COOPERATIVE ADAPTIVE CRUISE CONTROL ALGORITHMS FOR VEHICULAR PLATOONS BASED ON DISTRIBUTED MODEL PREDICTIVE CONTROL

by

Tuğba Taplı

B.S., Control and Automation Engineering, Istanbul Technical University, 2013

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2019

COOPERATIVE ADAPTIVE CRUISE CONTROL ALGORITHMS FOR VEHICULAR PLATOONS BASED ON DISTRIBUTED MODEL PREDICTIVE CONTROL

APPROVED BY:

Prof. Mehmet Akar .....
(Thesis Supervisor)

Prof. Yağmur Denizhan .....

Prof. Tankut Acarman .....

DATE OF APPROVAL: 28.05.2019

ABSTRACT

COOPERATIVE ADAPTIVE CRUISE CONTROL ALGORITHMS FOR VEHICULAR PLATOONS BASED ON DISTRIBUTED MODEL PREDICTIVE CONTROL

Over the past decade, autonomous driving and driver assistance systems have become popular research topics with the aim of reducing traffic congestion, driver labor and rate of accidents. This has changed the requirements and priorities of today's transportation perception and reveals a need for an increased level of complexity. Changes in autonomous driving algorithms as a result of this differentiation have led the platoon system applications turn into a difficult control problem when evaluated together with bidirectional communication topologies and delays. Furthermore, the increased number of vehicles added to the platoon requires higher computational power. Using distributed controllers in such applications where response time is important helps us get more reliable results.

The main objective of this thesis is to investigate Cooperative Adaptive Cruise Control Algorithms for Vehicular Platoons using Distributed Model Predictive Control (DMPC) under various communication links including unidirectional and bidirectional and also to propose a solution for a pre-known communication delay. Existing studies show that DMPC provides zero steady-state error with a fast response time under unidirectional communication topologies, however bidirectional links and communication delay effect are not adequately addressed. This thesis presents DMPC algorithms based on the information coming from the follower vehicles and delayed information received from the leader vehicle. It is shown that DMPC is the right tool to be utilized to both model and manage bidirectional communication. Simulation results demonstrate that the steady state error caused by the communication delay is successfully handled.

ÖZET

ARAÇ KONVOY SİSTEMLERİ iÇiN DAĞITILMIŞ MODEL ÖNGÖRÜLÜ KONTROLE DAYALI İŞBİRLİKÇİ UYARLAMALI SEYİR KONTROL ALGORİTMALARI

Geçtiğimiz on yılda, trafik sıkışıklığını, sürücü eforunu ve kaza oranını azaltmak amacıyla geliştirilen otonom sürüş ve sürücü destek sistemleri popüler araştırma konuları haline gelmiştir. Bugünün ulaşım ihtiyaçlarının farklılaşması sonucunda otonom sürüş algoritmalarında ortaya çıkan değişimler, çift yönlü iletişim topolojileri ve veri aktarımındaki gecikmelerle birlikte değerlendirildiğinde, konvoy sistemi uygulamalarının zor bir kontrol problemine dönüşmesine neden olmuştur. Ek olarak, konvoydaki araç sayısı arttıkça kontrol problemine yönelik çözümler için gereken hesaplama gücü de artmakta ve tepki süresinin önemli olduğu bu tip uygulamalarda dağıtılmış denetleyicilerin kullanılması, daha güvenilir sonuçlar elde edilmesine olanak tanımaktadır.

Bu tezin amacı, tek yönlü ve çift yönlü de dahil olmak üzere çeşitli iletişim bağlantıları altında Dağıtılmış Model Öngörülü Kontrol (DMPC) kullanarak araç konvoyları için bir İşbirlikçi Uyarlamalı Seyir Kontrol Algoritmaları önermek ve iletişim gecikmesinin neden olduğu sürekli hal hatası için bir çözüm üretmektir. Mevcut araştırmalar, DMPC'nin farklı tek yönlü iletişim topolojileri altında hızlı yanıt süresiyle sıfır sabit durum hatası sağlayabildiğini göstermektedir. Ancak çift yönlü bağlantıların ve iletişim gecikme etkisinin yeterince ele alınmadığı tespit edilmiştir. Bu tezde, takip araçlarından gelen bilgileri optimizasyon sorununa dahil etmek ve lider araçtan alınan gecikmeli bilgilerin sonuçlarını yönetmek için uygulanabilecek DMPC algoritmaları araştırılmıştır. DMPC'nin, iki yönlü iletişimi modellemek ve yönetmek için kullanılacak doğru yöntem olduğu gösterilmiştir. Simülasyon sonuçları, iletişim gecikmesinin sürekli hal hatası üzerindeki olumsuz etkilerinin başarıyla önüne geçilebildiğini göstermiştir.

