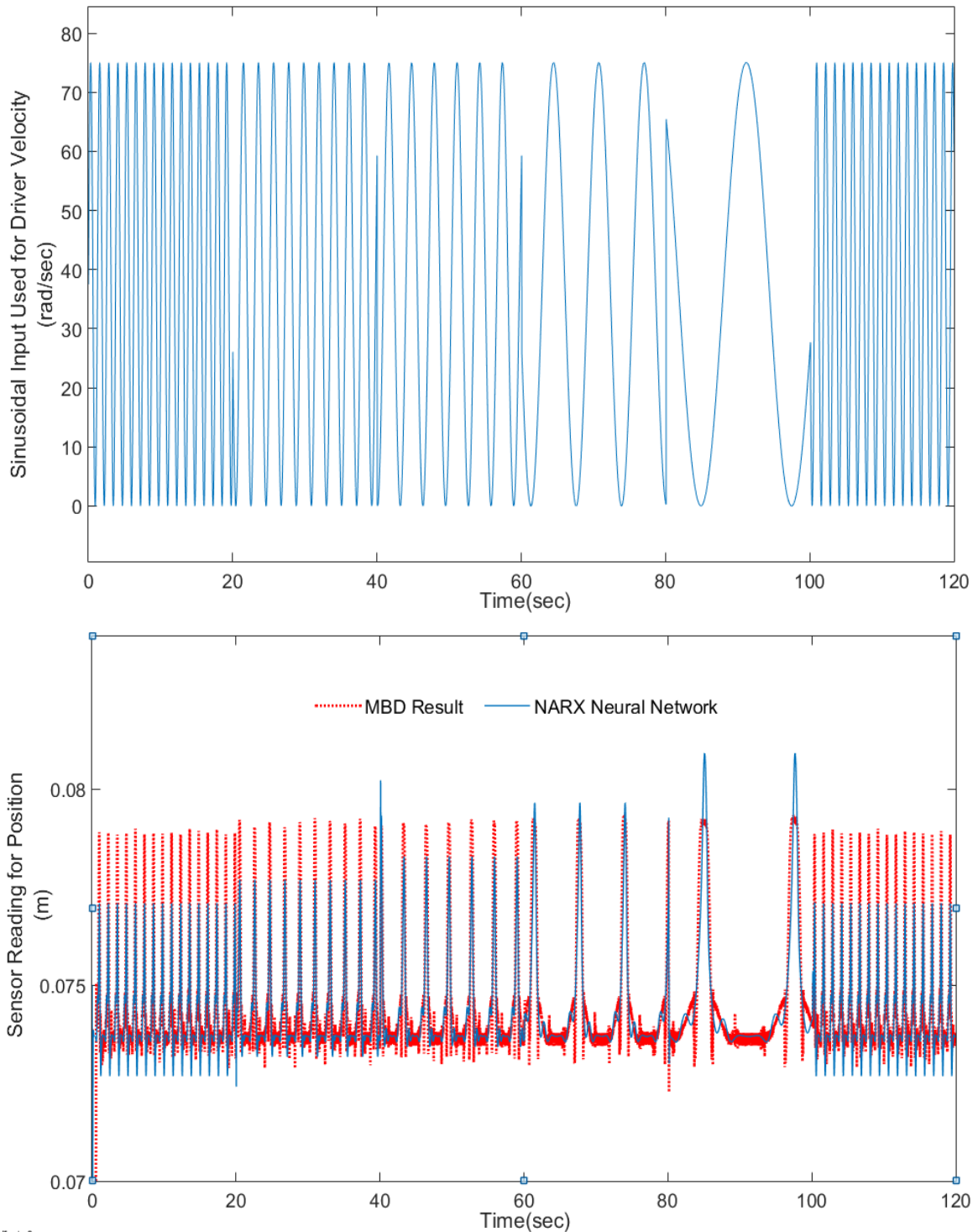


Figure 5.21 Narx model confirmation trained with sinusoidal input for different driver velocity in case of no tensioner force is applied

5.1.4 Step Input

The training inputs used for the step input velocity profile are given in the below figure.

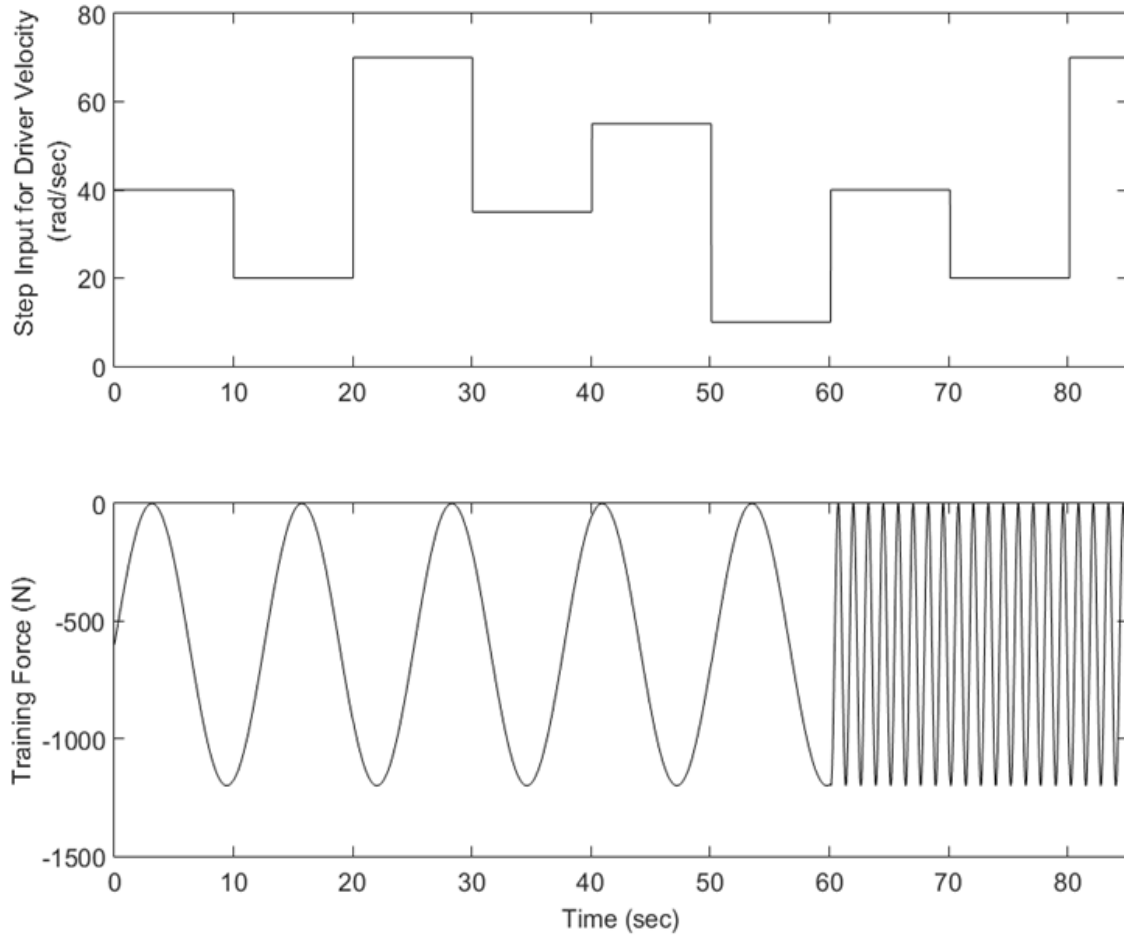


Figure 5.22 The training inputs for step driver velocity

Figure 5.23 shows the outputs of NARX model for the data plotted in the above figure. The performance value of the artificial neural network decrease to 1.422×10^{-8} mean square error for validation. The training and the test errors are also close to 10^{-8} value.

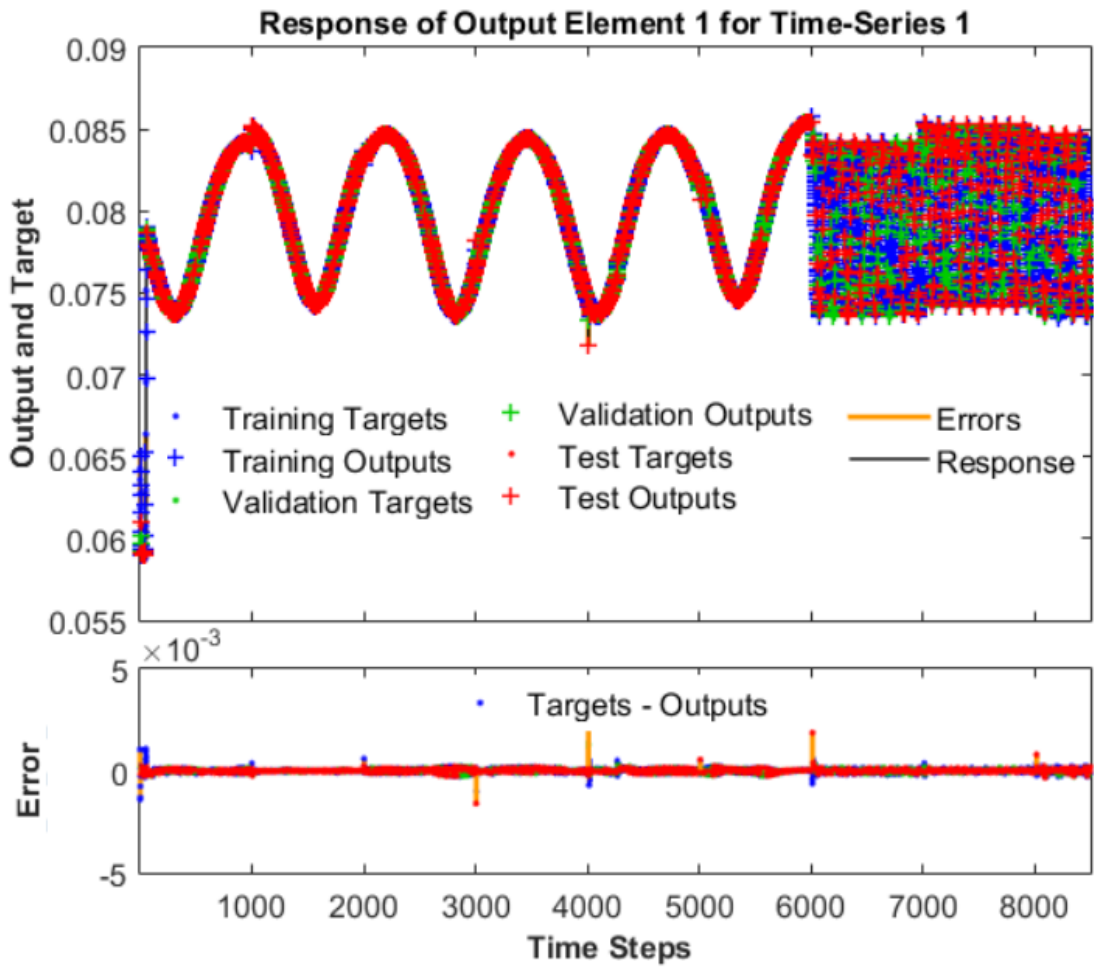


Figure 5.23 Narx responses for the step training inputs

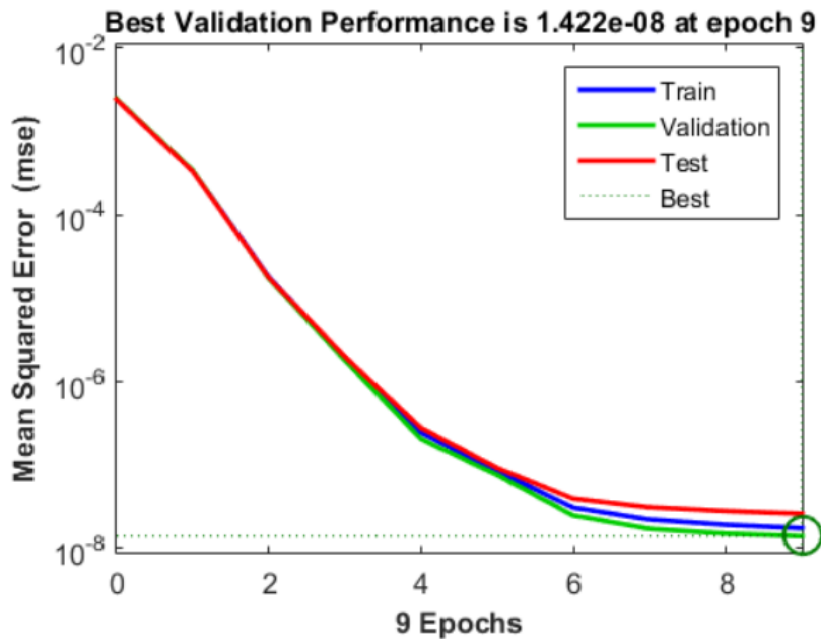


Figure 5.24 Narx performance for the step training inputs

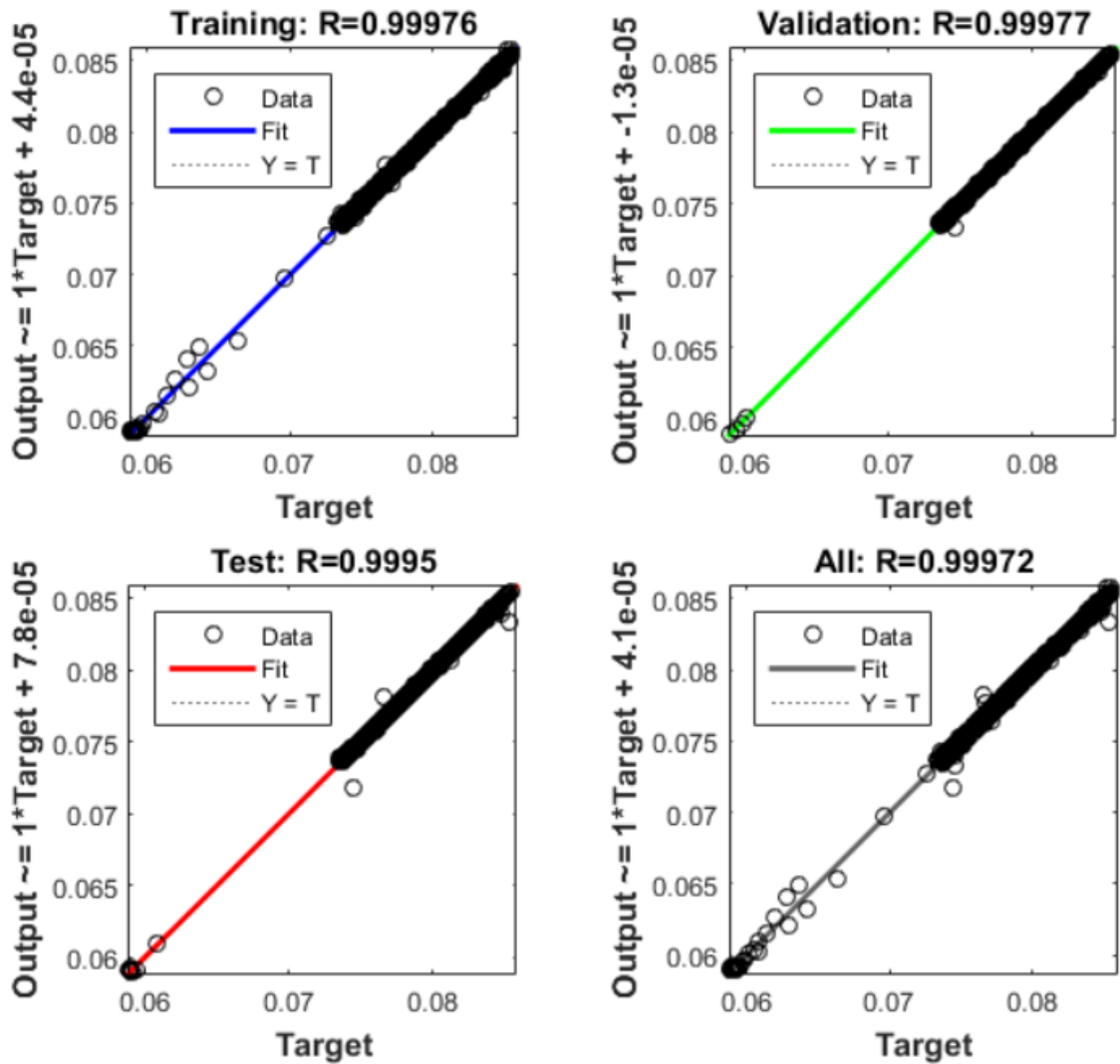


Figure 5.25 Narx regression for the step training inputs

Another verification for the model, is using the regression plots. According to Figure 5.25 , data circles are either on or close to the fitting line. Moreover, R value for training, validation, test sets and combination for all of the sets are bigger than 0.9995 value.

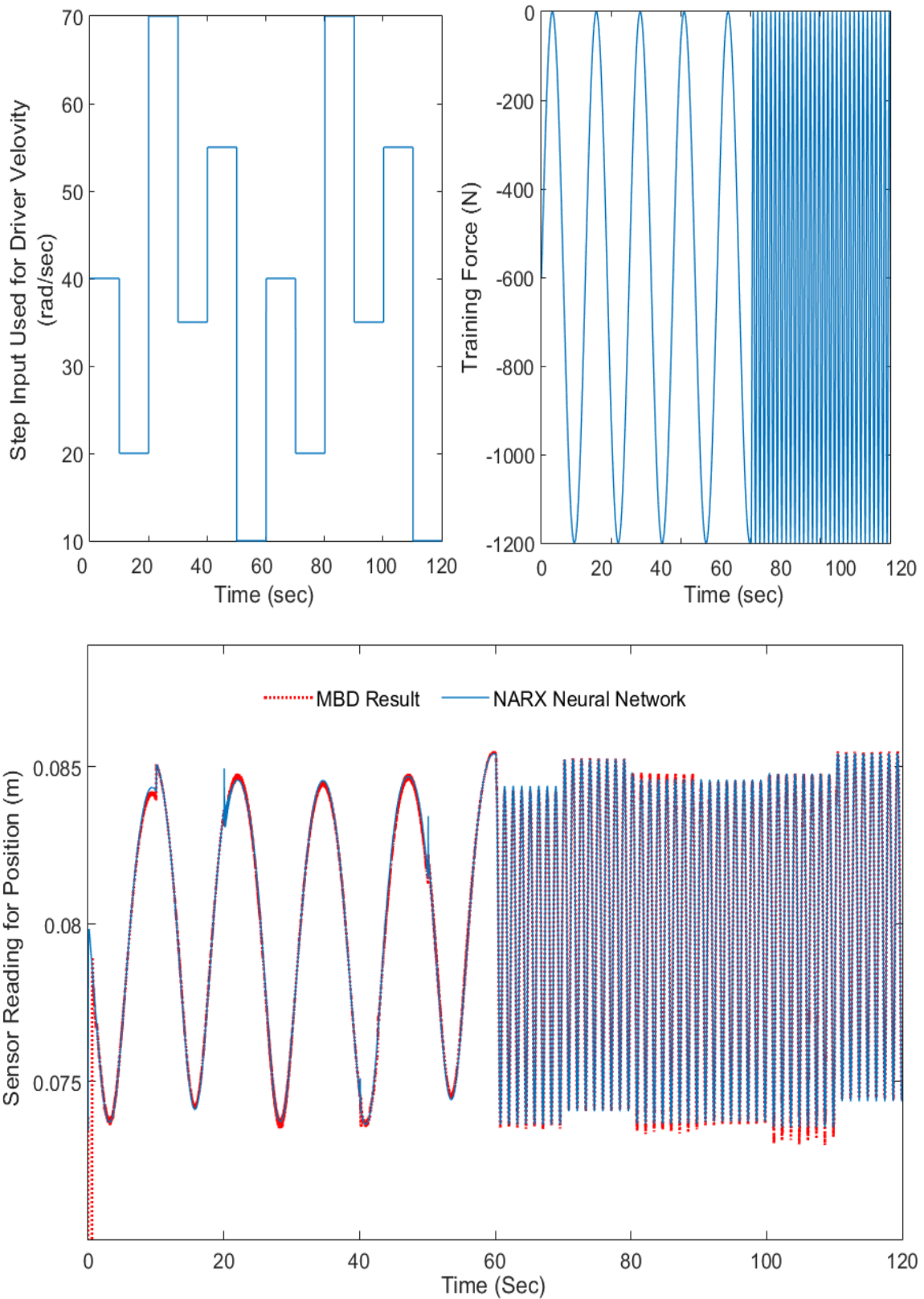


Figure 5.26 Narx confirmation for the extended training inputs

The extended training set is used for the step input as in impulse, ramp, sinusoidal inputs.

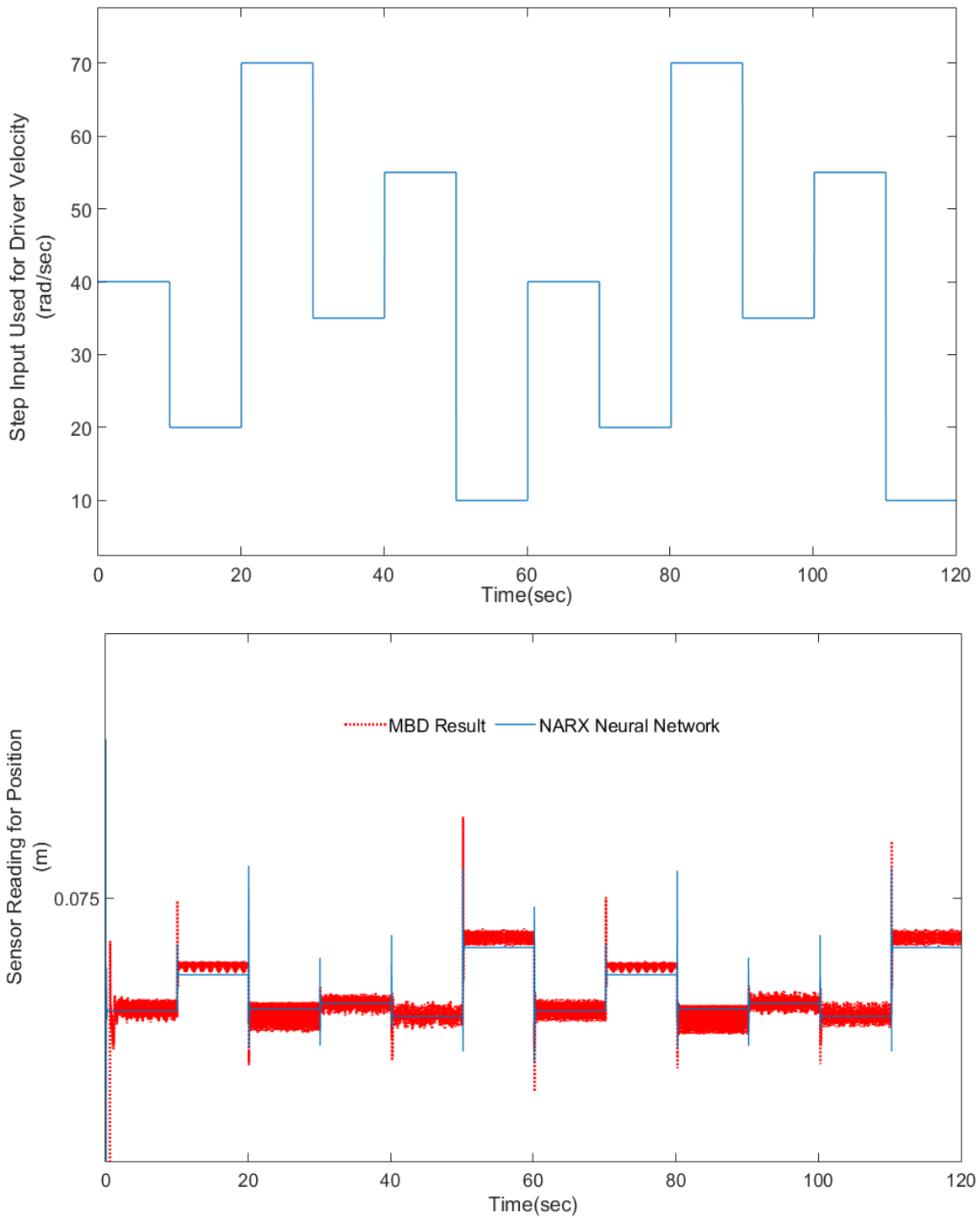


Figure 5.27 Narx model confirmation trained with step input in case of no tensioner force is applied

The tensioner force does not exist in Figure 5.27 and 5.28. That is, the only input affecting the belt span behaviour is the driver velocity. Thus, it is expected that the driver velocity affects the system directly. As driver velocity increases, the tension on the belt span

increases. As a result, the slack side go upward and sensor reads this as decrease in the value given by position sensor.

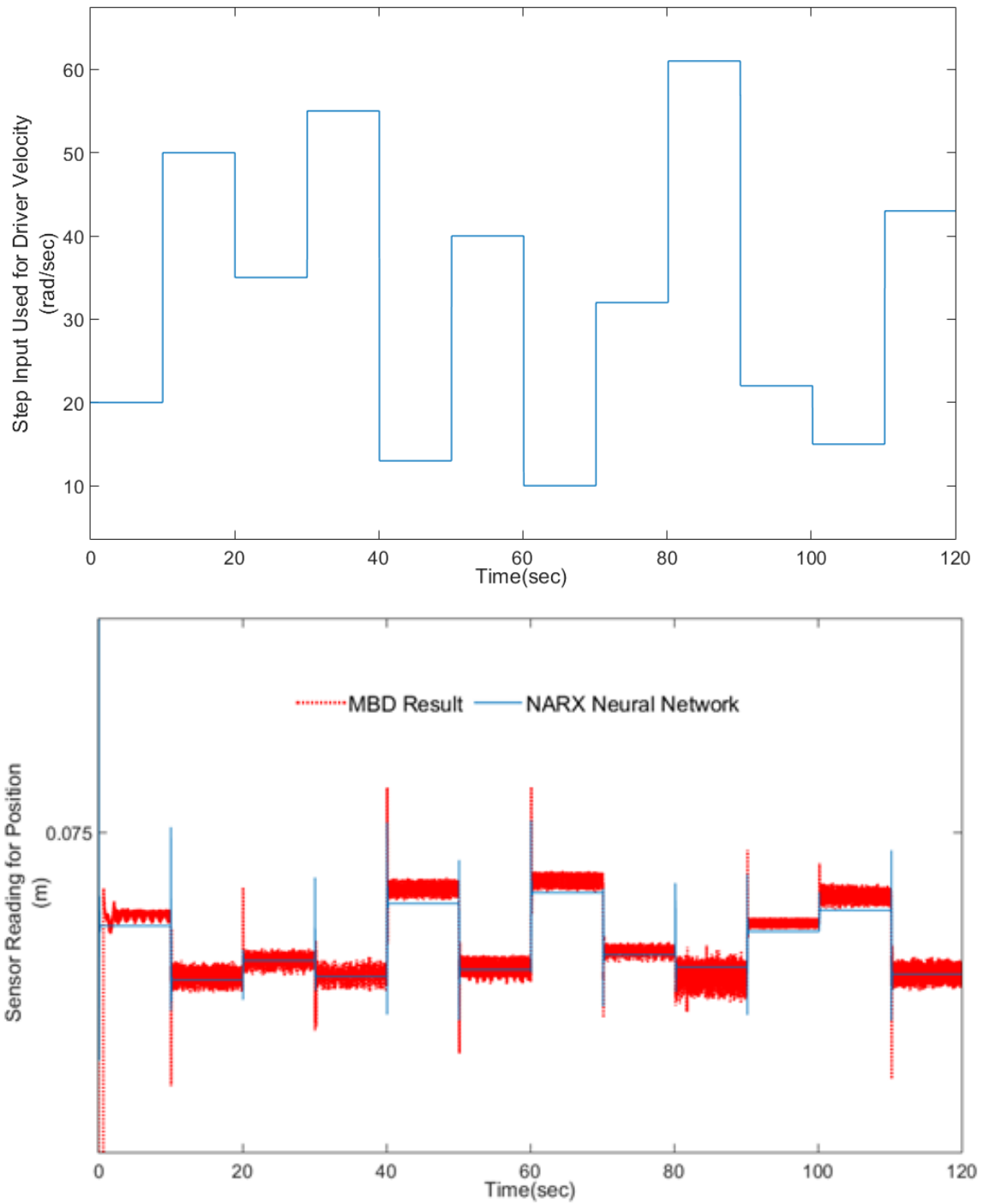


Figure 5.28 Narx model confirmation trained with step input for different driver velocity in case of no tensioner force is applied

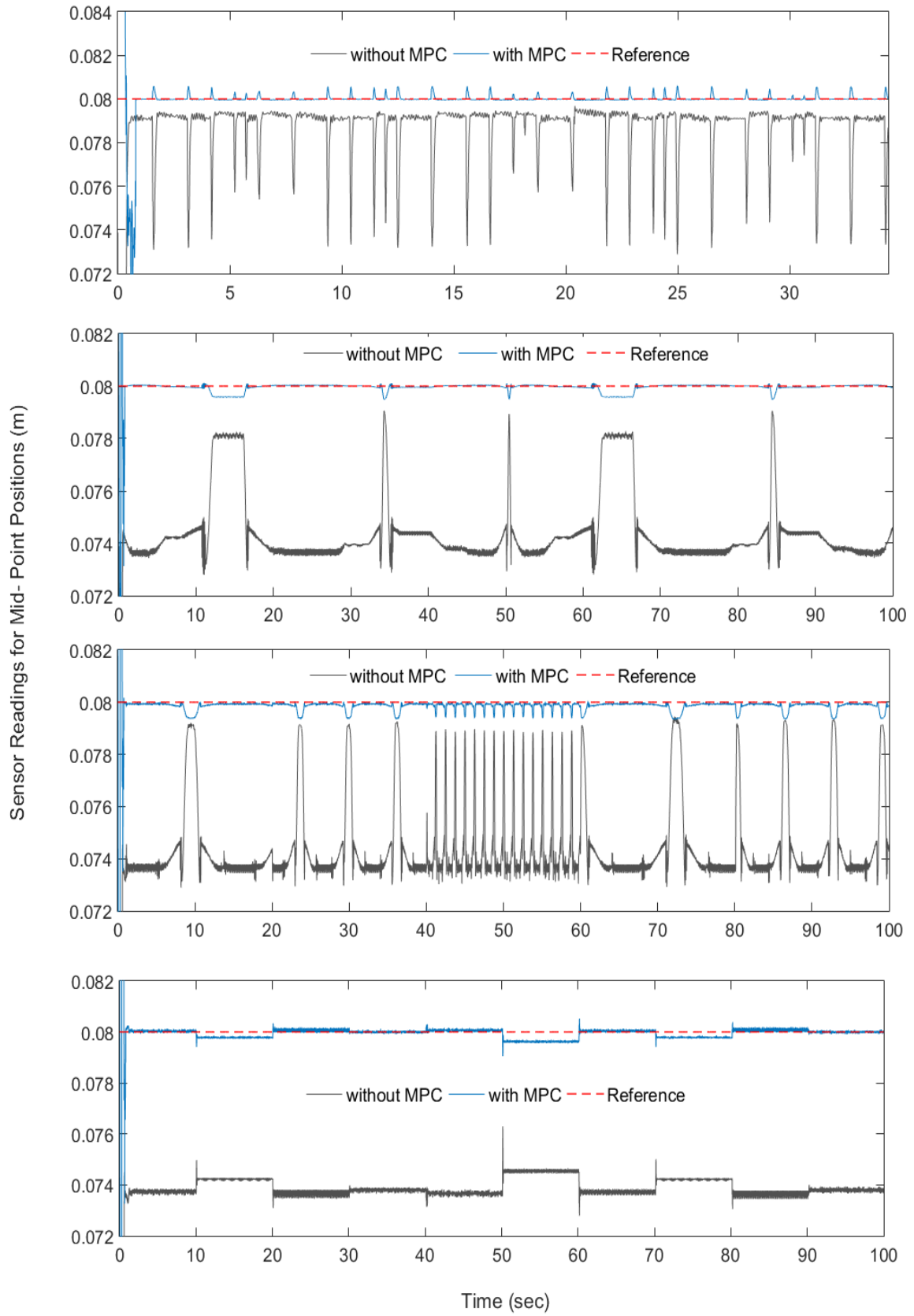


Figure 5.29 Mpc results for impulse, ramp, sinusoidal and step inputs

After design of the prototype serpentine belt drive system with multi body dynamic simulation program, the model predictive controllers are used for the data shown in Figure 5.6, 5.13, 5.20, 5.27 which are used for the impulse, ramp, sinusoidal and step driver velocity types respectively.

5.2 Effect of Sensor Numbers on the Belt Drive System Control

Although decrease in the oscillation of the middle point of the upper belt span reveals that transverse vibration is throughout the span, more than one sensors will be used to decrease the transverse vibration of the span. This case can be thought as a result of a theoretical approach rather than being a feasible one because using more than one sensor makes the system more complex and increase the cost.

The critical point here is to decide the weight of the selected points so that their effects on overall system is established well. Although there can be other methods about how to give weights correctly, two main approaches are described here. Whereas the first approach based on making a reasonable guess and making a correction according to the results, the second one relies on giving a weight decision according to the observations or calculations. The second approach is used for the prototype belt drive system in this study.

In order to apply the approach mentioned above, displacement amounts are considered initially. Using displacements instead of distance value is the key point if the oscillation amount is the concern since the position of the points in the direction of gravity can be somehow closer to the origin due to the geometry of the system. That is, values from the sensors should not be used directly. After displacement amounts are found, points are ordered and evaluated according to the difference between the maximum and minimum values of each point.

Figure 5.31 is shown to present the weights in a more understandable way. K represents the difference between the maximum and minimum distance of observed points and K_{\min} is the minimum value among the K values.

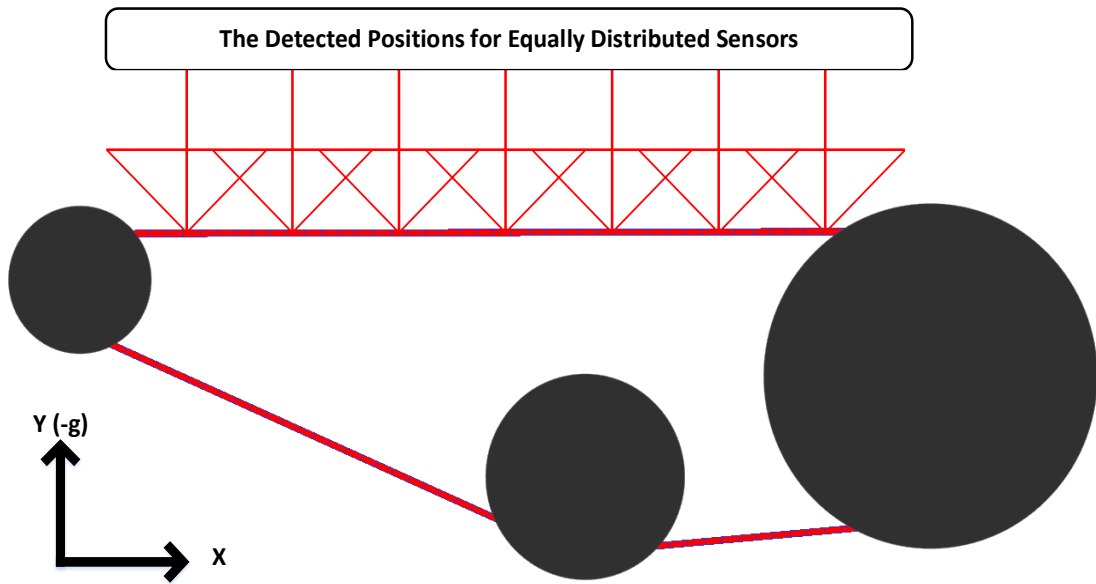


Figure 5.30 Points measured by the equally distributed seven sensors

The sharp edge of the arrows in Figure 5.30 indicates the points measured by position sensors. As can be seen from this figure, seven sensors are employed for the purpose mentioned in the introduction of this section. The sensors are numbered from left to right order according to the reader's perspective. For instance, if sensor one is mentioned, it means the leftmost one according to reader.

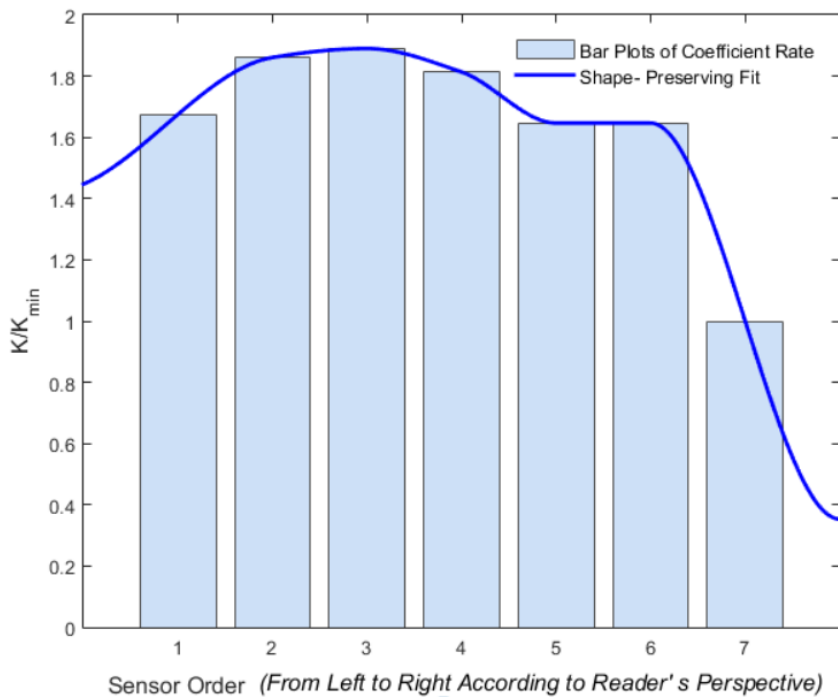


Figure 5.31 Unitless comparison of measured points

The above figure is only used to make comparison among the results for equally distributed seven sensors. The weights are given so that sum of the all coefficients is equal to 1 in the co-simulation. The weight values of each point are given as 0.145258978670547, 0.161354587336083, 0.163886711791279, 0.157228329935026, 0.142780042824176, 0.142780042824176 and 0.0867113066187129 in order.

The results can be summarized as follows:

- Maximum displacement occurs at the 3rd point
- Maximum displacement occurs at the 7th point
- The displacement amounts are greater on the left side (*from the reader's perspective*) of the span mid point
- Displacement decrease on the right end is great compared to the left end

The results seem reasonable. In order to make clarification on this point, another belt and pulley system can be considered with driver and driven pulleys which have same radius. For this situation, it is expected to see the maximum displacement on the midpoint and close displacement values for the left and the right sections of midpoint. However, the radius of driven pulley of the belt and pulley system considered is bigger which restricts the motion of the belt more compared to driver pulley with smaller radius. Briefly, the displacement amount is strictly related to the restriction of the belt motion.

The below figure shows the co-simulation blocks used to control the tensioner when both one sensor and seven sensor are used.

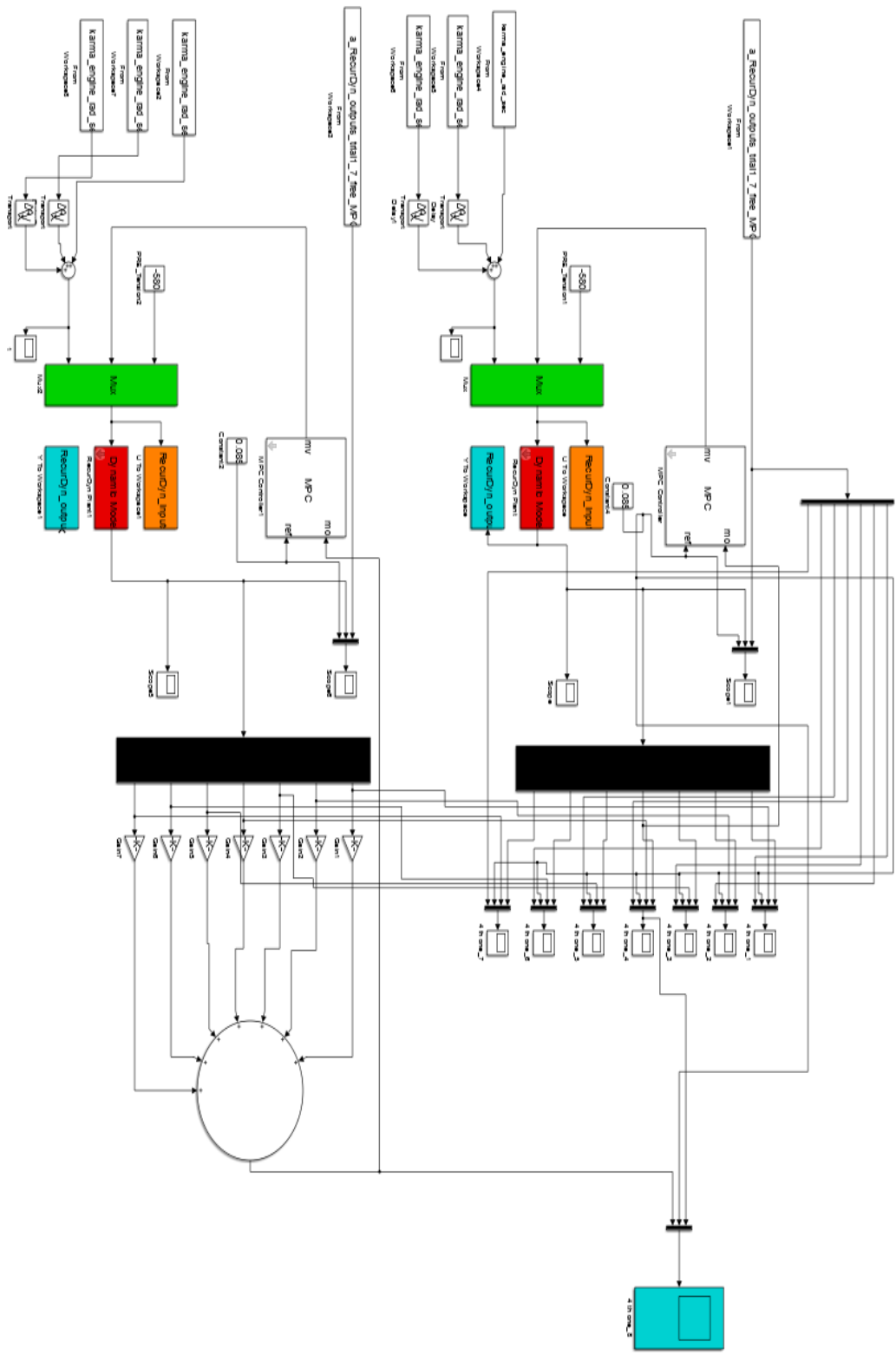


Figure 5.32 Simulink diagram for comparison of model predictive controllers

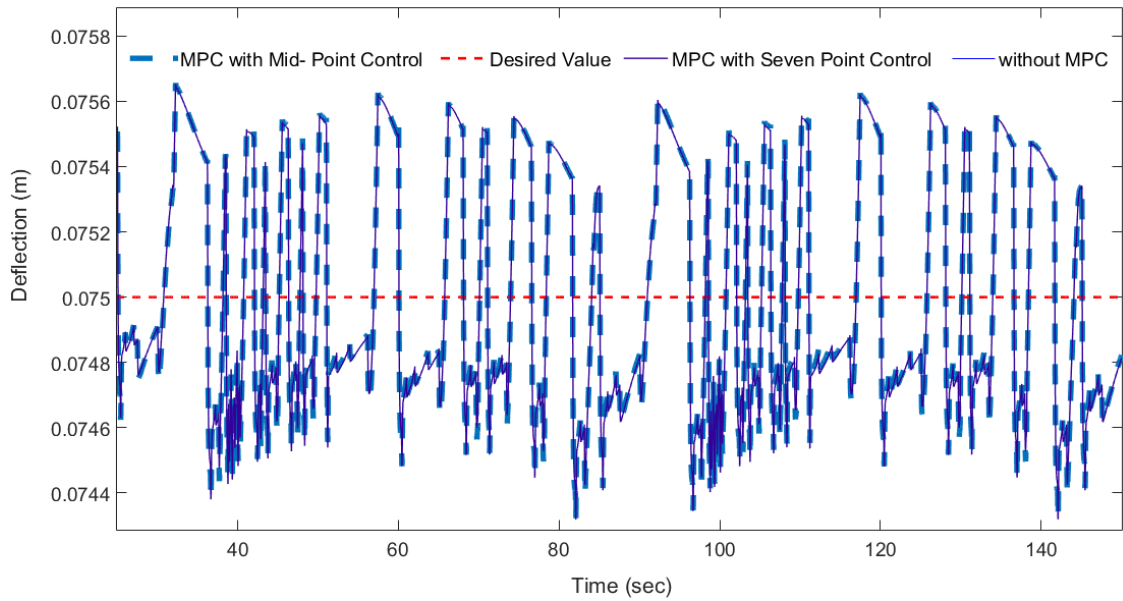
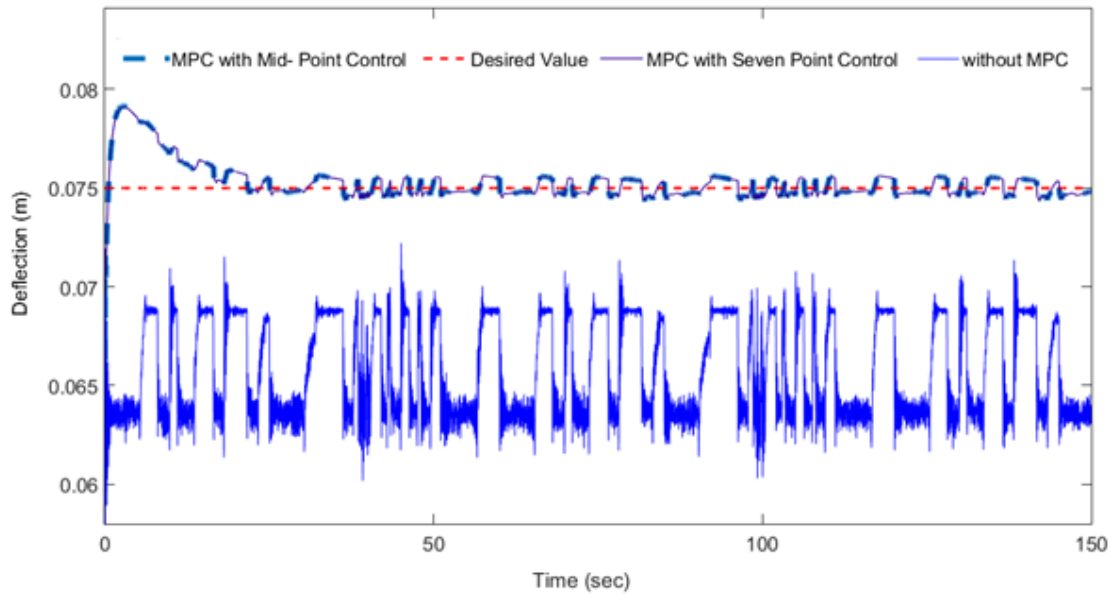
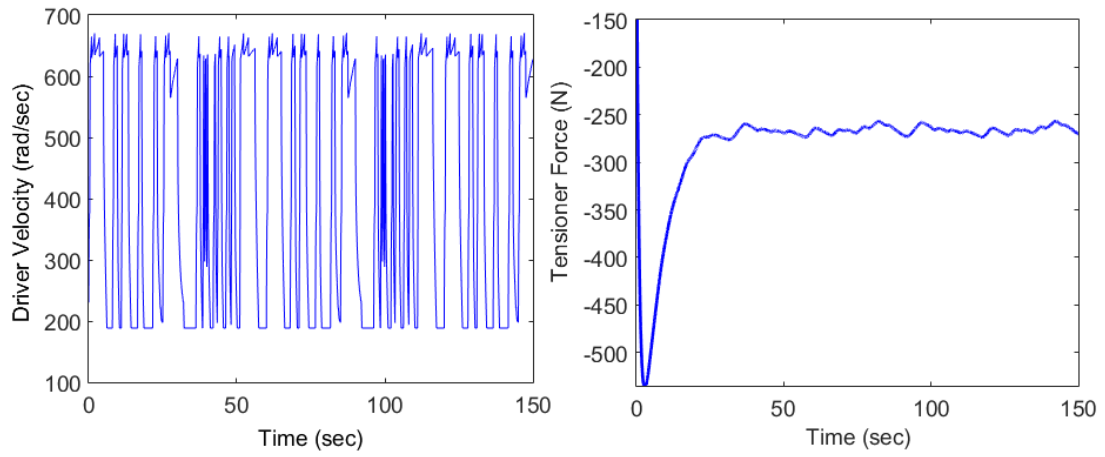


Figure 5.33 Comparison between multiple and single sensor cases

Figure 5.33 consists of four sub-plots. The two plots at the top are used to demonstrate the driver velocity and tensioner force comes from the model predictive controller. The results related to two different control approaches, reference line and uncontrolled plant output are given in the plot placed at the middle of the figure.. Except for the start phase, reference line is followed with -0.7 and $+0.7$ mm tolerance. In order to make a better comparison between the approaches Figure 5.34 is given below.