TABLE OF CONTENTS

ABSTRACT iii
ÖZET iv
LIST OF FIGURES viii
LIST OF TABLES xii
LIST OF SYMBOLS xiii
LIST OF ACRONYMS/ABBREVIATIONS xv
1. INTRODUCTION 1
1.1. Related Work 2
1.2. Motivation of the Thesis 4
1.3. Organization of the Thesis 5
2. VEHICULAR PLATOON MODEL 7
2.1. Nonlinear Vehicle Longitudinal Dynamics 7
2.2. Communication Topology 9
2.2.1. Graph Theoretic Definitions 9
2.2.2. Communication Model of the Platoon 10
2.2.3. Predecessor Following (PF) Communication Topology 12
2.2.4. Predecessor Leader Following (PLF) Communication Topology . 13
2.2.5. Two Predecessor Following (TPF) Communication Topology . . 13
2.2.6. Two Predecessor Leader Following (TPLF) Communication Topology 14
2.3. Summary of the Chapter 15
3. DISTRIBUTED MODEL PREDICTIVE CONTROL FOR VEHICULAR PLATOON 16
3.1. DMPC Algorithm for Unidirectional Topologies 16
3.2. Optimization Problem Definition for Unidirectional Topologies 18
3.2.1. String Stability Notion 18
3.2.2. Cost 19
3.2.3. Constraints 20
3.2.3.1. Input Constraints 21

3.2.3.2. Output Constraints 21
3.2.4. Problem Solver 21
3.3. Test Scenarios and Simulation Results for Unidirectional Topologies . . 22
3.3.1. Test Scenario 1: Rapid Acceleration 23
3.3.2. Test Scenario 2: Rapid Deceleration 26
3.4. Summary of the Chapter 28
4. VEHICULAR PLATOON CONTROL WITH BIDIRECTIONAL COMMUNICATION TOPOLOGY 29
4.1. Bidirectional Communication Topology 29
4.2. DMPC Algorithm for Bidirectional Topologies 30
4.3. Optimization Problem Definition for Bidirectional Topologies 31
4.3.1. Cost 32
4.3.2. Constraints 32
4.3.2.1. Input Constraints 32
4.3.2.2. Output Constraints 32
4.3.3. Problem Solver 33
4.4. Test Scenarios and Simulation Results for Bidirectional Topologies . . . 34
4.4.1. Test Scenario 1: Rapid Acceleration 34
4.4.2. Test Scenario 2: Rapid Deceleration 36
4.5. Summary of the Chapter 39
5. VEHICULAR PLATOON CONTROL WITH COMMUNICATION DELAY 40
5.1. Communication Topology with Uniform Delay 40
5.2. DMPC Algorithm for Unidirectional Topologies with Uniform Communication Delay 43
5.3. Optimization Problem Definition for Bidirectional Topologies with Uniform Communication Delay 45
5.3.1. Cost 45
5.3.2. Constraints 45
5.3.2.1. Input Constraints 45
5.3.2.2. Output Constraints 45
5.3.3. Problem Solver 45

5.4. Test Scenarios and Simulation Results for Bidirectional Topologies with Uniform Communication Delay 46
5.4.1. Test Scenario 1: Rapid Acceleration 46
5.4.2. Test Scenario 2: Rapid Deceleration 47
5.5. Communication Topology with Information Packet Loss 50
5.6. Summary of the Chapter 57
6. CONCLUSION 58
REFERENCES 61

LIST OF FIGURES

Figure 2.1. Graph Theoretic Definitions (a) Undirected Graph (b) Directed Graph 9

Figure 2.2. Example of a Spanning Tree 10

Figure 2.3. Communication Topologies (a) Predecessor Following (b) Predecessor Leader Following (b) Two-Predecessor Following (d) Two-Predecessor Leader Following 11