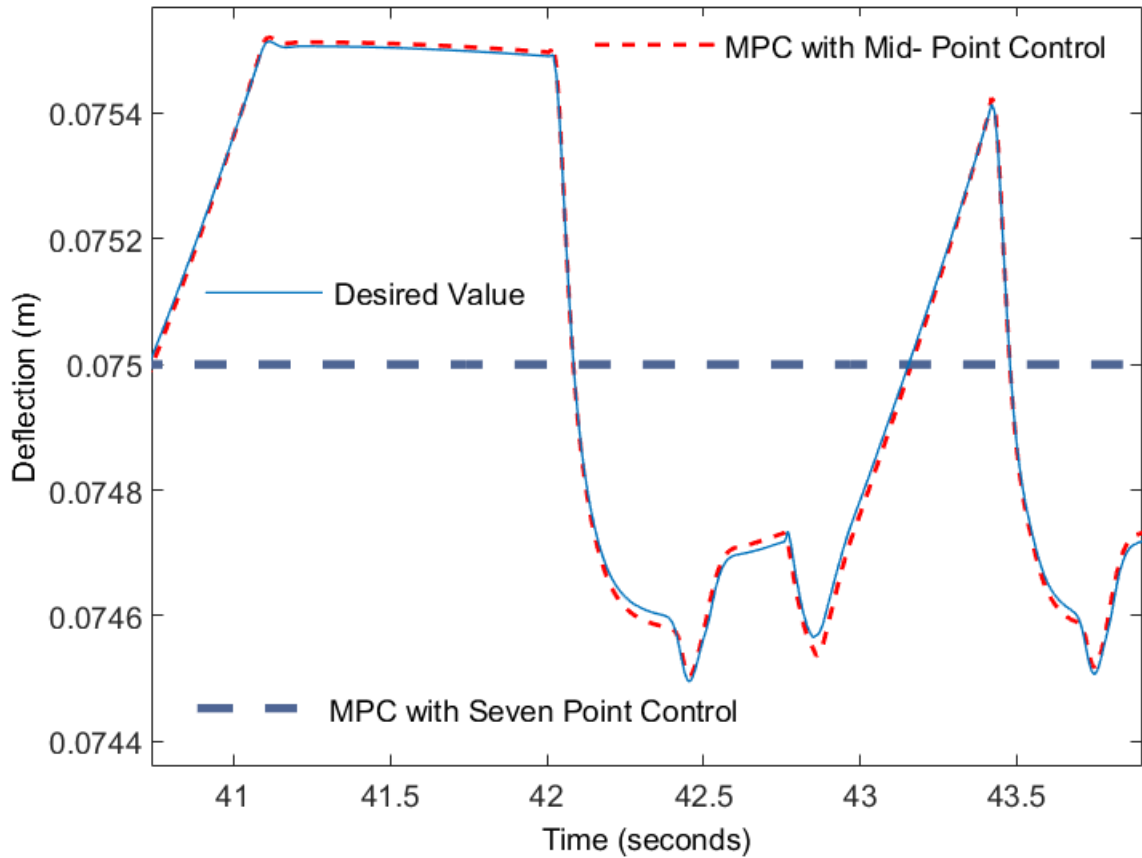


Figure 5.34 Detailed view of comparison for different number of sensor use As can be seen from the above figure the results are compatible with each other except for the small difference which can be ignored.

5.3 Controller Comparison

Section 5.3 composed of three main branches. MPC (Model Predictive Controller) design, PID (Proportional-Integral-Derivative) Controller Design and comparison of these controllers are the branches in this section.

5.3.1 Parametric Study on MPC

Since there are number of parameters to be changed in order to reach the best MPC parameters, it is better to start with a reference MPC and make a parametric study on this reference design. The common approach is to decide sampling time, T_s , in the initial time, then change it if the first choose was poor after several trials tried for the other parameters. Generally, decrease in T_s results in increase in the reduced disturbances but this increase has negative consequence in terms of computation effort. The balance between these two criteria is the key for MPC design. Whereas it is preferred that $T_s \gg 1$ in process control design studies, $T_s < 1$ is the preference in the area of automotive related studies. In case of belt drive system in this study, the sampling time is chosen as 0.01s to respond the sudden changes in transient state despite of the increased computation steps. Similar to the sampling time, prediction horizon is chosen initially. Since the aim of using MPC is the future anticipation, the prediction horizon is increased as soon as the impact of the change is relatively small as a rule of thumb. The control horizon is used for minimization of errors over the prediction horizon and satisfying the conditions defined with constraints. Thus, control horizon is chosen as a smaller value than the predicted horizon.

Table 5.1 reveals the first parametric study for model predictive controller design. As they can be seen from the table, prediction and control horizons are fixed at 10 second and 3 second respectively up to the 20th MPC design to see the effect of weights. The prediction horizon and the control horizon are changed together at 20th design; i.e, the transitions are not done step by step by changing the control parameters individually and evaluate the results one by one because the aim of this table is to show the effect of weights rather than horizons. The controllers after 19th design is used to reveals that similar weight effects can be observed in different horizon combinations.

Table 5.1 Parametric study for mpc design

Controller Name	Prediction Horizon	Control Horizon	Weights		
			Manipulated Variable	Manipulated Variable Rate	Controlled Variable
MPC 1	10	3	1000	10000	1000
MPC 2	10	3	500	10000	1000
MPC 3	10	3	250	10000	1000
MPC 4	10	3	0,01	10000	1000
MPC 5	10	3	0,01	0,01	1000
MPC 6	10	3	0,01	0,1	1000
MPC 7	10	3	0,01	0,01	0,01
MPC 8	10	3	0,01	0,01	10
MPC 9	10	3	0,01	0,01	1000
MPC 10	10	3	0,01	0,01	100000
MPC 11	10	3	0	0,01	100000
MPC 12	10	3	0,01	0	100000
MPC 13	10	3	0	0	100000
MPC 14	10	3	0	0,0001	100000
MPC 15	10	3	0	0,00001	100000
MPC 16	10	3	0	0,01	100000
MPC 17	10	3	0	0,001	1
MPC 18	10	3	0	0,0001	1
MPC 19	5	2	0	0,001	1
MPC 20	5	2	0	0,01	0
MPC 21	5	2	0	0,01	0,001
MPC 22	5	2	0	0,01	0,01
MPC 23	5	2	0	0,01	10
MPC 24	5	2	0	0,001	2
MPC 25	5	2	0	0,001	0.50
MPC 26	5	2	0	0,0001	100000
MPC 26	5	2	0	0,0001	100000

The controllers of which parameters are given in Table 5.1 are plotted in Figure 5.37 based on the driver velocity shown in Figure 5.35 and -580 N constant pre-tension force. The results without controller for these inputs are given in Figure 5.36 separately. Since the change in each parameters can have different effects when other parameters are varied, several MPC designs are demonstrated in this table. The summary of this parametric study is that if the controlled variable is raised, the manipulated variable rate is made closer to the zero value and the manipulated variable is fixed at zero at the same time, the controller follow the reference line better. Time to reach the desired value decreases. Although several reference trajectory is tested, the effect of weights are the same; hence, only step input reference trajectory is shown.

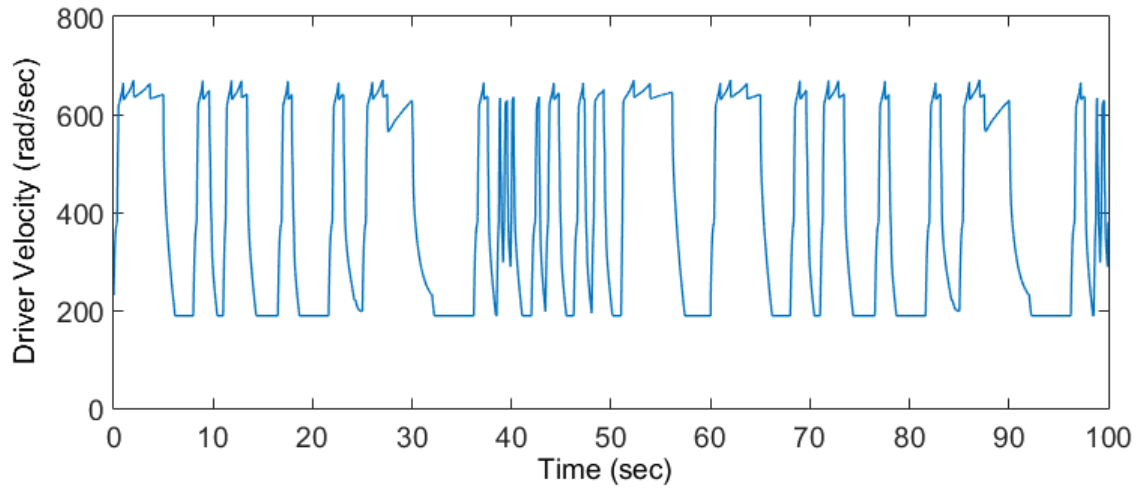


Figure 5.35 Driver Velocity Input with Respect to Time Used For Controller Comparison

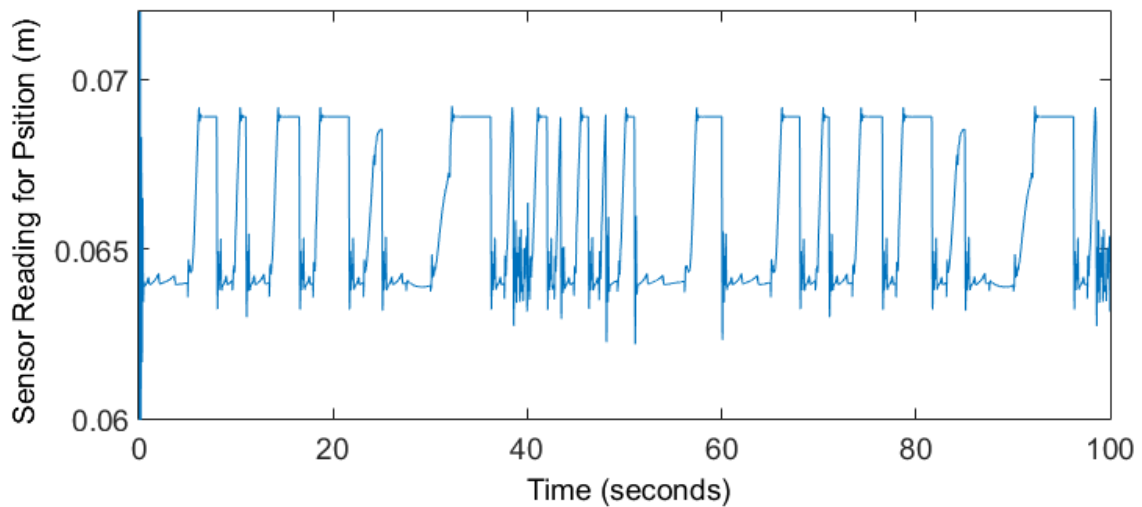


Figure 5.36 Sensor reading in case of no tensioner force

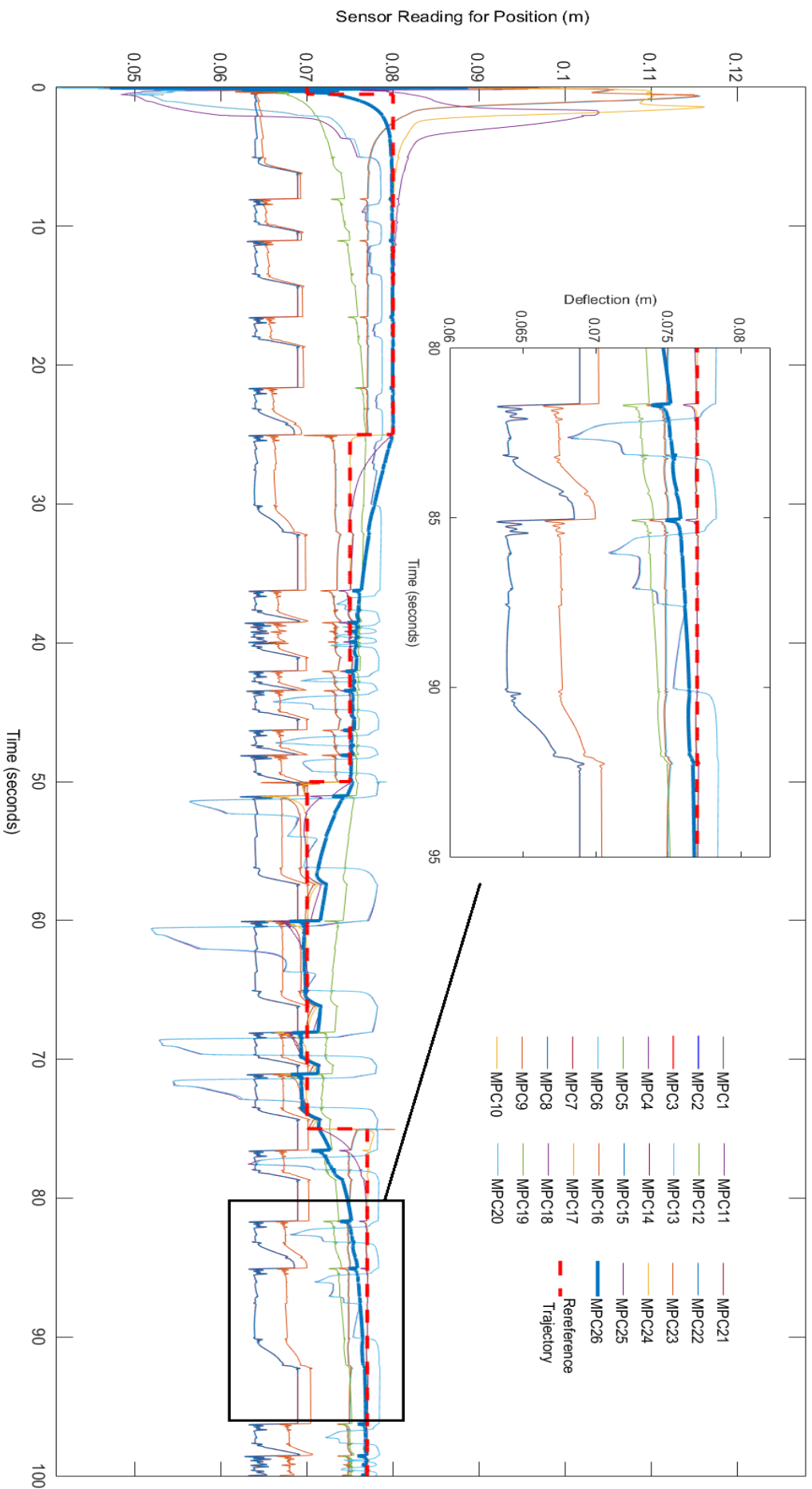


Figure 5.37 Comparison of model predictive controllers

In order to see how control horizon affects the controller; prediction horizon is fixed at 10, control horizon is changed for the value of 3. In addition, weights are fixed at the 26th controller parameters of previous parametric study.

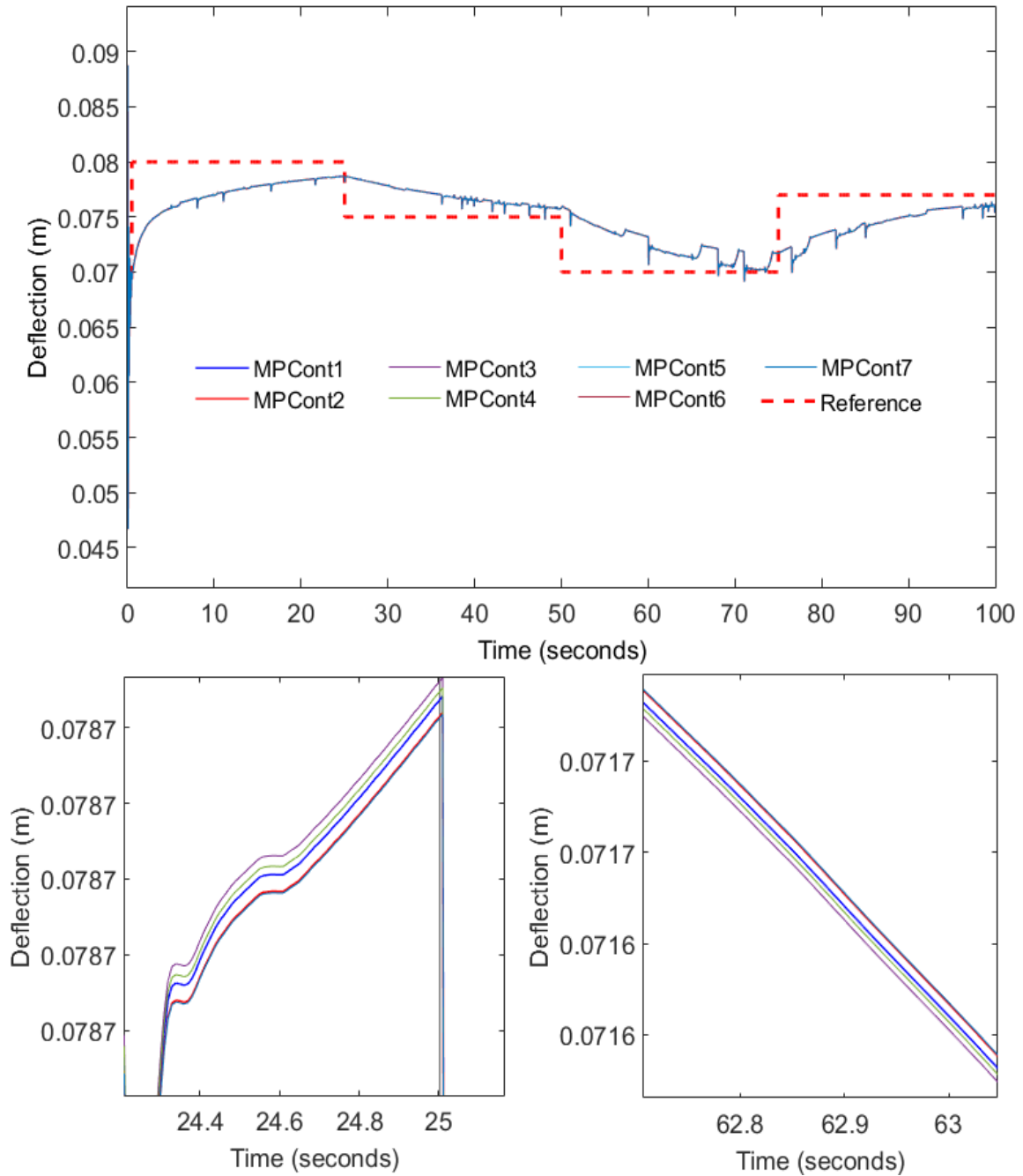


Figure 5.38 Mpc controller's reference line tracking performance for different control horizons

The control horizons are 3,7,1,2,10,15,11 for MPCCont1, MPCCont2, MPCCont3, MPCCont4, MPCCont5, MPCCont6, MPCCont7 respectively.

The summary of the above figure is given in the table below. It can be easily inferred from the table that as control horizon decreases, time to reach reference line in case of sudden change decrease.

Table 5.2 Effects of control horizon on mpc reference line tracking

Controller Name	Prediction Horizon	Control Horizon	Order of Success for Reaching the Desired Value
MPC ont1	10	3	3
MPC ont2	10	7	4
MPC ont3	10	1	1
MPC ont4	10	2	2
MPC ont5	10	10	5
MPC ont6	10	15	7
MPC ont7	10	11	6

Then, control horizon is fixed at the best value for the same weights and prediction horizon is increased for the reference time, 10 sec. The result are given in the below table.

Table 5.3 Effects of prediction horizon on mpc reference line tracking

Controller Name	Prediction Horizon	Control Horizon	Order of Success for Reaching the Desired Value
MPCont3	10	1	4
MPCont 9	25	1	2
MPCont 10	40	1	1
MPCont 11	15	1	3

The increase in the prediction horizon for the belt drive system model decreases time to reach the followed trajectory. In order to support this idea, prediction horizon is decreased for the reference value and the comparison is made with the controller given in the above table. The controller named with MPC Controller is better than the others in terms of

reaching the desired level in the quickest way. However, it is hard to say that increasing the prediction horizon and decreasing control horizon at the same time gives better result without trying the other combinations. Although it is possible to give the limits and steps for horizons and weights and then test the combinations of all in a single function, it takes great amount of time for simulation. In order to escape from this huge amount of time consuming job, trial and error procedure is followed as the most practical engineering way. In this procedure it is realized that though the effect of weights on the tried horizons are almost the same, choosing different horizon combinations can provide distinguishable positive benefits from different perspectives. In order to demonstrate this phenomena, MPCCont 10 in table 5.3 is compared with the controller whose prediction horizon is 5 sec and control horizon is 1 sec. This new controller is named as MPC_P5H2 and the previous one is named as MPC_P40H1.

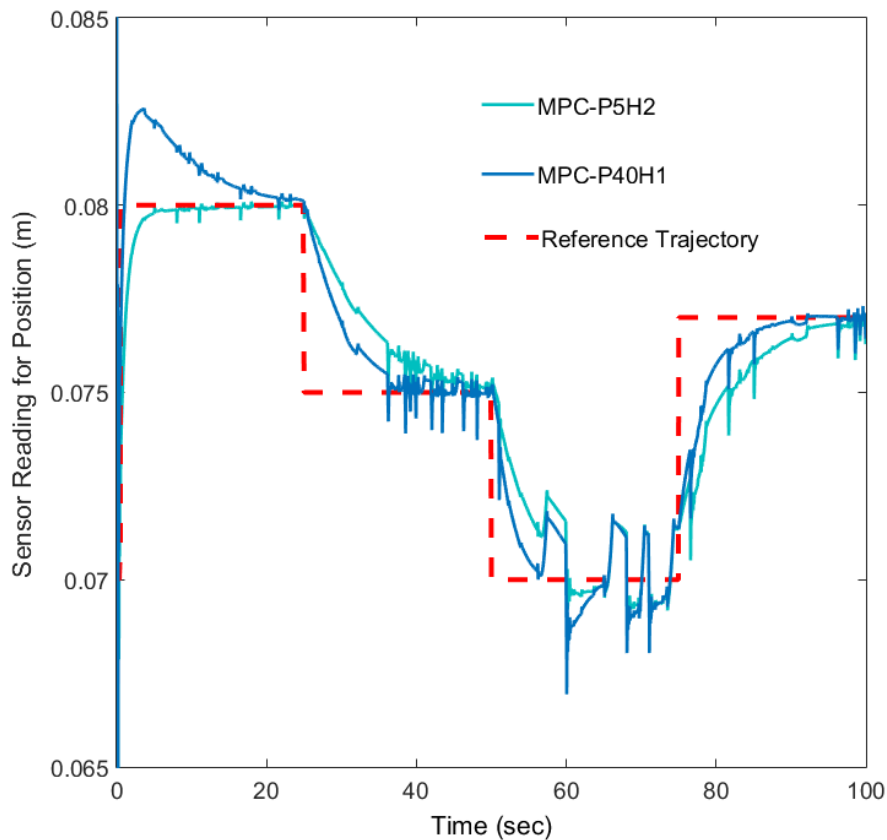


Figure 5.39 Comparison of Mpc for step reference trajectory

As it can be seen from the above figure, MPC-P40H1 enable the drive system to show reactions more. Thus, it reaches the reference line faster; nevertheless, it should be noted that this achievement does mean that it continues to follow the reference line. Even though MPC-P5H2 reaches the reference line slower it keeps to follow the reference line for 0.08m.

Briefly, whereas MPC-P5H2 is better in the initial state, MPC-P40H1 is better after 21 seconds. Both of the controller will be used when they are compared with PID controllers.

5.3.2 Parametric Study on PID

Even though the main focus of this study is to model and control the belt and pulley system with model predictive controller, its performance is compared to the most commonly used controller, PID. For this conventional and commonly used controller approach several automatic tuning algorithms are used and they are shown in the table below.

Table 5.4 Tuned parameters for pid controller design

Automatic Tuned Parameters for PID			
PID Parameters		PID Parameters/ output limit	
IDEAL		IDEAL	
Parameters	Values	Parameters	Values
P	2283	P	-7
I	286	I	100000
D	0	D	0
PARALLEL		PARALLEL	
Parameters	Values	Parameters	Values
P	0	P	0
I	-653620	I	-653620
D	0	D	0

Best P,I and D Gains	
Parameters	Corresponding Values
P	10000
I	-6536
D	100

Despite of the success of the reference tracking of the controllers whose parameters are gathered from the automatic Matlab tuning listed in the above table, a more successful PID controller is established by changing the parameters manually which are shown under the title of best P,I and D gains in the above table. Since automatic tuning gives different results for ideal and parallel cases for which PID controllers will give different results, both results are shown in the table in spite of the fact that they can be equated only by changing K_p value. Compensator formula for ideal and parallel cases are given in order below:

$$P \left(1 + I \frac{1}{s} + D \frac{N}{1+N\frac{1}{s}} \right) \quad \text{and} \quad P + I \frac{1}{s} + D \frac{N}{1+N\frac{1}{s}} \quad (28)$$

where

- P proportional
- I integral
- D derivative
- N filter coefficient

or the formulas can be rewritten in terms of gains and time constants as follows

$$K_p \left(1 + \frac{1}{\tau_i s} + \frac{\tau_d s}{\frac{\tau_d s}{N} + 1} \right) \quad \text{and} \quad K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \quad (29)$$

where

- K_p proportional gain
- K_i integrator gain
- K_d derivative gain
- T_f derivative filter time
- T_i integrator time
- T_d derivative time
- N derivative filter constant

The name ‘best’ in the table refers to the best among the PID controllers used for parallel representation in this study. In order to have this controller parameters, several trial and errors are made. Firstly, all the gains are changed by isolating the effect of others by giving zero as the value to the others. The purpose is reaching the desired value as soon as possible and tracking the reference value in case of input changes. K_p is used in the initial step. It is

observed that the increase in the magnitude of K_p has positive effect on the purpose mentioned above around the values gathered by trial and error. Changing K_i gains by putting zero value to the others, do not have a considerable effect. In addition, using K_d gain only will not be a good idea owing to the significant amount of oscillation. Secondly, values other than zero is assigned to the gains by changing them in the order mentioned above after fixation of the others to use all the gains for the improvement of controller performance. As mentioned above, the increase in the magnitude of proportional, about the value 10000, ease to reach the reference line; however, negative K_p values restrict the benefits from K_i and K_d gains. Thus, positive value for this gain is used. After fixing the value for K_p , K_i is changed. Opposite to the case in K_p , the positive effect on reaching and following the desired value can be adjusted for negative values of K_i . Despite the reduced oscillation effect of further decrease in the value of K_i , following the reference trajectory is affected negatively from this change. Thus, it is fixed about the value such that the positive effect is maximized. Then, K_d gain is adjusted with Filter Coefficient. Filter coefficient is fixed at 100 because further increase results in great amount of change in the amplitude of oscillations and decrease of N value from the fixed value, eliminate the effect of K_d .

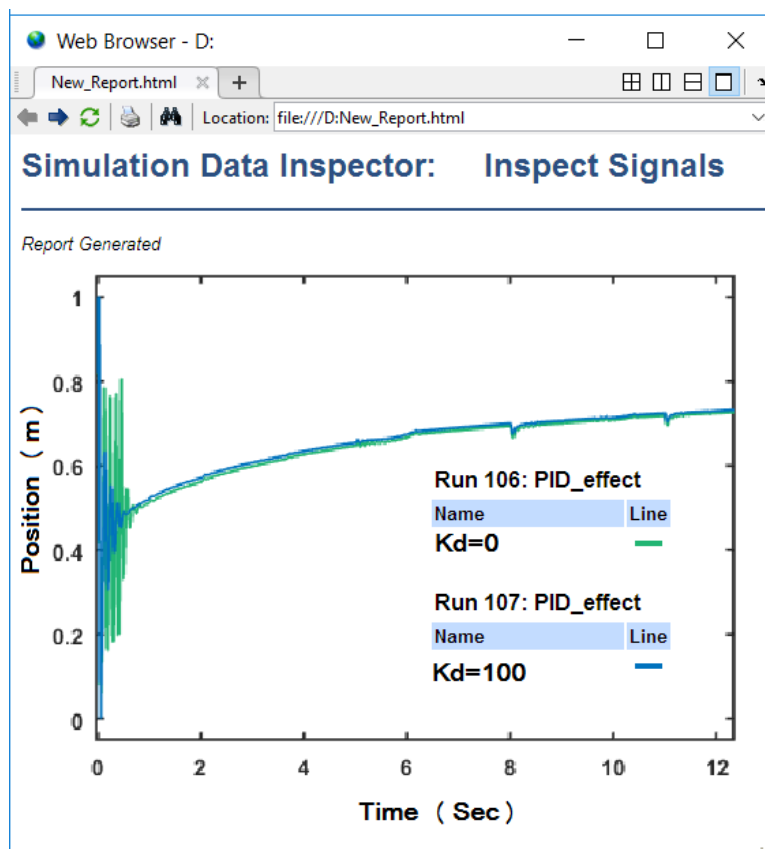


Figure 5.40 Html report screen prepared for kd effect

Figure 5.40 shows the positive contribution of K_d in the initial step. With the given K_d value, oscillation amount is reduced for the first second. Other the initial step, there is almost no effect of K_d for the system considered. Simulink block diagram of PID controller used in this study is shown below.

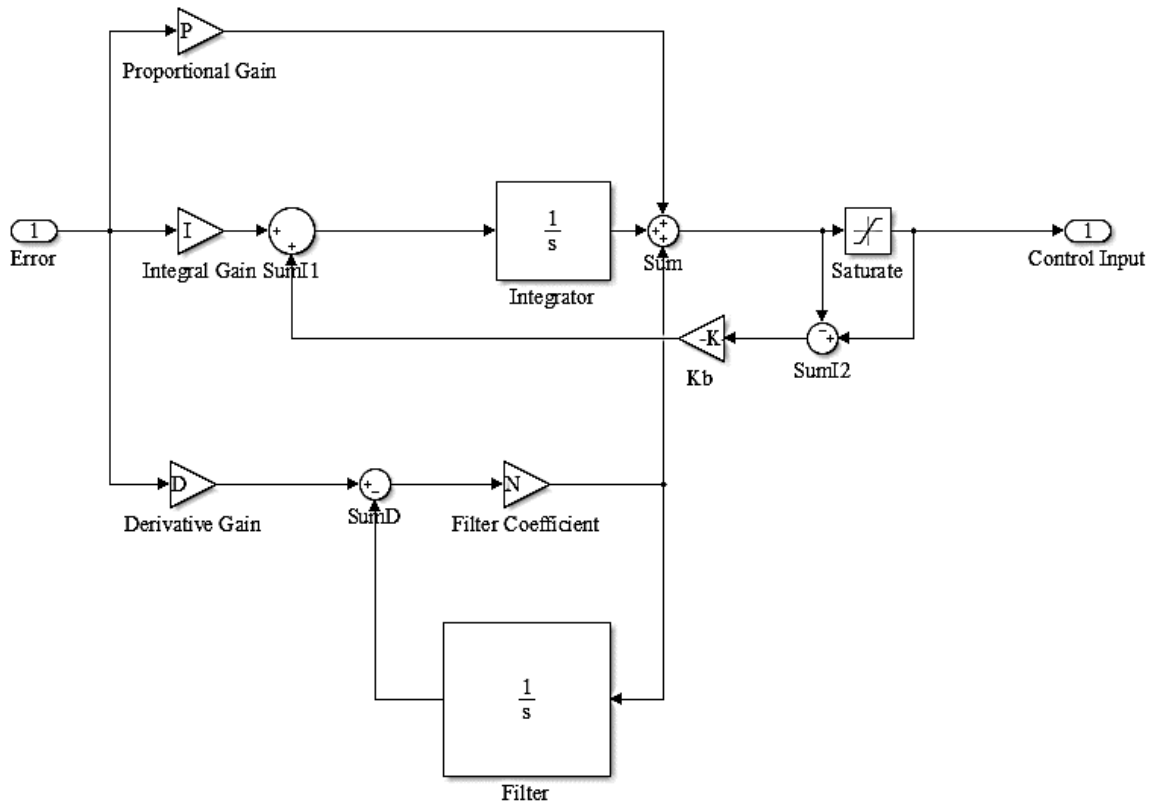


Figure 5.41 PID block used for comparison

Finally, the abovementioned procedure for attaining a value for the constants P,I,D are repeated by changing one and fixing the others over the last design to improve the controller for the design objective.

5.3.3 MPC and PID Comparison

The MPC and PID controller comparison is done in this section. The parameters of these two types of controller are given in figure below for the prototype belt drive model with -580N pre-force. The driver velocity for the comparison is same with the velocity profile shown in Figure 5.33.

Since automatically designed PID controllers gives not satisfactory results in terms of in terms of reference tracking and oscillation amplitude, they are not used in the following figure to make a more clear comparison.

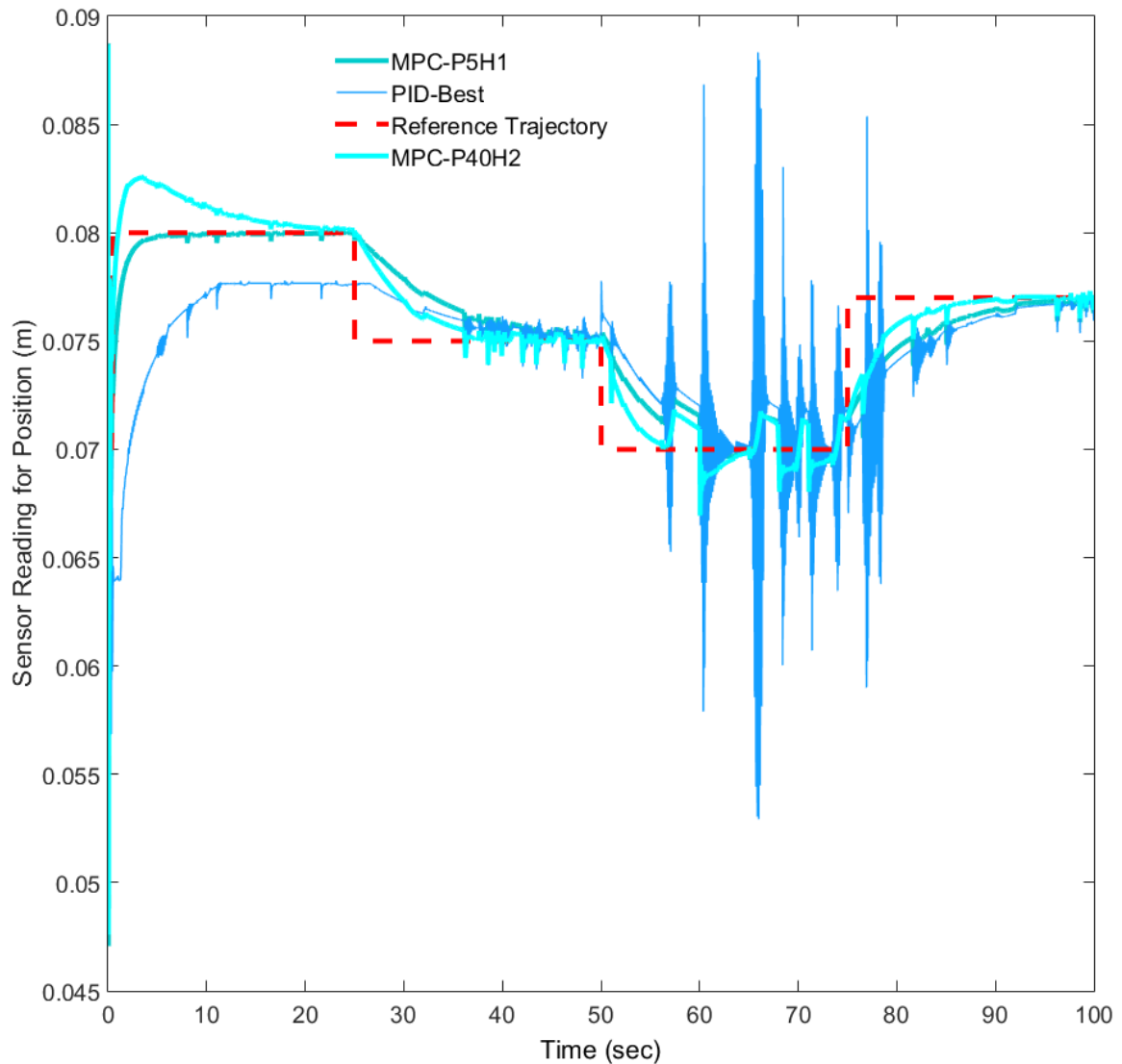


Figure 5.42 MPC vs PID controller

. The oscillation amount with PID controller reaches unacceptable level for the reference line of 0.07m. The table 5.5 is provided to make a comparison from the perspective of mean value and speed of reaching the desired values. The mean values are calculated for the overall time range. It is used to show that all the controllers tries to be near the reference lines. However, the maximum and minimum values are changing; therefore, this data cannot be used alone. When we look at the time to reach the desired level, **MPC-P40H1** is the fastest one followed by **MPC-P5H2**.

Table 5.5 Comparison of controllers

	Reference	MPC-P40H1	MPC-P5H2	PID
Mean Values (m)	0,07542	0,07561	0,07582	0,07451
Absolute Values of Mean Differences (m)	-	0,00019	0,0004	0,00091
Time To Reach Reference Line with 0.080 m (sec)	-	(1,02) / 23,17	4,89	-
Time To Reach Reference Line with 0.075 m (sec)	-	38,06	48,07	51,62
Time To Reach Reference Line with 0.070 m (sec)	-	56,86	60,03	61,1
Time To Reach Reference Line with 0.077 m (sec)	-	89,91	94,97	98,82

In addition to the above comparison, model predictive controllers are compared in terms of cost values in Figure 5. 42. The closeness of the cost values to zero reveals the degree of controller performance.

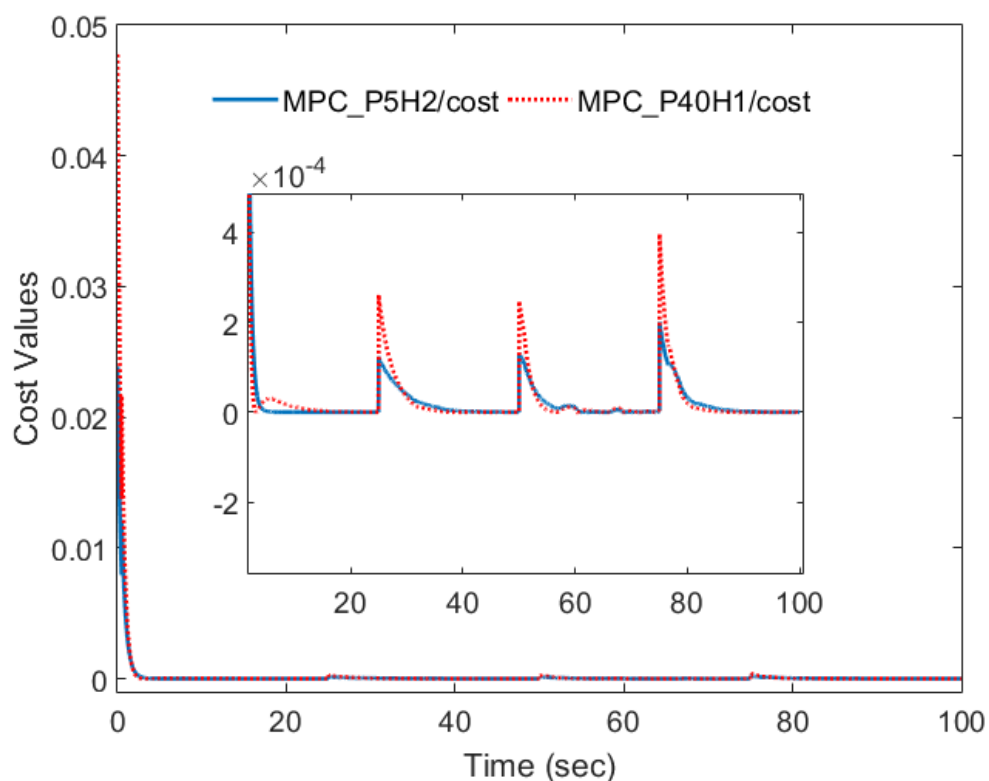


Figure 5.43 The comparison of the controllers

5.4 Case Studies

Since number of case studies introduced in this study, below table is supported to better understand the different situations in the cases.

Table 5.6 Case studies

Elementary Level Case Studies	
Cases	Velocity Profile
I.a	Sudden Increase in Vehicle Speed
II.a	Sudden Decrease in Vehicle Speed
III.a	Combination of Case1 and 2
IV.a	Case3 for Narrower Brake- Throttle Range
Intermediate Level Case Studies	
Cases	Velocity Profile
I.b	Sudden Increase in Vehicle Speed
II.b	Sudden Decrease in Vehicle Speed
III.b	Combination of Case1 and 2
IV.b	Case3 for Narrower Brake- Throttle Range
Advanced Level Case Studies	
Cases	Velocity Profile
I.c	Sudden Increase in Vehicle Speed
II.c	Sudden Decrease in Vehicle Speed
III.c	Combination of Case1 and 2
IV.c	Case3 for Narrower Brake- Throttle Range

The main categorization depends on the construction of the system components and sub-categorizations are according to the input values. Input related cases for the elementary, intermediate and advanced level case studies are common. In order to escape from the unnecessary repetition of these common data, they are demonstrated in this section. MPC_P5H2 is preferred for the case studies.

5.1.1 Elementary Level Case Studies

Elementary level case study is considered as a prototype model for the belt drive system. Figure 5.43 shows the response of the deflection for NARX type neural network whose details are given in Chapter 4.

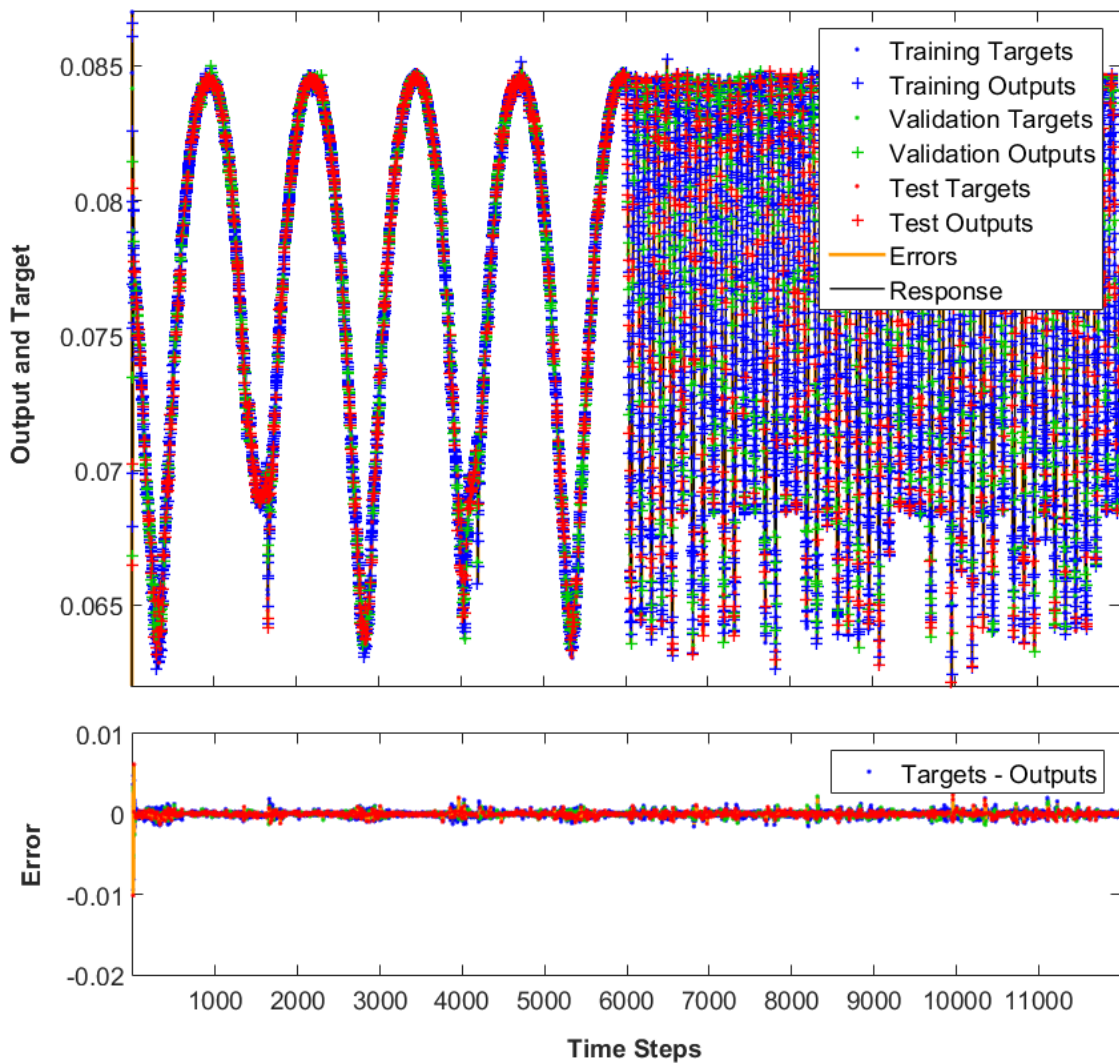


Figure 5.44 Neural network responses for elementary level case studies

Although above figure gives an idea about the success of the neural network for system identification, regression plots and performance graph are also demonstrated in the following figures. Regression plots are drawn for training, validation and test one by one. Then, over all regression plots are combined in a single one. In other words, regression plot for all supplied data are drawn as a fourth one. Data's being on or near the fitting line indicates that neural network model seems good enough for system identification.

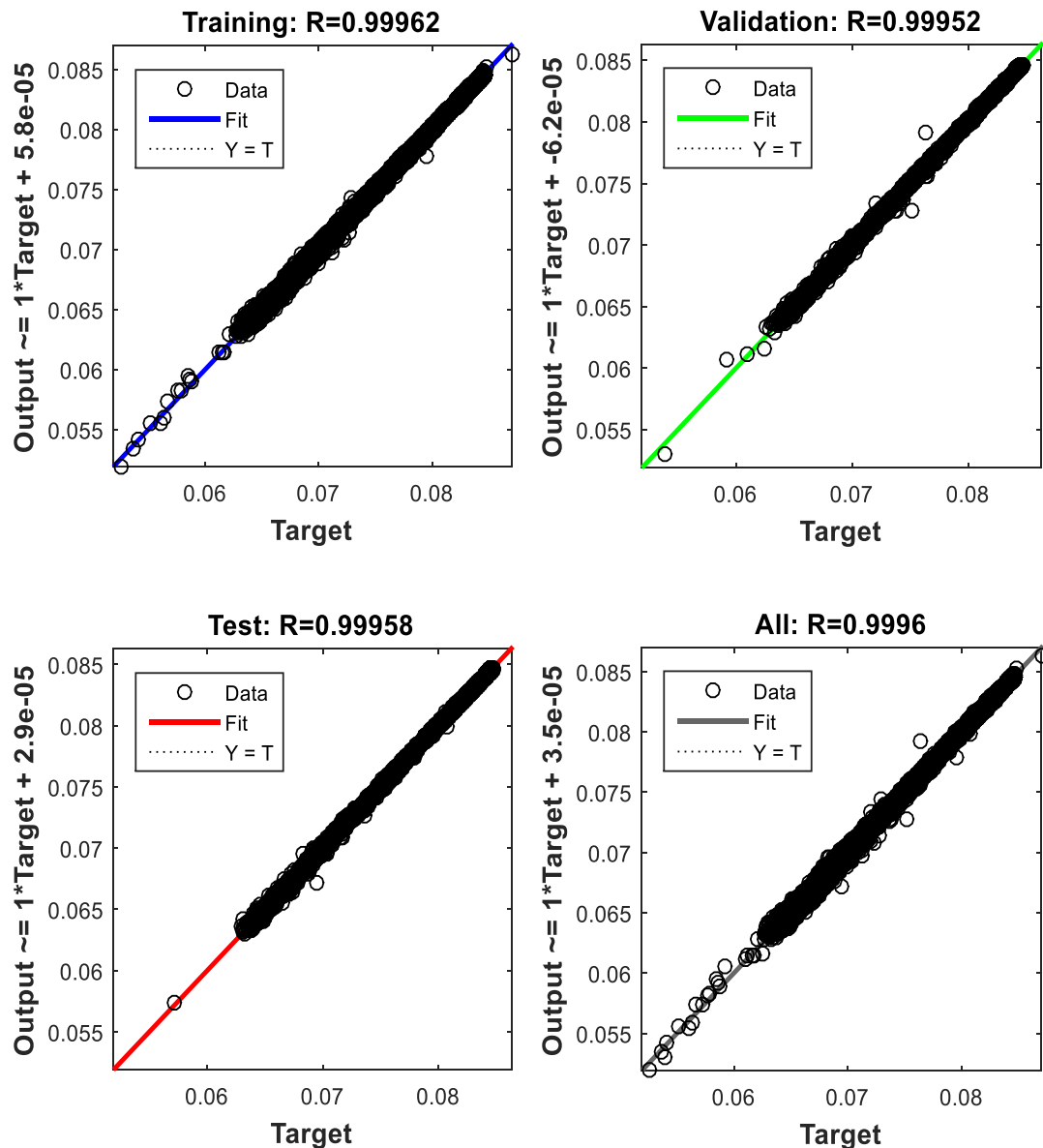


Figure 5.45 Regression Plot of the System for Elementary Level Case Studies

The next figure shows the same success in terms of system's performance values according to the mean square error. There are three solid and two dotted lines in Figure 5.45. Two dotted lines are used to show the best validation performance value. Blue, green and red solid lines are used for training, validation and test data respectively. Mean square error values for all of the solid lines reaches values smaller than $10 \cdot 10^{-7}$. In addition, convergence of the solid lines on the same line shows the consistency for training, validation and test.