Figure 3.1. Distributed MPC Algorithm 17

Figure 3.2. Rapid Acceleration under PF 23

Figure 3.3. Rapid Acceleration under PLF 24

Figure 3.4. Rapid Acceleration under TPF 24

Figure 3.5. Rapid Acceleration under TPLF 25

Figure 3.6. Rapid Deceleration under PF 26

Figure 3.7. Rapid Deceleration under PLF 26

Figure 3.8. Rapid Deceleration under TPF 27

Figure 3.9. Rapid Deceleration under TPLF 27

Figure 4.1. Bidirectional Communication Topology a) Predecessor Following (b) Predecessor Leader Following (b) Two-Predecessor Following (d) Two-Predecessor Leader Following 30

Figure 4.2. Distributed MPC Algorithm for Platoon with Bidirectional Topology 31

Figure 4.3. Rapid Acceleration under PF-bi 34

Figure 4.4. Rapid Acceleration under PLF-bi 34

Figure 4.5. Rapid Acceleration under TPF-bi 35

Figure 4.6. Rapid Acceleration under TPLF-bi 35

Figure 4.7. Rapid Deceleration under PF-bi 36

Figure 4.8. Rapid Deceleration under PLF-bi 37

Figure 4.9. Rapid Deceleration under TPF-bi 37

Figure 4.10. Rapid Deceleration under TPLF-bi 38

Figure 5.1. Communication Topology with Delay a) Predecessor Leader Following (b)Two-Predecessor Leader Following 41

Figure 5.2. Rapid Acceleration under PLF with Uniform Communication Delay 42

Figure 5.3. Rapid Acceleration under TPLF with Uniform Communication Delay 42

Figure 5.4. Desired Trajectory Updates in Matrices for Uniform Delay Handling 43

Figure 5.5. Distributed MPC Algorithm for Platoon with Communication Delay 44

Figure 5.6. Rapid Acceleration under PLF with Uniform Communication Delay 46

Figure 5.7. Rapid Acceleration under TPLF with Uniform Communication Delay 47

Figure 5.8. Rapid Deceleration under PLF with Uniform Communication Delay 47

Figure 5.9. Rapid Deceleration under TPLF with Uniform Communication Delay 48

Figure 5.10. Rapid Acceleration under PLF with Communication Delay with $N_p = 25$ 49

Figure 5.11. Rapid Acceleration under TPLF with Communication Delay with $N_p = 30$ 49

Figure 5.12. Topology Switch from PLF to PF at t=0.3s 51

Figure 5.13. Topology Switch from PLF to PF at t=0.4s 51

Figure 5.14. Topology Switch from PLF to PF at t=0.5s 52

Figure 5.15. Topology Switch from PLF to PF at t=1s 52

Figure 5.16. Topology Switch from TPLF to TPF at t=0.3s 53

Figure 5.17. Topology Switch from TPLF to TPF at t=0.4s 54

Figure 5.18. Topology Switch from TPLF to TPF at t=0.5s 54

Figure 5.19. Topology Switch from TPLF to TPF at t=1s 55

Figure 5.20. Continual Topology Switching between PLF and PF 56

Figure 5.21. Continual Topology Switching between TPLF to TPF 56

LIST OF TABLES

Table 3.1. Weights in the cost functions 20

Table 3.2. Parameters for the Platoon 23

Table 4.1. Weights in the cost functions 33

LIST OF SYMBOLS

A	Adjacency matrix
a_i	Acceleration of the ith vehicle
$C_{A,i}$	Aerodynamic drag coefficient of the ith vehicle
$d_{i,i-}$	Spacing between ego vehicle and its preceding vehicle
$d_{i,i+}$	Spacing between ego vehicle and its following vehicle
e_i	Position error of the ith vehicle
f_i	Rolling resistance of the ith vehicle
F_i	Cost function weight of J_y^a for the ith vehicle
$F_{veh,i}(v_i(t))$	Function of speed of the ith vehicle to balance wind resistance and road friction
g	Gravity constant
G_i	Cost function weight of J_{yi-}^a for the ith vehicle
J	Overall cost function
J_u	Cost function to penalize input error
J_y	Cost function to penalize output error
J_y^a	Cost function to maintain assumed input
J_{yi-}^a	Cost function for preceding vehicles to stay close
J_{yi+}^a	Cost function for follower vehicles to stay close
$m_{veh,i}$	Mass of the ith vehicle
N_c	Control horizon
N_i	Neighbor set
N_p	Prediction horizon
P_i	Cost function weight of J_{yi+}^a for the ith vehicle
Q_i	Cost function weight of J_y for the ith vehicle
R_i	Cost function weight of J_u for the ith vehicle
$R_{t,i}$	Tire radius of the ith vehicle
s_i	Position of the ith vehicle
${u_i}^a$	Assumed control input of the ego vehicle
${u_i}^p$	Predicted control input of the ego vehicle