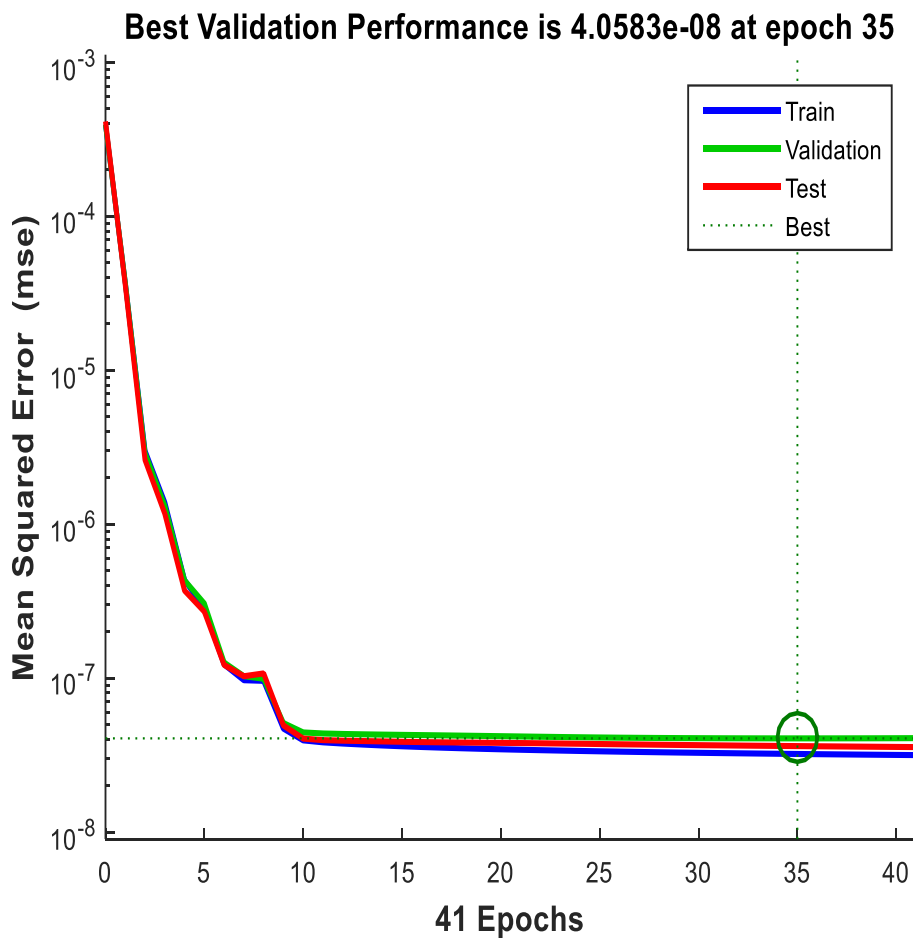


Figure 5.46 Performance values of the system for elementary level case studies

According to the neural network model, model predictive controller is designed. This controller is tested for four different cases and the results are demonstrated in the below.

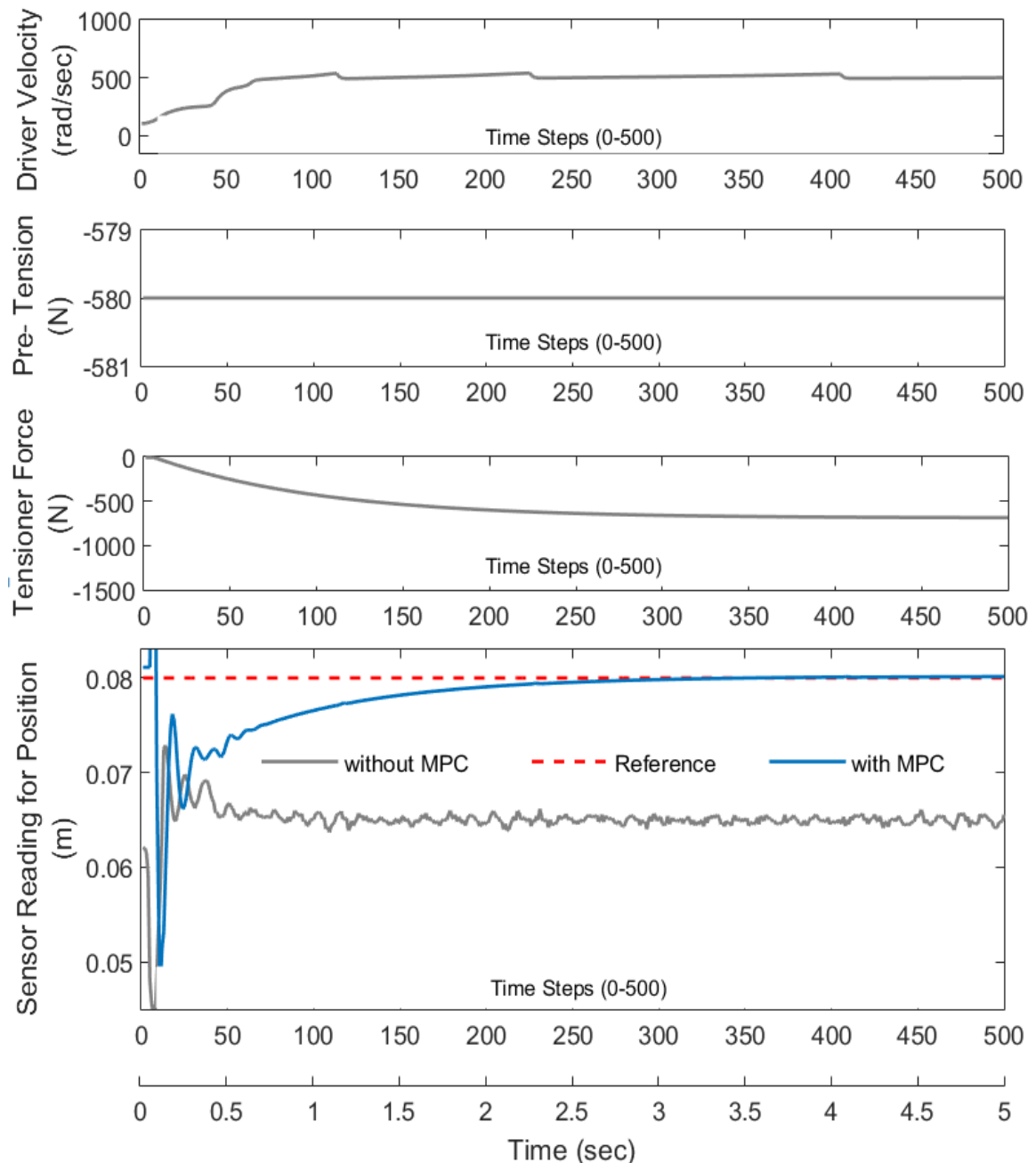


Figure 5.47 Test data 1 for elementary level case studies

The above figure can be summarized as follows:

- There is a sudden increase in the driver velocity in the initial step
- Relatively small step changes in the driver velocity profile exhibits the change in the gears.
- Sudden change in the driver velocity is compensated with relatively big change in the control input; i. e., tensioner force.
- The oscillation in the initial step is decreased as time goes
- Reaching a reference value of 0.08 takes about three seconds.

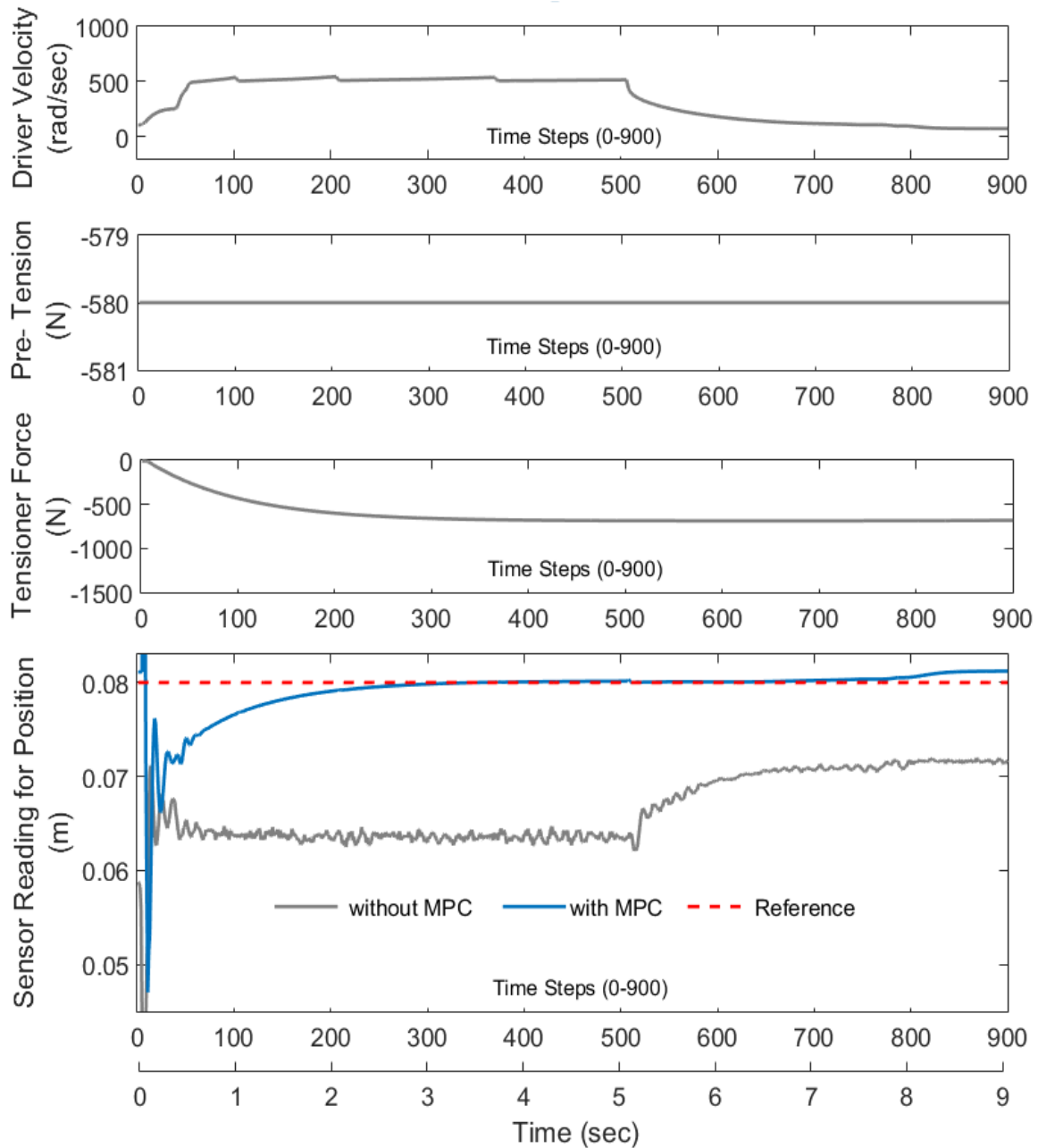


Figure 5.48 Test data 2 for elementary level case studies

The summary of Figure 5.47 is given as follows:

- After sudden increase in the driver velocity profile, there is a sudden decrease after five seconds.
- Due to the decrease in the driver velocity profile, belt moves in a more relaxed manner. In other words, tightening effect of driver velocity is decreased. As a result, belt span moves in a lower value as can be seen from the result without MPC. The decrease in the belt tension can be also understood from the decrease in the small oscillations in the same line.

- After 5 seconds, there is a tendency of exceeding the reference line for the result with MPC. Although maintained tensioner force do not enable so much overshoot in the initial step, the slope for overshoot increase after 7.5 seconds due to the internal model. Fortunately, overshoot value is fixed at about 8.6 seconds with the help of feedbacks.

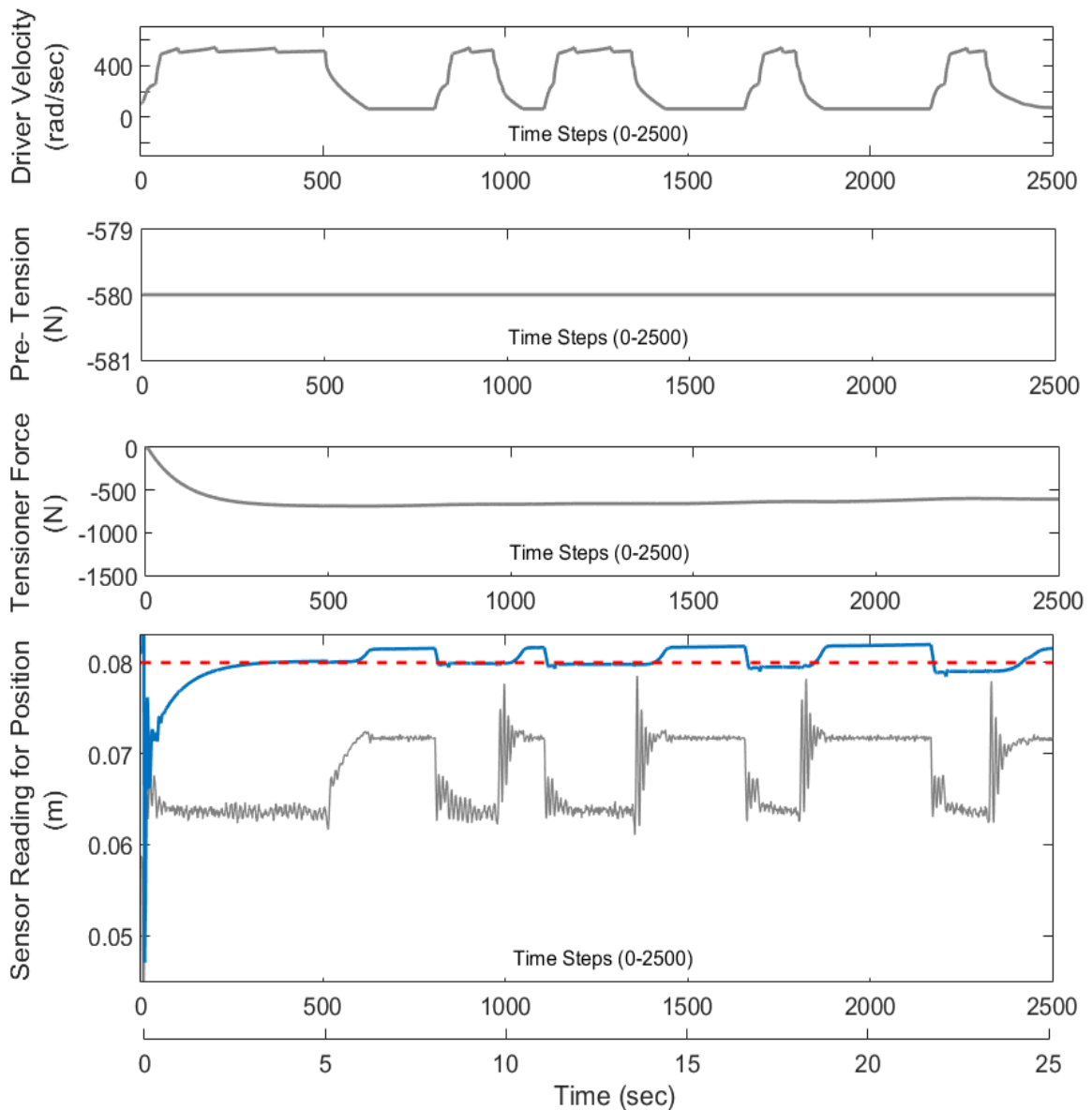


Figure 5.49 Test data 3 for elementary level case studies

The critical part for test data 3 are given as:

- Sudden increase and decrease in the velocity profile is extended.

- The opposite relationship between the driver velocity and the sensor reading for the system without MPC can be seen more obviously in this case.
- The effect of the same opposite relation is less for the system with MPC.
- Although control input seems as if it is constant after three seconds, it is not. The reasons why the control input line looks like this is due to the limits and size of the figure. The following figure demonstrates this reality.

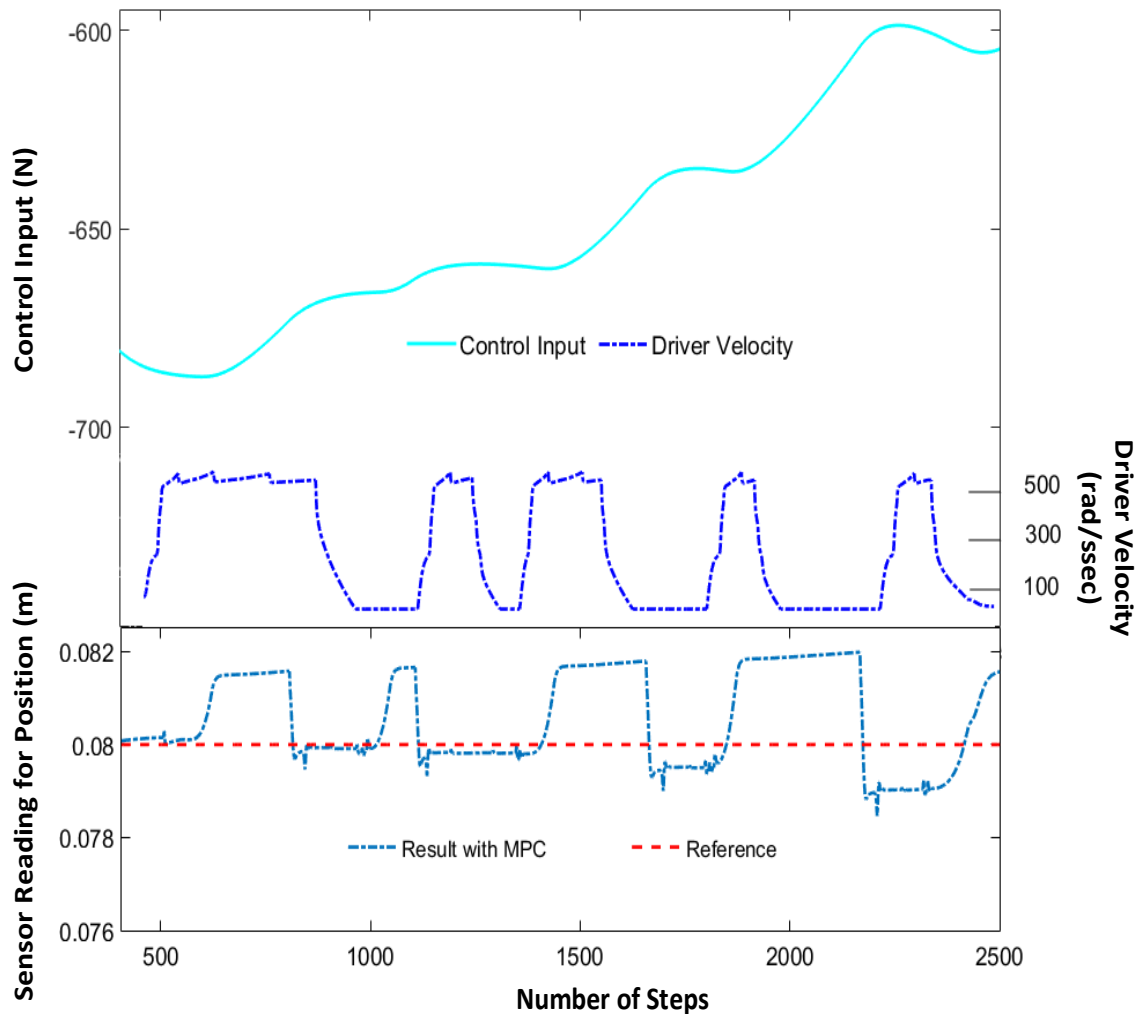


Figure 5.50 Detailed view of test data 3 for elementary level case studies

The important points of the detailed view of test data 3 is shown below:

- It is detailed version for case three.
- As velocity profile increases, the tension in the belt span increases which results in difficulty in the belt movement in the vertical direction. As a consequence, the magnitude of the control input increases.
- Although there is an increase in the applied tensioner force when there is an increase in the velocity profile, the general tendency is in the direction of decrease.

One reason for this situation is the differences between the rate of increase and decrease in the velocity profile; nevertheless, the main reason is the amount of constant velocity time.

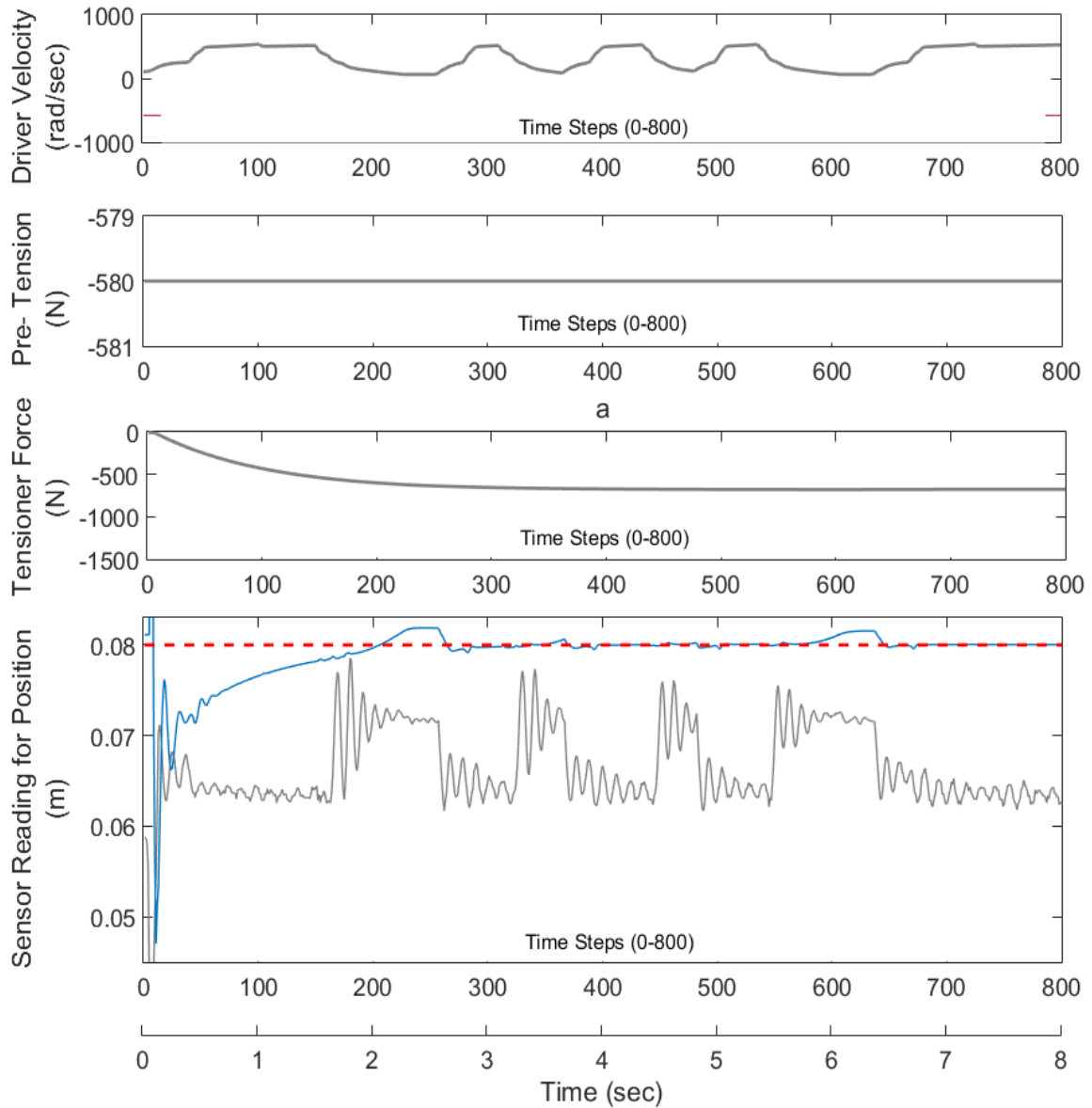


Figure 5.51 Test data 4 for elementary level case studies

The comments on test data 4 is revealed as below:

- The increase in the change rate of the velocity profile is modeled in case 4.
- Reaching the desired value is close to the other cases.
- Similar to the other figures, compensation in case of slackness as in the time range between 2-2.5 seconds and 5.8-6.3 seconds takes time.

In this point it can be beneficial to note that in case of real life applications, there are number of linear actuators who can satisfy the required properties for tensioner reactions preferred in this study. In addition, special designs can be offered by some producers.

5.1.2 Intermediate Level Case Studies

Since each level case studies depends on the different structural components, it is expected to model them separately. However, the same artificial neural networks established for each case study level can be used for different velocity profiles.

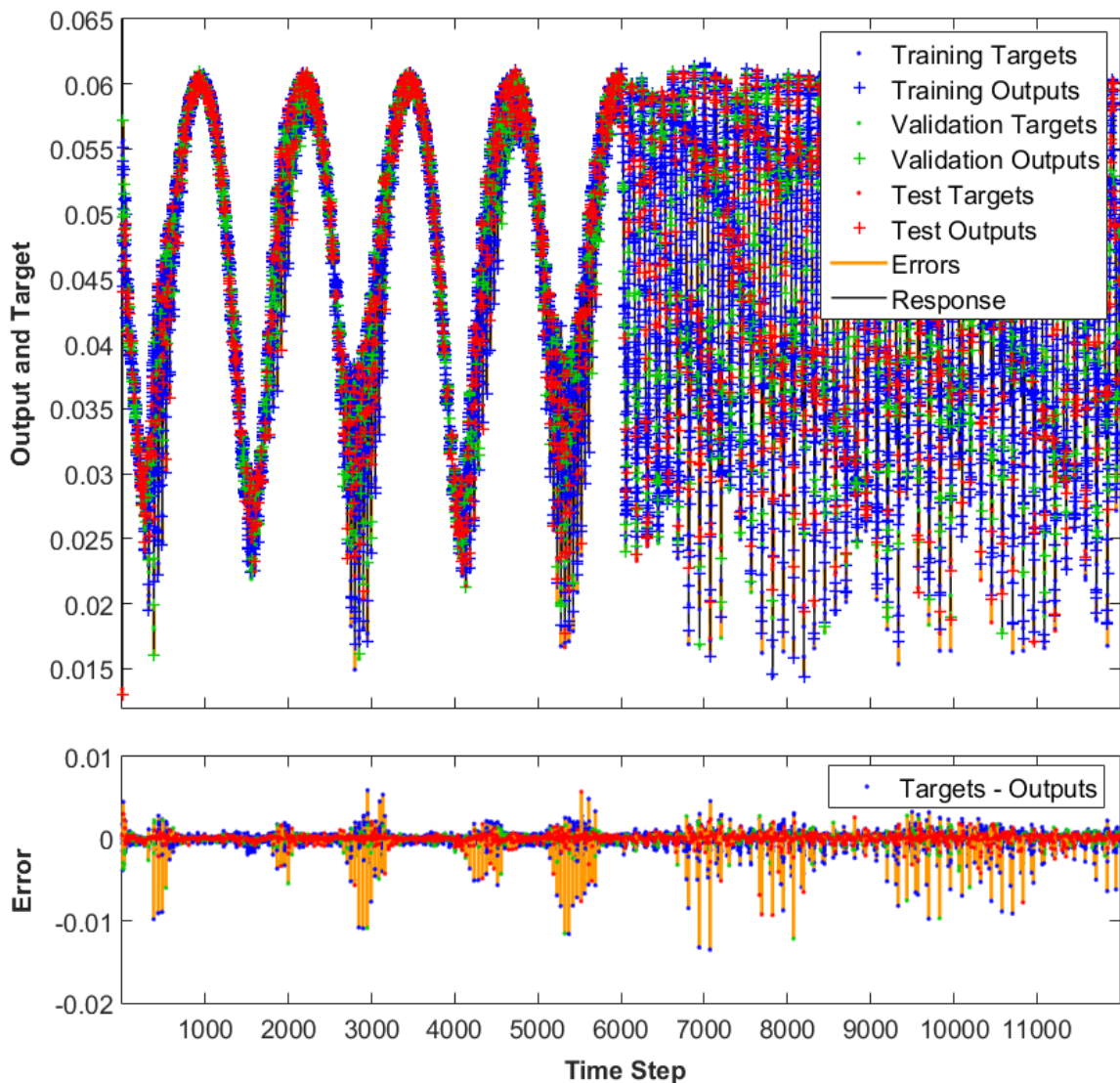


Figure 5.52 Neural network response for intermediate level case studies

The neural network response for the intermediate level case studies is given in Figure 5.51. Similar to the intermediate level case studies, regression plots and performance plots are

shown in the following figures. Closeness of the drive system outputs which are represented by black circles and neural network are compared based on the fitting lines for training, validation and test sets in regression figure. The decrease in the mean square error below $e-006$ shows the degree of model closeness to the belt drive system in the next figure.

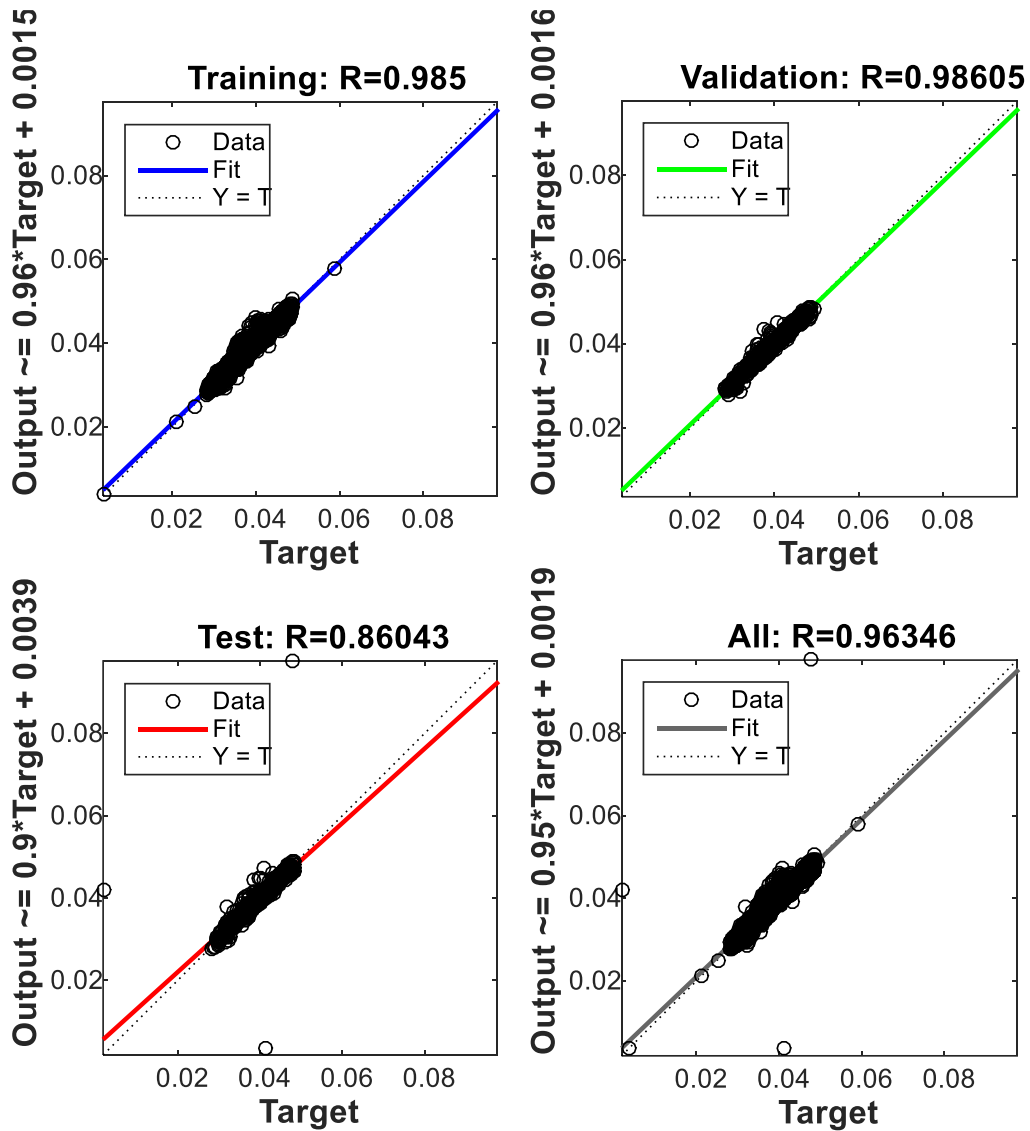


Figure 5.53 Regression plot of the system for intermediate level case studies

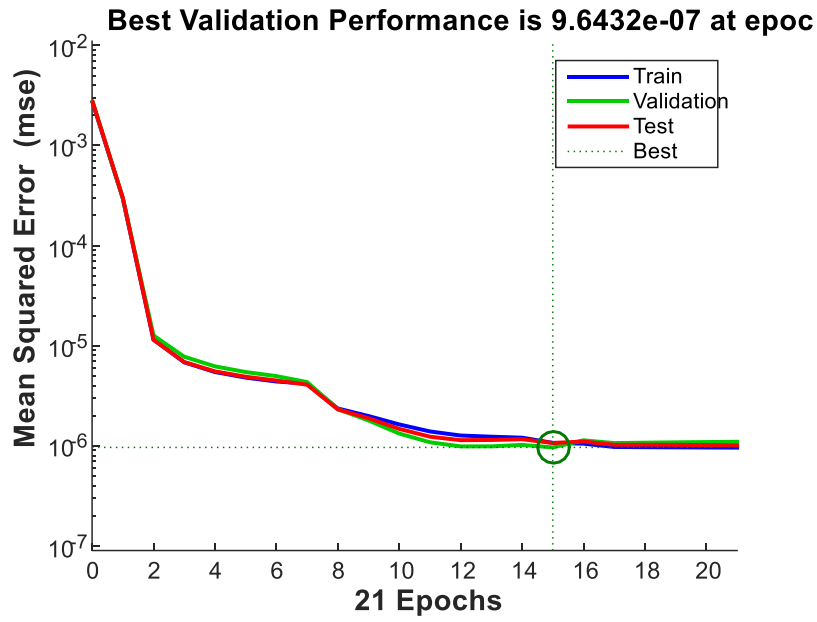


Figure 5.54 Performance values of the system for intermediate level case studies

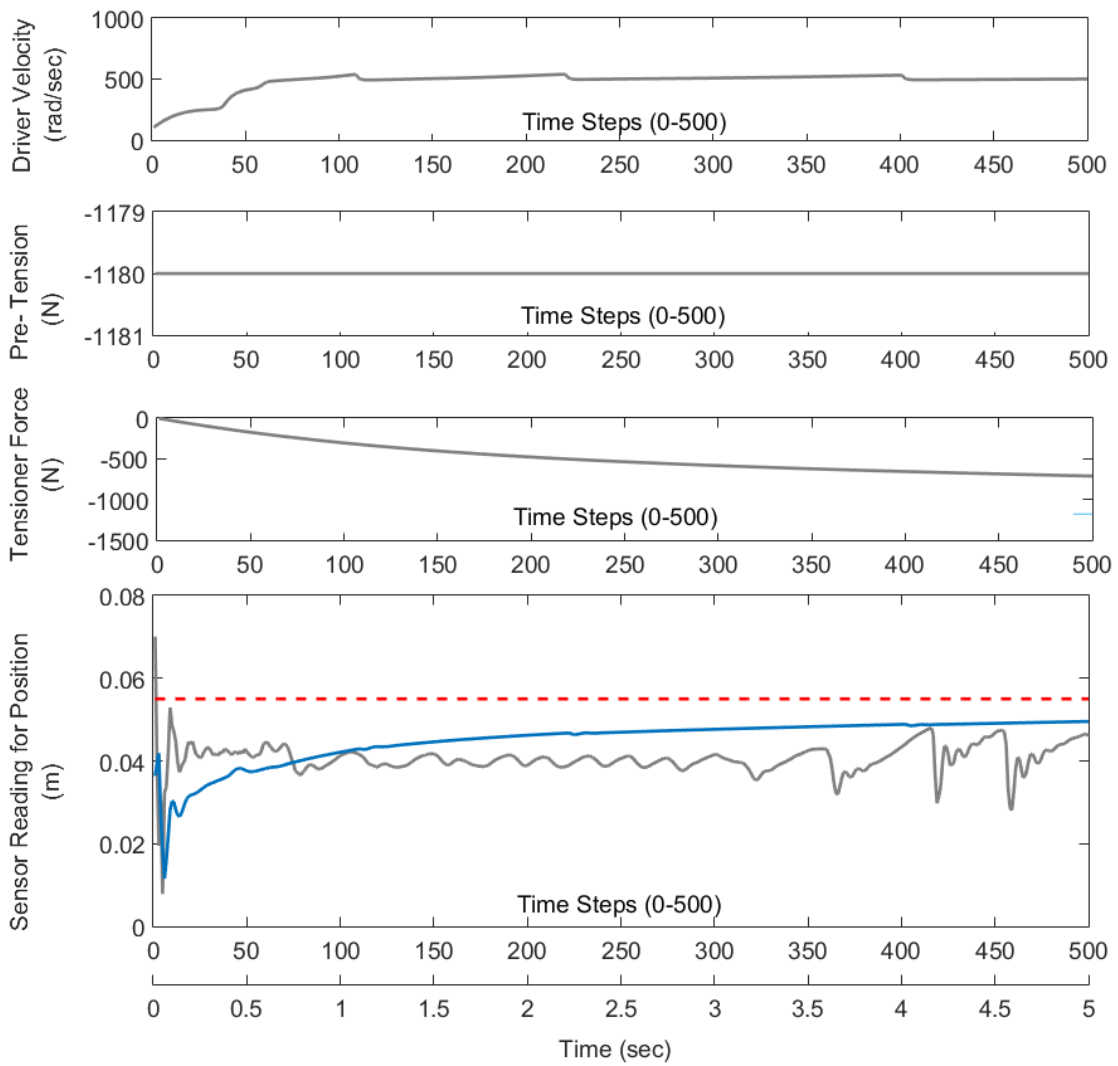


Figure 5.55 Test data 1 for intermediate level case studies

The important results related to the Figure 5. 54 is in below:

- Since the length of the belt is increased in this case, it is expected to increase the belt tension by inspection. This idea was compatible with the trial and error procedure described in the previous chapter. The magnitude of the pre- tension is increased to the 1180 N by trial and error.
- As can be seen from the last graph in the figure, five seconds is not enough time to reach the desired value. However, convergence to the reference line exhibits the success of the MPC.

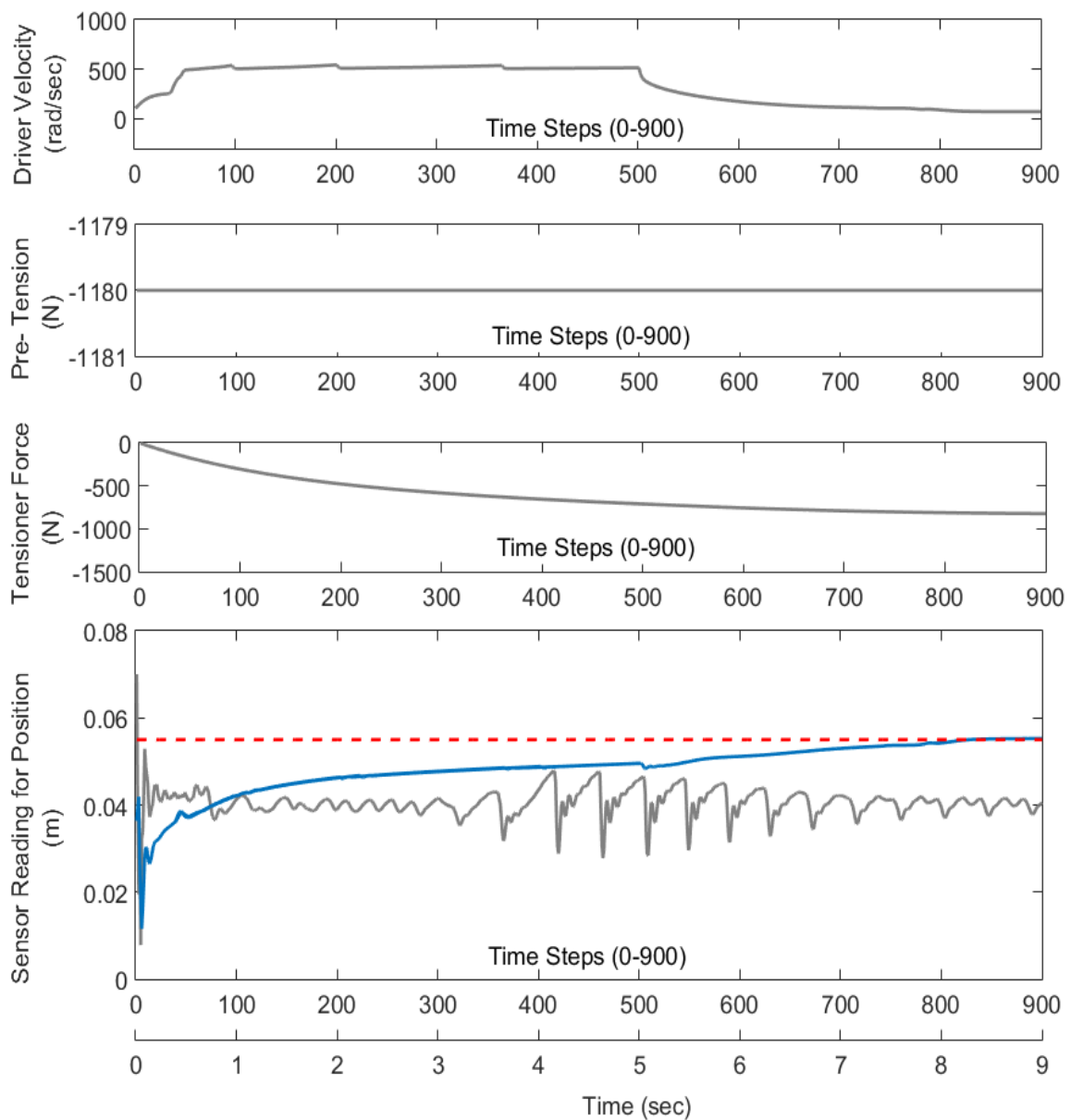


Figure 5.56 Test data 2 for intermediate level case studies

The important consequences of test data 2 for intermediate level case study is summarized as follows :

- Reaching a desired value for the four pulley system takes about eight seconds.
- Owing to the increase in the length and high pre-tension, the tension on the belt is at higher levels compared to the elementary level case studies. Thus, oscillation in the transverse direction is increased. In addition, the effect of increase and decrease of velocity takes time.
- Due to the gear change in the transmission, there are sudden changes in the profile which affects the belt motion characteristic as can be seen from the result without MPC in Figure 5.55 After the last driver velocity drop due to the gear change when car velocity is tried to increase, the belt span vibration goes into a deflection increase trend and it reaches a maximum value after a step change due to the decrease in the car velocity. Although driver velocity is decreasing after five second, the slope of velocity decreases. In other words, the slackness level of the span increases which results in a decrease trend in the oscillation but this oscillation is higher than the initial driver velocity increase trend. The control input cannot reach a fixed value due to this complex oscillation trends but it is increased continuously during the case time.

The crucial points related to the test data 3 plot is as in below:

- It is the extended version of the previous ones. Hence, the effect of driver velocity on the transverse oscillation is more obvious.
- The high pre-tension effect is observed as peg-top like shapes in the transverse belt span movement. The high level of velocity between 8 and 14 seconds results in the small deflection. Although there is a velocity decrease in the range mentioned, the transverse span deflection does not affected due to the high tension in the belt in this range. However, three seconds of low driver velocity profile trigger the increase of belt span deflection up to sudden increase in the velocity profile.

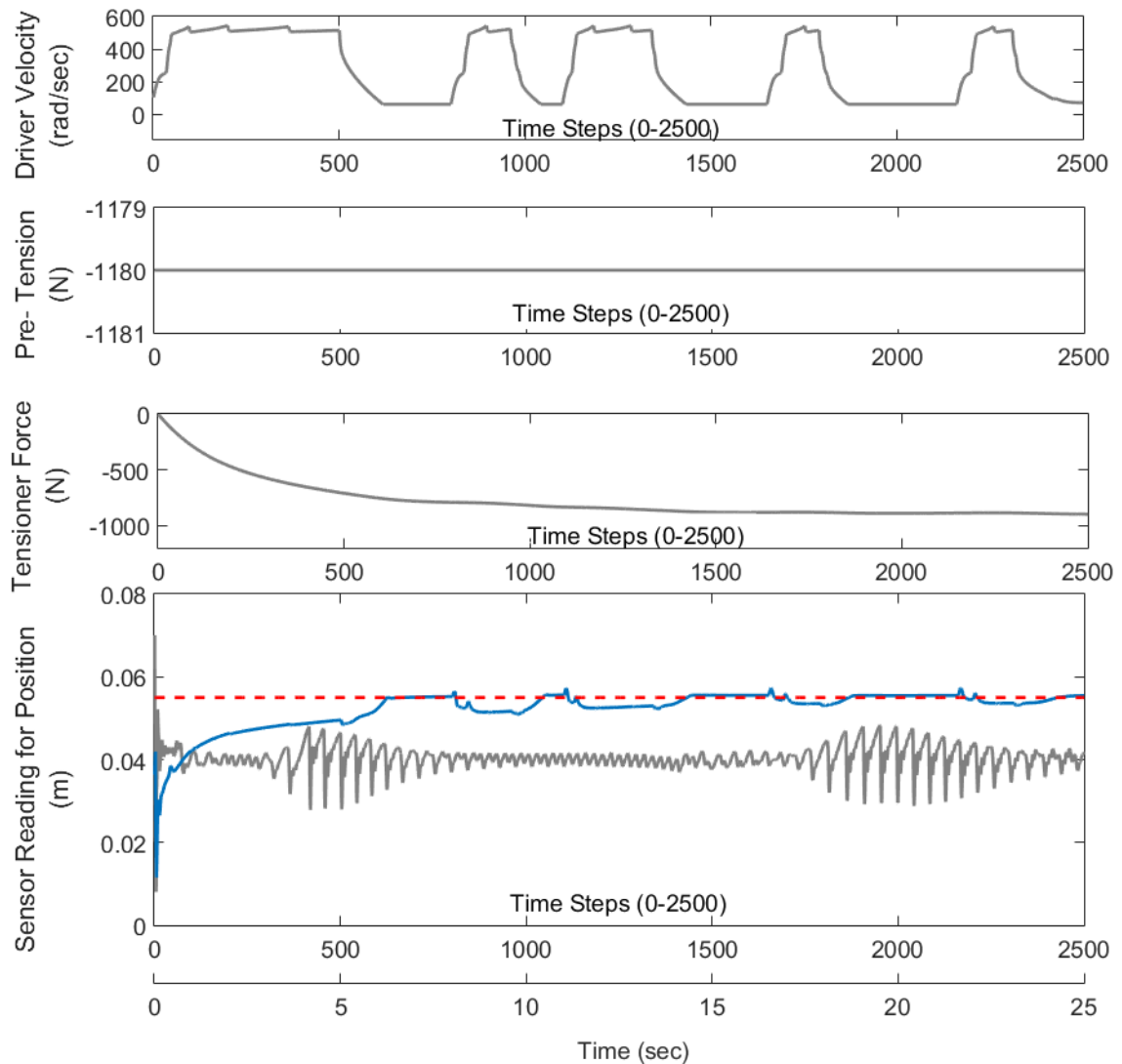


Figure 5.57 Test data 3 for intermediate level case studies

The detailed view results of test data 3 for intermediate level case study is shown below:

- In order to better understand the control input behavior for the case considered in Figure 5.57, the above figure is demonstrated. In order to observe the change in the control input better, both figure is extended and time limit start from five second before which there is a sudden increase in the magnitude of tensioner force which dominate the figure.
- Though there is an increase in the magnitude of the control input through the considered time in the case, there are several decrease in the magnitude to balance the increased tension in the belt span to follow the reference trajectory or line.

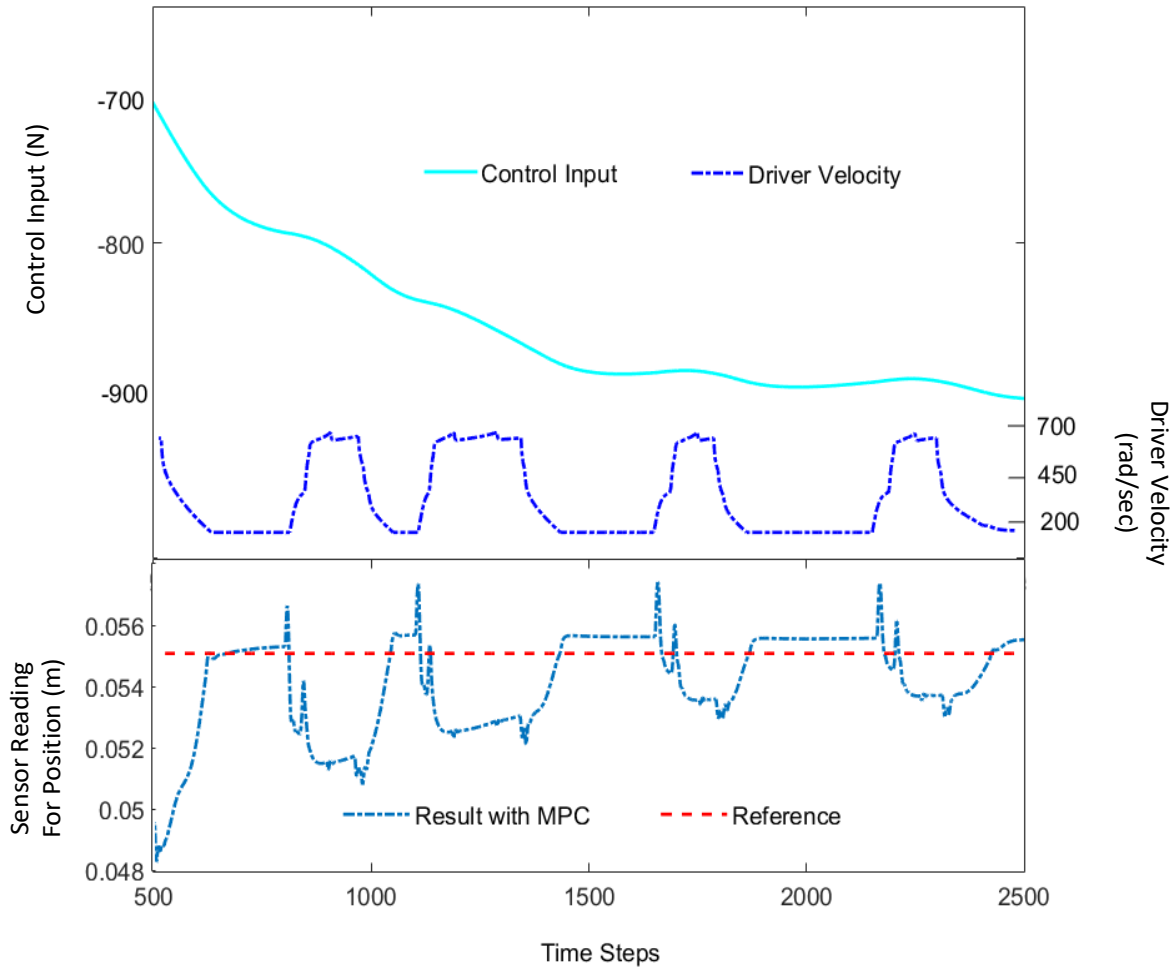


Figure 5.57 Detailed View of Test Data 3 for Intermediate Level Case Studies

- In case of narrower throttle and brake range as shown in Figure 5.58, peg-top like shapes appear earlier and the result with MPC is not as stable as the previous ones because reaction from the controller to the belt drive system follow a pattern which mainly depend on the internal model.