${u_i}^*$	Optimal control sequence
$U_{max,i}$	Upper bound for the control input of the ith vehicle
$U_{min,i}$	Lower bound for the control input of the ith vehicle
v_i	Velocity of the ith vehicle
${v_i}^p$	Predicted velocity of the ego vehicle
x_i	Current states of the ego vehicle
$y_{des,i}$	Desired output of the ego vehicle
$y'_{des,i}$	Desired output of the ego vehicle under communication delay
${y_i}^a$	Assumed output of the ego vehicle
${y_{i-}}^a$	Assumed output of the preceding vehicle
${y_{i+}}^a$	Assumed output of the following vehicle
${y_i}^p$	Predicted output of the ego vehicle
$\eta_{T,i}$	Driveline efficiency of the ith vehicle

LIST OF ACRONYMS/ABBREVIATIONS

ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CDG	Constant Distance Gap
CTG	Constant Time Gap
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
PATH	Partners for Advanced Transit and Highways
PF	Predecessor Following
PLF	Predecessor Leader Following
TPF	Two Predecessor Following
TPLF	Two Predecessor Leader Following
TTC	Time-to-collision
V2V	Vehicle-to-Vehicle

1. INTRODUCTION

Autonomous driving exploded in the middle of the twentieth century with significant improvements in sensor and processor technology and since then it has been very generous on creating new investigation areas for researchers. It embodies a number of features like Green-Light-Optimized-Speed-Advisory (GLOSA), autonomous parking, lane changing, lane keeping assistance, traffic light assistance and platooning. A group of vehicles intersecting in a communication range that send and receive information with each other directly or indirectly and conduct motion planning actions is called the vehicle platoon. Vehicle platooning gained considerable attention in recent years as it promises to increase the highway capacity with the ability to lower spacing between vehicles, stabilize the traffic flow, and reduce the fuel consumption and carbon emissions consequently [1].

The initial boost of the platoon control has been administered by the PATH (Partners for Advanced Transit and Highways) program in the 1980s. PATH investigated the control topics such as collision avoidance, and string stability [2], [3] and has been the pioneer to many topics such as spacing policy, communication topology and dynamic heterogeneity in the succeeding years. Moreover, state-of-the-art platoon control algorithms such as multi-agent consensus control and sliding-mode control have also found application area on platooning in the upcoming years [4], [5], [6].

SARTRE (Safe Road Trains for the Environment), is another project funded by European Commission to investigate and implement the latest technologies for the definition of new transportation concept. It is mainly driven by safety motivation and eventually, it aims to eliminate the traffic accidents caused by human driver [7]. National Highway Traffic Safety Administration (NHTSA) reports that human errors resulting from recognition latency and making wrong decisions constitute %94 of all accidents [8]. Autonomous driving can better manage pre-crash events with faster response time and reduce the number of fatal accidents, consequently.

CACC (Cooperative Adaptive Cruise Control) can be considered as a transitional step in the direction of automated vehicle platooning. There are a couple of deviations in CACC from the platoon control. In CACC, the longitudinal behavior of the vehicle is automated and speed control is conducted depending on the information received from vehicle to vehicle (V2V) communication or on-board radar-based sensors. Lane changing/keeping, vehicle steering tasks still need to be handled by the driver. Another difference is on spacing policy, generally, in platooning systems Constant Distance Gap (CDG) is preferred whereas in CACC, Constant-Time Gap (CTG). CTG can be classified as a dynamic spacing policy since it changes with the vehicle speed [9].

1.1. Related Work

Model predictive control (MPC) has been accepted in the industry as being able to provide systematic solutions and has been gaining popularity since it can respond to a large span of control problems due to its ability to optimize closed-loop performance by considering the constraints on desired inputs, states, and outputs [10]. MPC controller is also widely chosen by the researchers while generating autonomous driving algorithms.

Different from CACC, in ACC the vehicles do not have the communication links and they are assumed to be equipped with on-board sensors to measure the distance and the velocity of the preceding vehicle. In [11], a multi-objective ACC (Adaptive Cruise Control) algorithm is proposed with the motivation of velocity tracking, collision avoidance, driver comfort and fuel economy. MO-MPC (Multi-Objective MPC) controller was implemented by using a linear vehicle model to control and calculate the optimum acceleration of the vehicle with constraint of spacing to the preceding vehicle, moreover, time-to-collision (TTC) strategy is adopted as a safety criteria to compare the MO-MPC response against Linear Quadratic Regulator (LQR) performance. MO-MPC has shown better results over LQ-ACC (LQR based ACC) under rapid deceleration scenario by preserving the inter-vehicle safety distance. Although this study shows that MPC can be nominated to generate satisfying ACC algorithms, it should be noted that inter-vehicle states are represented with linear equations and

vehicle dynamics are not considered within the scope of this study.

Traffic flow regulation and increased driver comfort can be accounted as the principal offerings of ACC. CACC implementation has the advantage of communication links for achieving a shorter gap with higher safety over ACC. Despite the fact that shorter spaces between vehicles can decrease traffic load, it may also cause string instability. The notion of string stability and requirements to achieve it are described in [12]. This study is quite powerful in the sense of demonstrating that string stability cannot be achieved if vehicles use only relative distance information under CDG policy. In [13], Swaroop et al. compared CDG and CTG policies and highlighted that in order to increase the road capacity via the small spacing between vehicles, CDG policy shall be followed to guarantee string stability even though it necessitates inter-vehicle communication due to the fact that CTG leads to input saturation for small headways.

It is also highlighted that decentralized approaches are preferable over centralized ones considering the increased computational burden for medium to large platoons. Another remarkable study on distributed control algorithms is [5]. This paper provides a distributed model predictive control approach for vehicular platoon systems under four different communication topologies. As being a distributed algorithm, each vehicle is expected to solve its own optimal control problem relying on the information received from its neighbor vehicles defined by the communication topology. Vehicle longitudinal dynamics are captured in the nonlinear form and throttle/brake torque is calculated in order to control vehicles. It has been highlighted that the leader vehicle desired set point information is not shared with the vehicles other than the ones that are linked to the leader. It provides robust vehicle following and string stability performances for both ACC and CACC vehicle strings. The given optimization problem is able to minimize the platoon size to reduce the traffic and achieve the string stability at the same time.

1.2. Motivation of the Thesis

There exist numerous studies in the literature aiming to develop more effective and stable algorithms constructed with different controllers applied on linear or nonlinear platoon models and with different communication topologies. In [14], it has been identified that the string stability could not be promised with CDG policy under bidirectional predecessor following topology using linear controllers. In view of this fact, this thesis is structured to propose an algorithm for bidirectional topologies to satisfy string stability with CDG policy.

In addition, bidirectional links are implemented only on predecessor following and predecessor-leader following communication topologies, but results have not been compared with two-predecessor following and two-predecessor-leader following communication topologies, yet. This thesis also aims to implement the bidirectional links to more communication topologies as they are considered to be a strong tool for collision avoidance under cases such as rapid deceleration. While emergency braking, the follower vehicle information can be as crucial as the predecessor vehicle information and controllers with the bidirectional links might be able to reduce the collision risks.

Communication delay effects have not been addressed for two-predecessor-leader following communication topology in the literature. Considering this deficiency, it is aimed to investigate the leader information delayed platoon systems under predecessor-leader following and two-predecessor-leader following communication topologies for comparison in this thesis. Furthermore, an alternative approach to handle the delay is proposed. Wireless communication dropout is also analyzed and communication topology change is simulated to examine the effects.

Although some studies separately focused on the bidirectional topology [15], [16] and communication delay [17], [18], [19], these topics have not been considered together in a comprehensive work that discusses the application of DMPC under different communication topologies with a nonlinear platoon model. Besides the discussed items above, the main targets of this study can be listed as follows:

- To construct a simulation environment for a heterogeneous platoon in Matlab and to apply a nonlinear DMPC algorithm under four different communication topologies
- To include bidirectional communication links in order to have the states of the nearest following vehicle to the ego vehicle and to construct a DMPC algorithm to manage the information coming from the following vehicle and keep the platoon stable under bidirectional four topologies
- To apply communication delay to data transmitted by the leader vehicle for topologies that contain leader information and to make the necessary modifications in the algorithm for avoiding dissatisfying controller performance.

1.3. Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, nonlinear equations of the vehicle longitudinal dynamics are given. They are used in the model predictive controller and also to create the plant model for the simulation. Graph theory definitions that are going to be adopted in this thesis are briefly explained and the communication model of the platoon is presented based on these definitions.