5.1.3 Advanced Level Case Studies

In addition to the complexity in the intermediate level case studies, three millimeter revolute joint axes dislocation is put intentionally to test the neural networks and model predictive controller in a more complex situation. As can be seen from the time series figure of NARX, the response of the system is maximum in terms of span deflection.

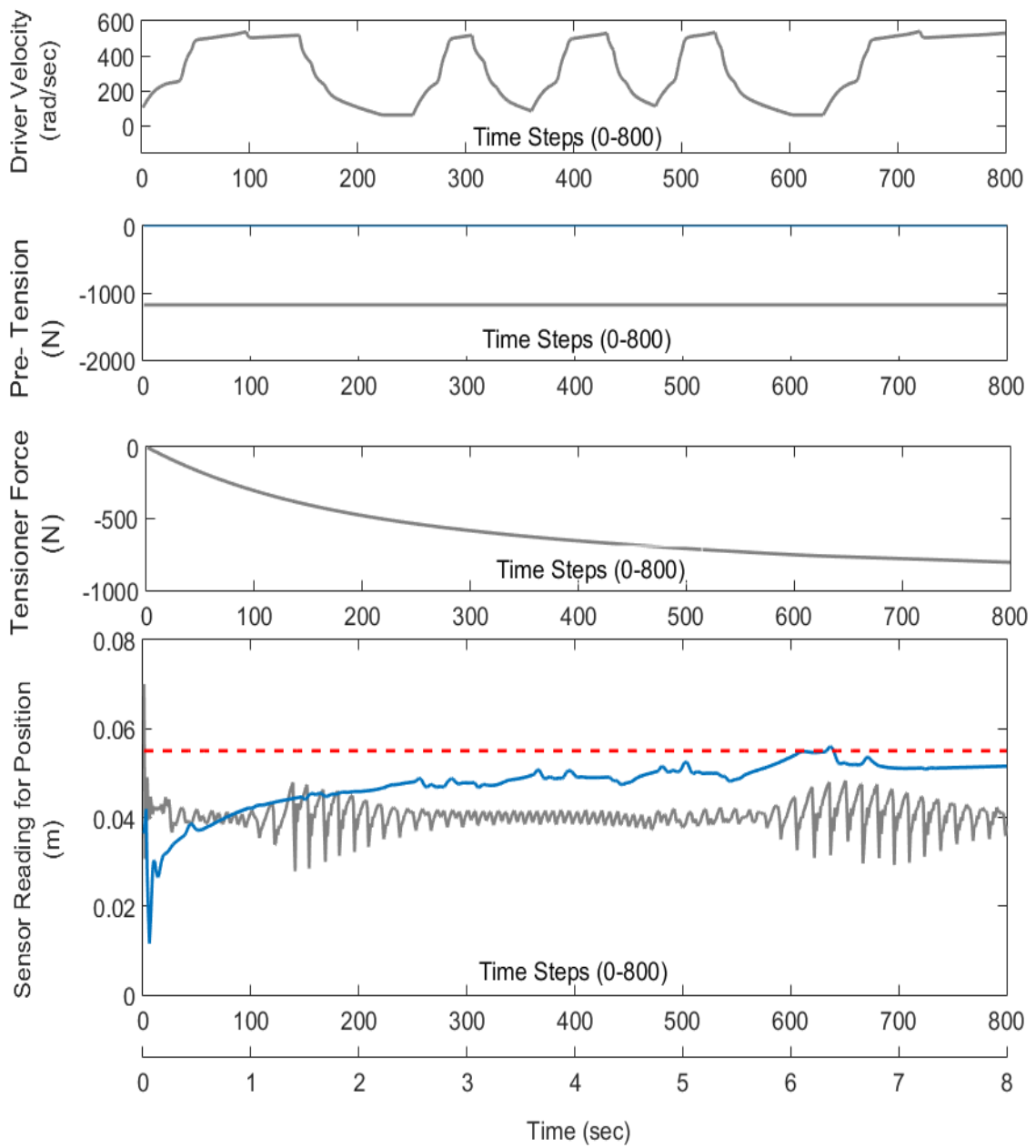


Figure 5.58 Test data 4 for intermediate level case studies

Regression plots and performance plot of the NARX for advanced level case study are demonstrated in the following figures. All of these figures are used to support the neural network model consistency with multi body dynamic results.

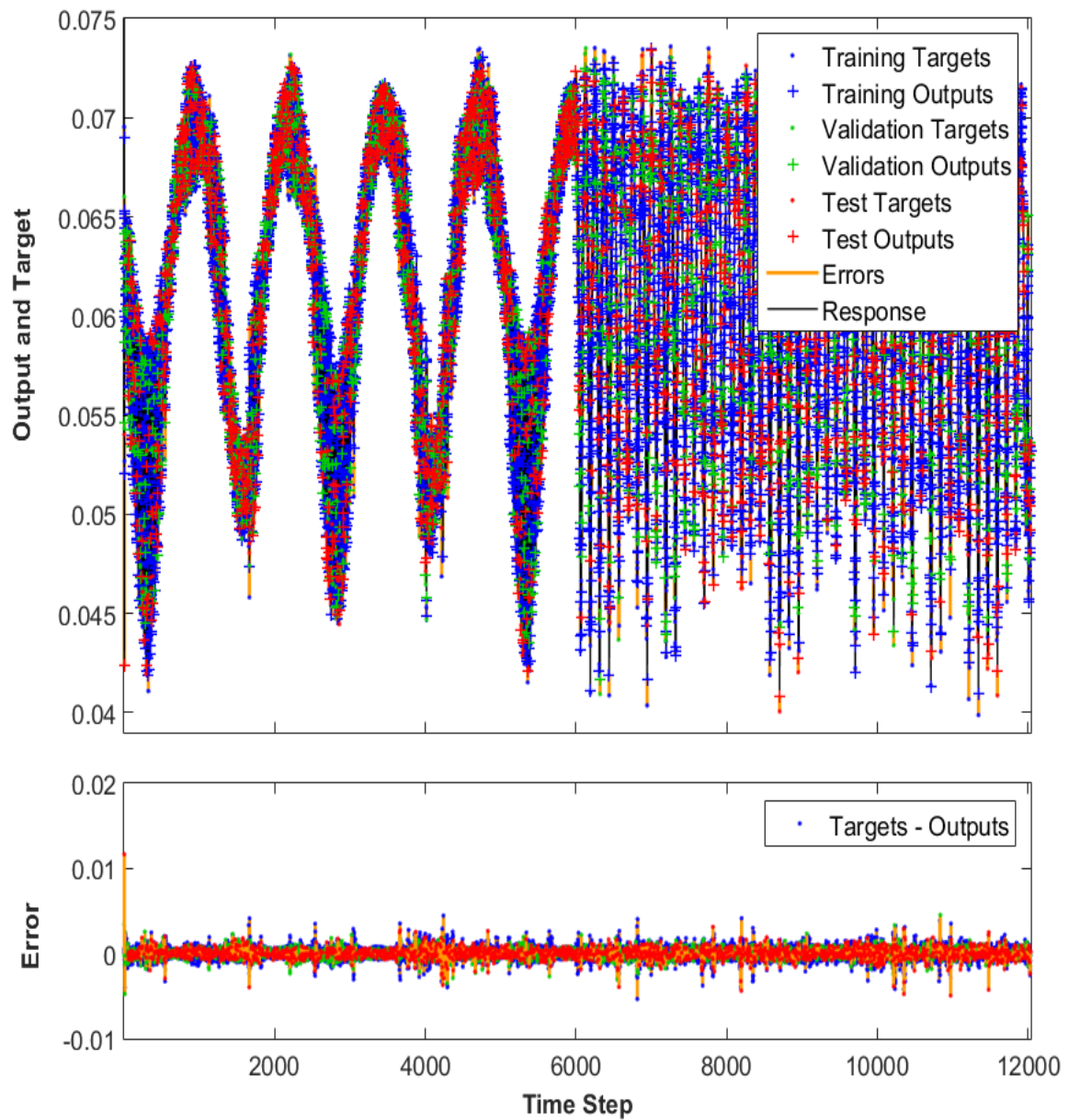


Figure 5.59 Time series response of Narx for advanced level case studies

The critical points of Figure 5. 62 are as follows:

- Pre-Tension applied to the belt is increased about 34% according to the intermediate level case studies in order to maintain the contact between the belt and the pulleys.
- It is expected that as driver velocity increases, the transverse vibration of the belt span is restricted due to the increased tension in the span and whenever enough

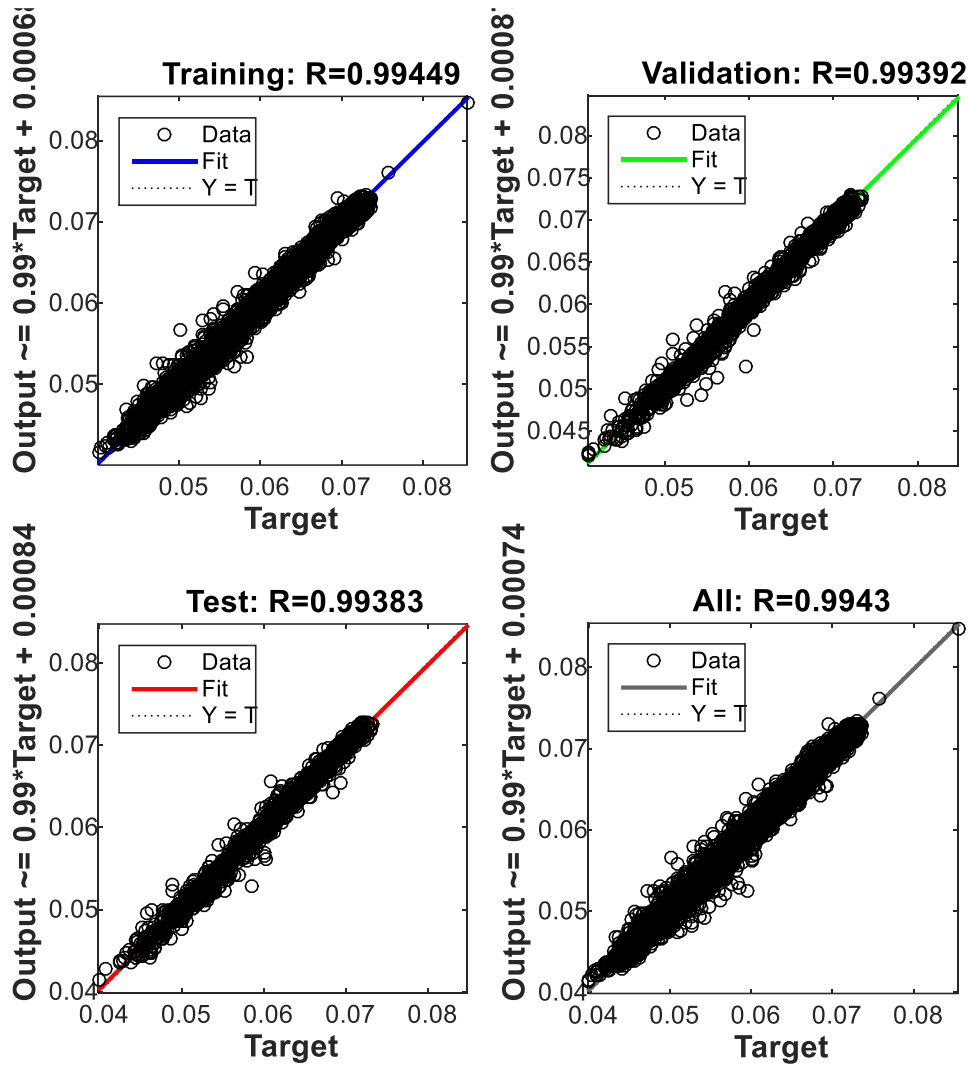


Figure 5.60 Regression plot of the system for advanced level case studies

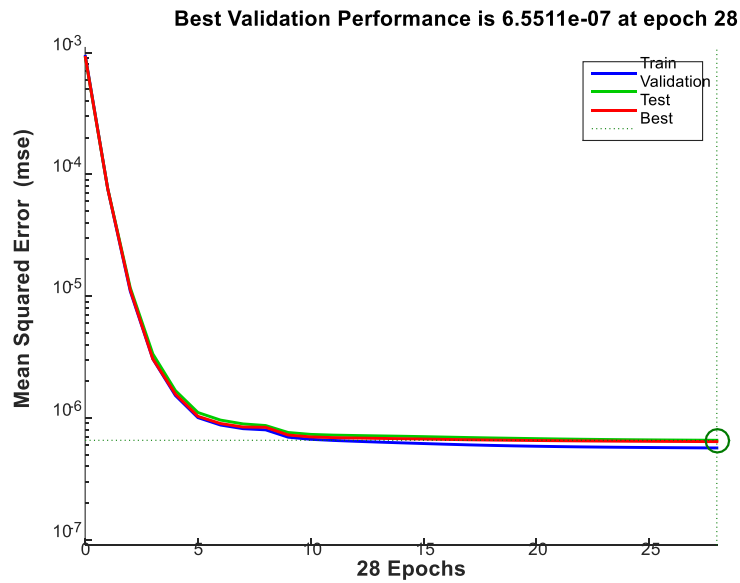


Figure 5.61 Performance values of the system for advanced level case studies

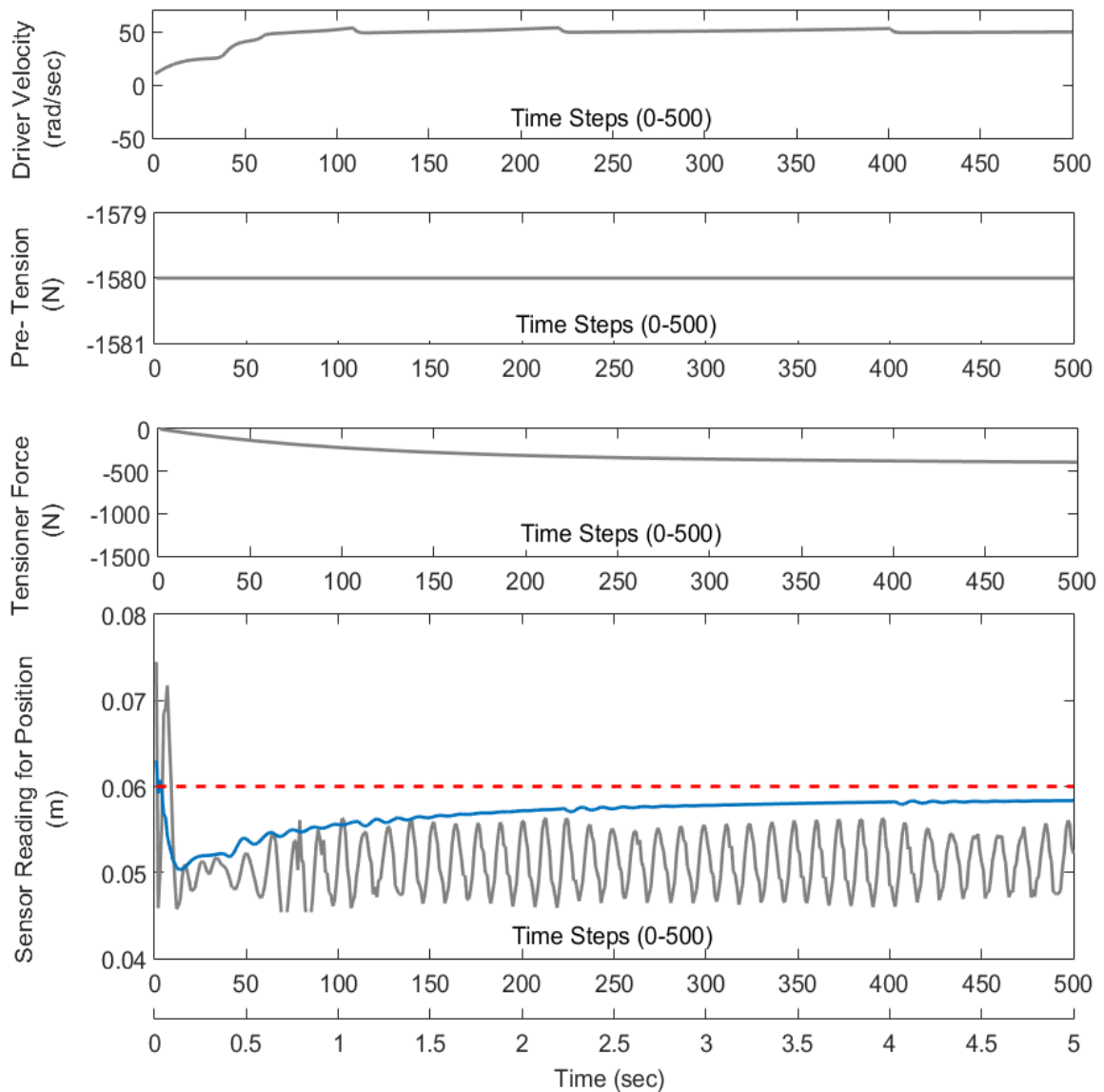


Figure 5.62 Test data 1 for advanced level case studies

amount of decrease in the driver velocity occurs, span tries to move in a more flexible environment due to the decreased belt tension. However, this time there is a 3 mm revolution axis dislocation for the driver pulley. It means that as velocity increases the effect of the dislocation will be felt more. In other words, a more complex belt motion is expected.

- Increase in the driver velocity determines the degree of change in the span motion. Gear change affects the velocity profile little as in Figure 5.62
- Although driver velocity in the advanced level case is decreased to the one tenth of the elementary and intermediate level case studies' and reference line fixed only one

centimeter above from the average position of the span, belt drive system response is under the reference line due to the complex behavior of the span in this case.

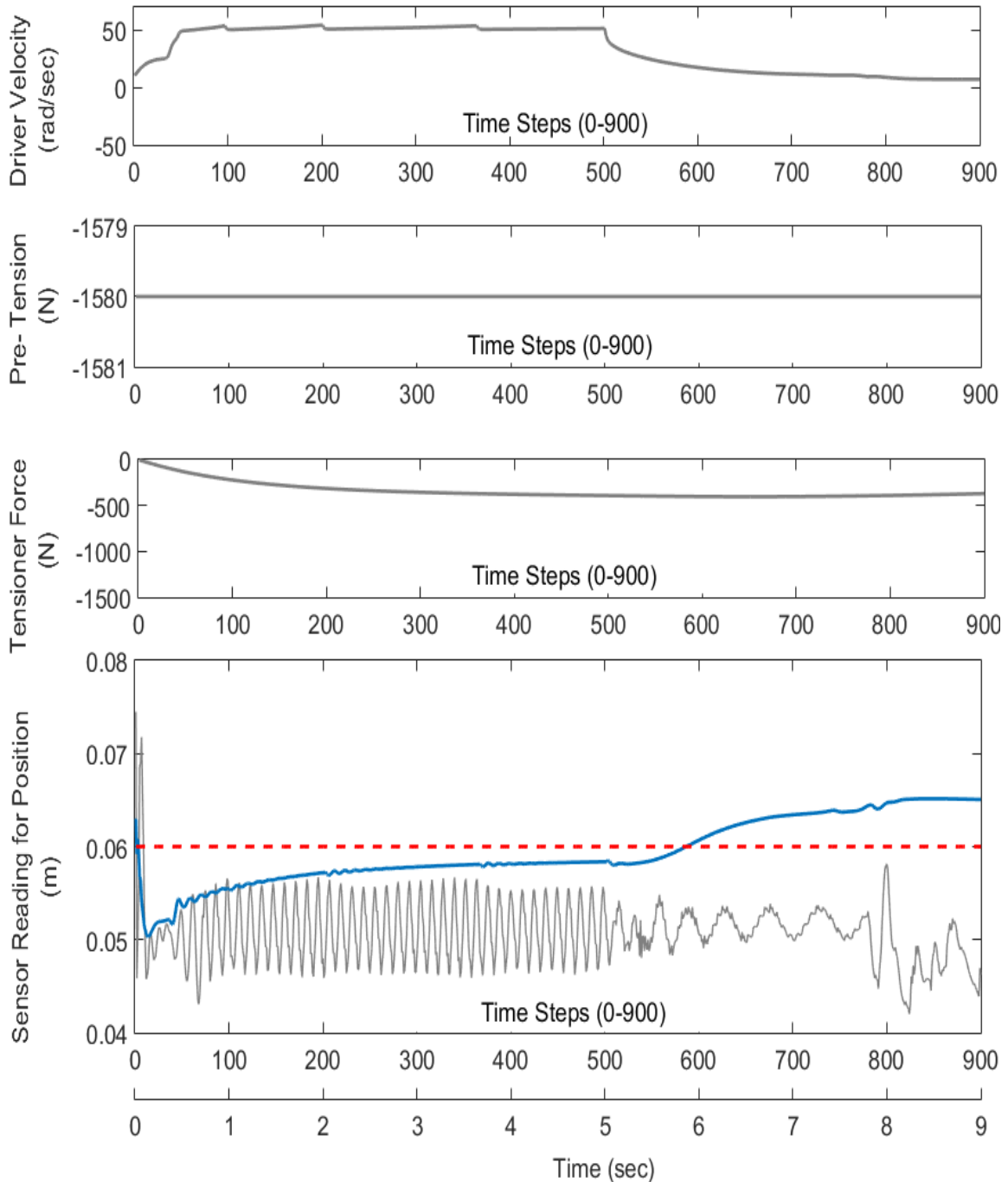


Figure 5.63 Test data 2 for advanced level case studies

Figure 5.63 can be summarized as:

- As car decelerate, the position of the span deflection decreases and the frequency of the up and down motion reduces.

- Further decrease in the driver velocity gives the span more area to move. Thus, comparatively great amount of position change around eight second is reasonable. Upper span motion of the belt after the sudden up and down motion around eighth second moves irregularly due to the freedom of reduced belt tension.
- Although convergence to the reference line seems slow, it is the 2.1 second that result with MPC is only 2 mm away from the reference. Reaching the reference line takes 5.8 seconds; however, the sudden decrease in the driver velocity makes the job of the controller difficult as decrease in the driver velocity goes on. In this point, it should be also put into account that after the start of the decrease in the driver velocity, the slope of the result with MPC is improved which means that it is possible to think the positive effect of the decrease in the driver velocity on closing to the desired value in the initial decrease phase.

The summary of Figure 5.64 can be demonstrated with following comments as:

- How the increase and the decrease in the driver velocity affects the span motion may be understood better by using the extended graph shown in Figure 5.64
- Since the pre-tension applied to the belt is at the higher limits which is 2.72 times of the elementary level case and 1.34 times of the intermediate level case studies, the tension on the belt restricts the motion of the belt to the upper side. That is why decrease in the driver velocity causes increase in the upper belt span unlike the elementary and intermediate level case studies. In other words, position reading from the sensor goes a general decreasing trend.
- However, seeking from the tensioning perspective, the logical response sequence from the controller can be seen better because belt motion is directly affected from the tension given to the system. There are two main reasons for the change in the upper belt span tensioning in case of not using a controller and considering not to change the pre-tension which is decided in the initial step in real life too.

Although they are both due to the velocity profile, considering them separately but not independent of velocity ease the observation. One of the reason is tension due to the stretching and releasing effect of velocity change and the other one is the change in the contact position of the belt with pulley due to the crooked driver pulley center of rotation. It is clear that up and down motion occurs in a larger scale in case of increase in the driver velocity. In addition, sudden decrease and increase in the driver velocity influence the belt tensioning strongly which makes reaching the desired position of the mid-span harder because of the high tension on the upper belt span.

In order to govern this difficulty, the controller gives a freedom to the belt span to reduce the effect of high tension by decreasing the magnitude of the tensioner force. This reasoning is valid on both the general behavior and the sudden changes in the velocity profile as can be seen from the Figure 5.64.

- It is not expected in advanced level case studies to reach and stay at the desired value continuously since there is a driver pulley behaving as if it is a cam whose effect on the upper belt span is hard to model perfectly. Yet even in case of narrower throttle and brake range, it is possible to follow desired level as shown in Figure 5.66.

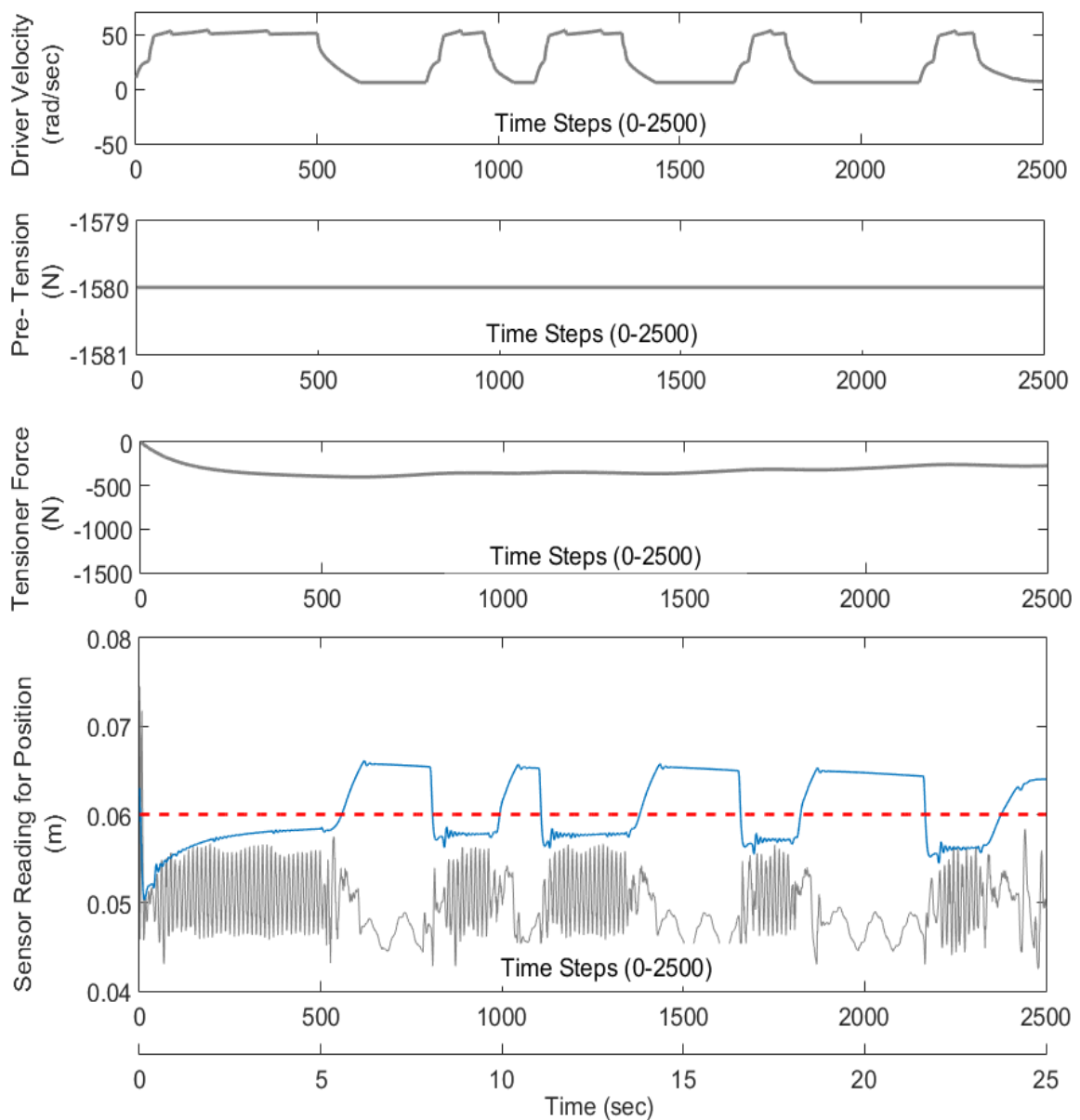


Figure 5.64 Test data 3 for advanced level case studies

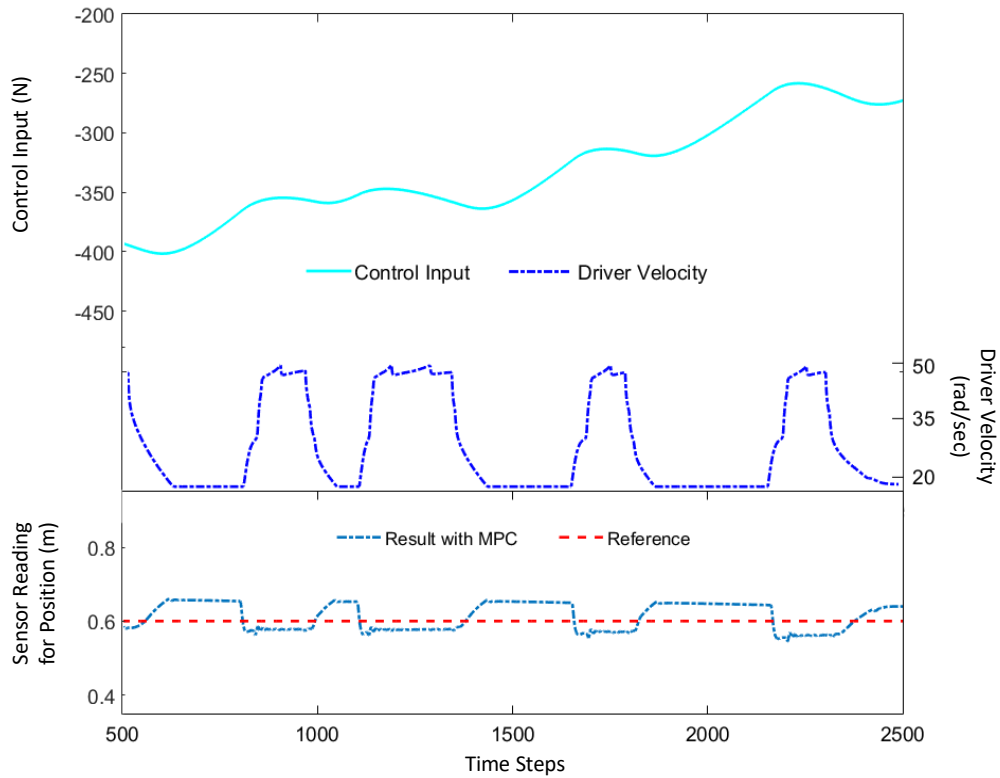


Figure 5.65 Test data 4 for advanced level case studies

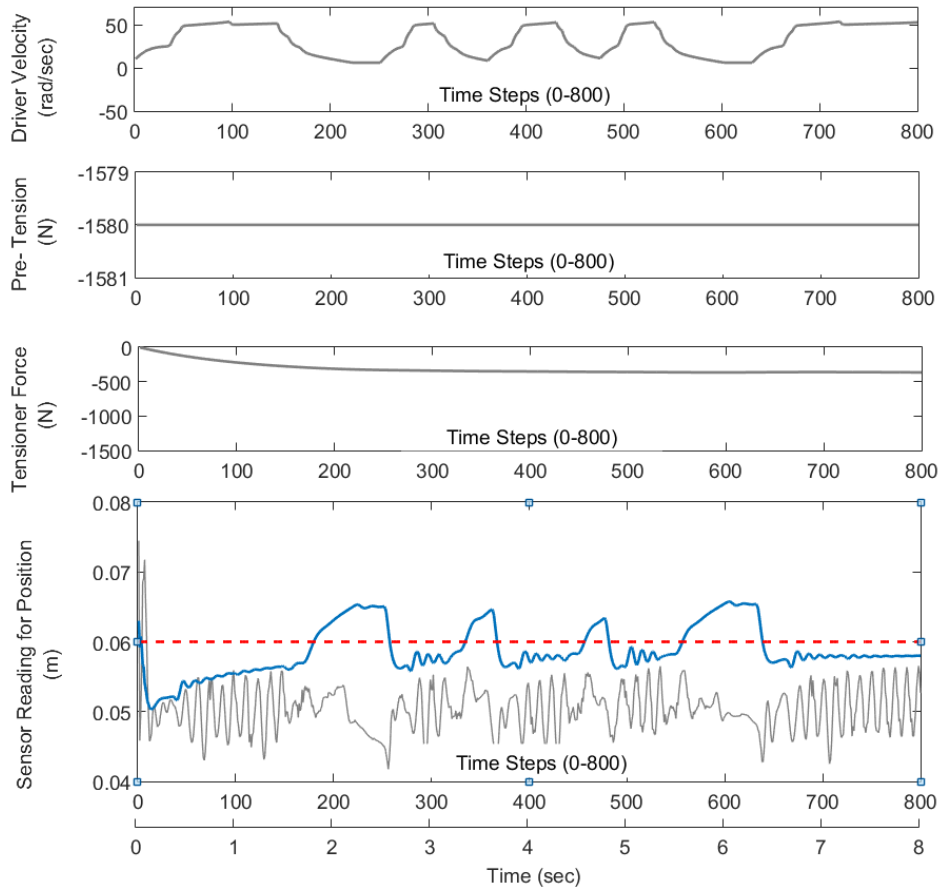


Figure 5.66 Test Data 4 for Advanced Level Case Studies

CHAPTER 6: CONCLUSION

Transverse vibration of a serpentine belt system, whose numeric derivation is really hard and time consuming, is modeled with the help of a multibody dynamic simulation program, Recurdyn. The belt system studied in this work includes basic components of a serpentine belt drive model which are driver pulley, driven pulley, tensioner and belt. After a multibody dynamic system is constructed, inputs are chosen properly so that transient behavior of belt in the drive system can be observed. The next step is to collect data and build a plant model with the help of Neural Network (NN) toolbox after a modification on NN codes in Matlab. Then, this plant is linearized to use in MPC. Finally, designed MPC is used in Recurdyn. The comparison between the results with MPC and without MPC demonstrated the success of this study. In addition to the parametric study done in MPC design procedure, another parametric study is done for PID design. These two types of controller are compared and results show that MPC is a better controller choice.

The critical contribution of this study can be summarized as follows:

- Neural Networks are used to model the complex (*transient state*) belt behaviours,
- Tensioner position is changed according to the desired input,
- Transverse vibration of the belt span is decreased by using MPC,
- Co-simulation with Model Predictive Controller challenge is solved

All in all, the tensioner of a belt-pulley system is controlled with a model predictive controller by using artificial neural networks in this study. It is clear that the use of belt-drive system in automotive industry will decrease as electric motors becomes commonly used type of drive choice. However, it should be noted that although use of electric motors are increasing day by day, the use of belt drive system continues due to the range limitation for distance travelled, cost and so on. Moreover, the use of this study cannot be restricted to the vehicle drive systems like vehicle accessory belt and trigger belt drives. The area of this study is as wide as the field of belt drive concept. Therefore, this study can be beneficial for both automotive industry and the other industries using belt-drive systems. Since this study reveals the success of using MPC, working on real world set-up will be reasonable.

REFERENCES

- [1] Kong, L. (2003). *COUPLED BELT-PULLEY MECHANICS IN SERPENTINE BELT DRIVES* (Ph.D). The Ohio State University.
- [2] Wagner, J., Paradis, I., Marotta, E., & Dawson, D. (2002). Enhanced automotive engine cooling systems - a mechatronics approach. *International Journal Of Vehicle Design*, 28(1/2/3), 214.
- [3] Solidur, P. (1999). *U.S. Patent No. 09/332,751*. U.S.
- [4] ContiTech (2017). Retrieved 5 January 2017, from http://aam.europe.contitech.de/pages/downloads/docs/Riemen_Komponenten_en.pdf
- [5] Rouge, B. (2017). AGCO Automotive Repair Service - Baton Rouge, LA - *Detailed Auto Topics - Serpentine Belt Tensioners*. Retrieved 19 March 2017 from http://www.agcoauto.com/content/news/p2_articleid/348.
- [6] ANSYS (2017). *Multibody Dynamic*. Retrieved 08 April 2017, from <http://www.ansys.com/-/media/Ansys/corporate/resourcelibrary/article/AA-V2-I2-Multibody-Dynamics.pdf>
- [7] Theoretical Manual of Recurdyn
- [8] Camacho, E., & Bordons, C. (2007). *Model Predictive control*. London: Springer London.
- [9] Hagan, M., Demuth, H., Beale, M., & De Jesús, O. (2014). *Neural network design* (2nd ed.). [S. l.: s. n.].
- [10] Muttiah, R. (2002). *From Laboratory Spectroscopy to Remotely Sensed Spectra of Terrestrial Ecosystems* (p. 151). Dordrecht: Springer Netherlands.
- [11] Essien, E. (2017). *ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) – BASED MODEL PREDICTIVE CONTROL (MPC) FOR CARBON DIOXIDE REFORMING OF METHANE (CDRM) IN A PLUG FLOW TUBULAR REACTOR FOR HYDROGEN PRODUCTION* (Graduate). University of Regina.
- [12] Grüne, L., & Pannek, J. (2011). *Nonlinear model predictive control*. Cham: Springer.
- [13] Szczerbicki, E. (2013). *Intelligent Systems for Knowledge Management*. Berlin: Springer Berlin.

APPENDIX:

Neural Network Code

```
% Solve an Autoregression Problem with Recurdyn Input with a NARX Neural
% Network for Recurdyn Belt Drive System Model
% This script assumes these variables are defined:
% (The below datas are gathered from Recurdyn)
% kk_RecurDyn_inputs_karma_forced - input time series./FREOM RECURDYN
% kk_RecurDyn_outputs_karma_forced_1 - feedback time series./FROM RECURDYN

X = tonndata(kk_RecurDyn_inputs_karma_forced,false,false);
T = tonndata(kk_RecurDyn_outputs_karma_forced_1,false,false);

% The training function is chosen by trial and error. 'trainlm' is the best one in terms of both being fast and
% more accurate
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

inputDelays = 1:4;
feedbackDelays = 1:4;
hiddenLayerSize = 20;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);
net.sampleTime=0.01;% If neural network toolboxes with nnstart are used
% directly, the sample time cannot be changed at training;however, with this
% script timestep of NN and the simulation program is equated
% Choose Input and Feedback Pre/Post-Processing Functions
% Settings for feedback input are automatically applied to feedback output
% For a list of all processing functions type: help nprocess
% Customize input parameters at: net.inputs{i}.processParam
% Customize output parameters at: net.outputs{i}.processParam
net.inputs{1}.processFns = {'removeconstantrows','mapminmax'};
net.inputs{2}.processFns = {'removeconstantrows','mapminmax'};

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,X,{},T);

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'time'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFns = {'plotperform','plottrainstate','ploterrhist', ...
```

```

    'plotregression', 'plotresponse', 'ploterrcorr', 'plotinerrcorr'});

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net);

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, ploterrhist(e)
% figure, plotregression(t,y)
% figure, plotresponse(t,y)
% figure, ploterrcorr(e)
% figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the outout layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,X,{ },T);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(net,tc,yc)

% Multi-step Prediction
% Sometimes it is useful to simulate a network in open-loop form for as
% long as there is known output data, and then switch to closed-loop form
% to perform multistep prediction while providing only the external input.
% Here all but 5 timesteps of the input series and target series are used
% to simulate the network in open-loop form, taking advantage of the higher
% accuracy that providing the target series produces:
numTimesteps = size(x,2);
knownOutputTimesteps = 1:(numTimesteps-5);
predictOutputTimesteps = (numTimesteps-4):numTimesteps;
X1 = X(:,knownOutputTimesteps);
T1 = T(:,knownOutputTimesteps);
[x1,xio,aio] = preparets(net,X1,{ },T1);
[y1,xfo,afo] = net(x1,xio,aio);
% Next the the network and its final states will be converted to
% closed-loop form to make five predictions with only the five inputs
% provided.

```

```

x2 = X(1,predictOutputTimesteps);
[netc,xic,aic] = closeloop(net,xfo,afo);
[y2,xfc,afc] = netc(x2,xic,aic);
multiStepPerformance = perform(net,T(1,predictOutputTimesteps),y2)
% Alternate predictions can be made for different values of x2, or further
% predictions can be made by continuing simulation with additional external
% inputs and the last closed-loop states xfc and afc.

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,X,{},T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(nets,ts,ys)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x,xi,ai);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    x1 = cell2mat(x(1,:));
    x2 = cell2mat(x(2,:));
    xi1 = cell2mat(xi(1,:));
    xi2 = cell2mat(xi(2,:));
    y = myNeuralNetworkFunction(x1,x2,xi1,xi2);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
% to generate simulink block
gensim(net)
%to see the performance of NN in terms of MSE(mean square error)
plotperform(net)
%to see how much fit is good
plotregression(net)
%to see how much fit is good
plotresponse(net)

```


Matlab Code for Co-Simulation

```
addpath('C:\Program Files\FunctionBay, Inc\RecurDyn V8R4\Toolkits\CoSim_Simulink');
%addpath('C:\Program Files\FunctionBay, Inc\RecurDyn V8R4\Toolkits\Controls\Matlab');
RecurDyn='C:\Program Files\FunctionBay, Inc\RecurDyn V8R4\Bin\';
RecurDyn_CoSim='C:\Program Files\FunctionBay, Inc\RecurDyn V8R4\Toolkits\CoSim_Simulink\';
RecurDyn_model='Model4_2';
r_temp___=size(RecurDyn_model);
RecurDyn_model_n=r_temp___(2);
Output_File='Model4_2';
r_temp___=size(Output_File);
Output_file_n=r_temp___(2);
if(exist([RecurDyn_model,'.rdyn']))
    RecurDyn_inputs = 'PRE_force!Tensioner_force!Driver_Vel';
    RecurDyn_outputs = 'Dist1!Dist2!Dist3!Dist4!Dist5!Dist6!Dist7!Tension1';
    RecurDyn_io_ids = [ 1 2 3 4 5 6 7 8 9 10 11 ];
    RecurDyn_controltimestep = 1.e-002;
    Plant_inputs = Rearrange_io( RecurDyn_inputs );
    Plant_outputs = Rearrange_io( RecurDyn_outputs );
    r_temp___=size(Plant_inputs);
    Plant_inputs_num=r_temp___(1);
    r_temp___=size(Plant_outputs);
    Plant_outputs_num=r_temp___(1);
    r_temp___=version;
    Matlab_version=str2double(r_temp___(1));
    disp("");
    disp('===== RecurDyn & Matlab/SIMULINK =====');
    disp('%%% INFO : RecurDyn plant actuators names :');
    disp([int2str([1:size(Plant_inputs,1)]),blanks(size(Plant_inputs,1)),Plant_inputs]);
    disp('%%% INFO : RecurDyn plant sensors names :');
    disp([int2str([1:size(Plant_outputs,1)]),blanks(size(Plant_outputs,1)),Plant_outputs]);
    disp('=====');
    % disp('***** Reserved Variables *****');
    % disp(' * Plant_inputs,Plant_outputs,RecurDyn,RecurDyn_model,RecurDyn_static*');
    % disp(' * RecurDyn_inputs,RecurDyn_io_ids,RecurDyn_outputs');
    % disp(' * RecurDyn_controltimestep,RecurDyn_show,RecurDyn_step');
    % disp('***** Reserved variables can not be changed *****');
    disp(' ');
else
    disp("");
    disp('%%% ERROR : missing RecurDyn plant model file !!!');
    disp("");
end
clear r_temp___;
```

Matlab Code for Sensor-Order

```
A1=RecurDyn_outputs(:,1);
B1=zeros(110,1);
for i=1:109
    fark=A1(i+1,1)-A1(i,1);
    B1(i,1)=fark;
end
Max1= max(B1);
Min1= min(B1);
K1=Max1-Min1;
%%
A2=RecurDyn_outputs(:,2);
B2=zeros(110,1);
for i=1:109
```

```

    fark=A2(i+1,1)-A2(i,1);
    B2(i,1)=fark;
end
    Max2= max(B2);
    Min2= min(B2);
    K2=Max2-Min2;
    %%
    A3=RecurDyn_outputs(:,3);
    B3=zeros(110,1);
    for i=1:109
        fark=A3(i+1,1)-A3(i,1);
        B3(i,1)=fark;
    end
    Max3= max(B3);
    Min3= min(B3);
    K3=Max3-Min3;
    %%
    A4=RecurDyn_outputs(:,4);
    B4=zeros(110,1);
    for i=1:109
        fark=A4(i+1,1)-A4(i,1);
        B4(i,1)=fark;
    end
    Max4= max(B4);
    Min4= min(B4);
    K4=Max4-Min4;
    %%
    A5=RecurDyn_outputs(:,5);
    B5=zeros(110,1);
    for i=1:109
        fark=A5(i+1,1)-A5(i,1);
        B5(i,1)=fark;
    end
    Max5= max(B5);
    Min5= min(B5);
    K5=Max5-Min5;
    %%
    A6=RecurDyn_outputs(:,6);
    B6=zeros(110,1);
    for i=1:109
        fark=A6(i+1,1)-A6(i,1);
        B6(i,1)=fark;
    end
    Max6= max(B6);
    Min6= min(B6);
    K6=Max6-Min6;
    %%
    A7=RecurDyn_outputs(:,7);
    B7=zeros(110,1);
    for i=1:109
        fark=A7(i+1,1)-A7(i,1);
        B7(i,1)=fark;
    end
    Max7= max(B7);
    Min7= min(B7);
    K7=Max7-Min7;
    %%
    K=[K1,K2,K3,K4,K5,K6,K7];
    Kmin=min(K);
    Ksum=sum(K);

```

```

K_order=K/Kmin;
sum_K_order=sum(K_order);
coefficient=K_order/sum_K_order;
%%% validation = == => sum(K_coeff)=1
validation=sum(coefficient);

```

(If sensors are given randomly in the simulation program, the above code will be good enough. Otherwise, using 3-D matrix will be better)

Matlab Code for Parametric Study of MPC

```

%Input name is defined
plant.InputName = {'Force'};
%Output name is defined
plant.OutputName = {'Deflection'};
%Timestep is adjusted so that MPC reaction to the plant output is well enough
%Sampling period
Ts = 0.01;
% prediction horizon
p = 5;
% control horizon
m = 2;
mpc_for_NN = mpc(plant,Ts,p,m);
% The parameters mentioned above is changed to get better results.
% Constraints are given as follows- they depends on the observation of the system behaviour
mpc_for_NN.MV = struct('Min',{0.055},'Max',{0.085},'RateMin',{-0.005});
% Weights on manipulated and controlled variables.
mpc_for_NN.Weights = struct('MV',[0],'MVRate',[.001 ],'OV',[1]);
% Increase in MV generally cause worse results whereas OV can compensate this
fo'neural_network_plant_defined_for_simulation';
open_system mdl) % Open Simulink Model
sim mdl); % Start Simulation
% If constraints are not established well, it is better to in matlabsay nothing
mpc_for_NN.MV = [];
mdl = neural_network_plant_defined_for_simulation_ss;
open_system mdl) % Open Simulink(R) Model
sim mdl); % Start Simulation

```

CURRICULUM VITAE

Credentials

Name, Surname : Veysel Murat ÖNAL

Place of Birth : ISPARTA

Marital Status : Single

e-mail : vmurat_onal@hacettepe.edu.tr

Address : Hacettepe University, Üniversiteler Mahallesi, 06800 Çankaya/ Ankara

Education

High School : Ankara Atatürk Anatolian High School, Ankara

BSc. : Middle East Technical University, Department of Mechanical Engineering, Ankara

MSc. : Hacettepe University, Department of Mechanical Engineering, Ankara

Foreign Languages

English : Advanced



HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ

...~~MAKİNA MÜHENDİSLİĞİ~~... ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 21/09/2017

Tez Başlığı / Konusu: AKSESUAR KAYIŞ SİTEMLERİNDE ENİME (YANAL) TİTREŞİMİN
AZALTILMASI: KAYIŞ GERİSİNİN MODEL ÖNGÖRÜLÜ KONTROLÜ

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 134... sayfalık kısmına ilişkin, 21/09/2017 tarihinde şahsım/tez danışmanım tarafından Turan... adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %10...'tür.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç ✓
- 2- Alıntılar hariç/dâhil ✓
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç ✓

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

Adı Soyadı: Veysel Murat ÖNAL
Öğrenci No: N14126869
Anabilim Dalı: Makina Mühendisliği
Programı: Makina Mühendisliği
Statüsü: Y.Lisans Doktora Bütünleşik Dr.

21.09.2017

DANIŞMAN ONAYI

UYGUNDUR.

(Unvan, Ad Soyad, İmza)

5



HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
THESIS/DISSERTATION ORIGINALITY REPORT

HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
TO THE DEPARTMENT OF MECHANICAL ENGINEERING

Date: 21/09/2017

Thesis Title / Topic: REDUCTION OF TRANSVERSE VIBRATION OF A SERPENTINE
BELT SYSTEM; MODEL PREDICTIVE CONTROL OF A BELT TENSIONER

According to the originality report obtained by myself/my thesis advisor by using the Turnitin plagiarism detection software and by applying the filtering options stated below on 21/09/2017 for the total of 134 pages including the a) Title Page, b) Introduction, c) Main Chapters, and d) Conclusion sections of my thesis entitled as above, the similarity index of my thesis is 10 %.

Filtering options applied:

1. Bibliography/Works Cited excluded ✓
2. Quotes excluded / included ✓
3. Match size up to 5 words excluded ✓

I declare that I have carefully read Hacettepe University Graduate School of Science and Engineering Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

I respectfully submit this for approval.

Date and Signature

Name Surname: Veysel Murat ÖNAL
Student No: 11126869
Department: Mechanical Engineering
Program: Mechanical Engineering
Status: Masters Ph.D. Integrated Ph.D.

21.09.2017

[Signature]

ADVISOR APPROVAL

APPROVED.

[Signature]
Assist. Prof. Dr. Can Ulas Doyun
(Title, Name Surname, Signature)

[Handwritten mark]