In Chapter 3, Distributed Model Predictive Control algorithm development and the optimization problem including cost and constraint selections are explained step by step along with the details of the solver. Test results are generated for all 4 communication topologies; Predecessor Following, Predecessor-Leader Following, Two-Predecessor Following and Two-Predecessor-Leader Following.

In Chapter 4, the bidirectional topology is investigated using the communication topologies defined in Chapter 3 as a basis. Necessary changes on the Distributed Model Predictive Control Algorithm to keep the platoon string stable and steady-state errors minimum are presented. The proposed algorithm is tested for rapid acceleration and deceleration scenarios.

In Chapter 5, the communication delay is added to the platooning system. The receiver node used in the DMPC algorithm is treated to deal with the communication delay effect. The performance of the DMPC controller is examined under rapid acceleration and deceleration scenarios with implemented delay to leader information in PLF and TPLF topologies. In addition, the interruption in wireless communication and the resulting topology change are investigated.

In Chapter 6, the thesis concludes with a brief summary and future proposals.

2. VEHICULAR PLATOON MODEL

There are number of studies available in platoon control where researchers use linear [11] or nonlinear models [5] to capture the longitudinal vehicle dynamics. In [11], Rajamani proposed an Adaptive Cruise Control algorithm using linear Model Predictive Controller with dynamic spacing policy where the desired gap is a function of vehicle velocity. The linear MPC was able to provide satisfying results in reference tracking and protecting the desired space compared to LQR based ACC algorithm.

Nonlinear models can be preferred to better reflect the characteristics of the drivetrain and therefore the platoon. In [5], Zheng et al. proposed a nonlinear MPC algorithm to track the reference velocity profile with constant spacing policy under different communication topologies. The authors revealed impressive results on sustaining the string stability and reference tracking with distributed MPC design.

In this thesis, longitudinal vehicle dynamics are captured in nonlinear equations with simplifications made on the drivetrain in order to have a relatively simple prediction model for MPC.

2.1. Nonlinear Vehicle Longitudinal Dynamics

Nonlinear longitudinal vehicle dynamics are adopted as represented in [5]. The plant model for the closed loop control system of the ith vehicle is described by

$$s_i(t+1) = s_i(t) + v_i(t)\Delta t \tag{2.1}$$

$$v_i(t+1) = v_i(t) + \frac{\Delta t}{m_{veh,i}}\left(\frac{\eta_{T,i}}{R_{t,i}}T_i(i) - F_{veh,i}(v_i(t))\right) \tag{2.2}$$

$$T_i(t+1) = T_i(t) - \frac{1}{\tau_i}T_i(t)\Delta t + \frac{1}{\tau_i}u_i(t)\Delta t \tag{2.3}$$

$$F_{veh,i}(v_i(t)) = C_{A,i}v_i^2(t) + m_{veh,i}gf_i \tag{2.4}$$

where $s_i(t)$, $v_i(t)$, $T_i(t)$ represent position, velocity and desired driving/braking torque values of the ith vehicle, at time t, respectively; $F_{veh,i}(v_i(t))$ is a function of speed of the ith vehicle to balance wind resistance and road friction, $m_{veh,i}$ denotes vehicle mass, $\eta_{T,i}$ is the driveline efficiency, $R_{t,i}$ is tire radius, $C_{A,i}$ is the aerodynamic drag coefficient, g is the gravity constant and f_i is the rolling resistance coefficient. Position, velocity and desired driving/braking torque are states of the MPC while desired driving/braking torque is also used as the control signal of the system [5].

It is possible to acquire the resulting platoon dynamics via (2.1)-(2.4). For each individual vehicle, the state is defined as $x_i(t) = [s_i(t), v_i(t), T_i(t)]^T$ where the output is $y_i(t) = [s_i(t), v_i(t)]^T$. Combining these will introduce

$$x_i(t+1) = \phi_i(x_i(t)) + \psi_i(t).u_i(t) \tag{2.5}$$

$$y_i(t+1) = \gamma x_i(t+1) \tag{2.6}$$

where $\psi_i = [0, 0, (1/\tau_i)\Delta t]^T \in R^{3\times 1}$; $\gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \in R^{2\times 3}$ and

$$\phi_i = \begin{bmatrix} s_i(t) + v_i(t)\Delta t \\ v_i(t) + \frac{\Delta t}{m_{veh,i}}\left(\frac{\eta_{T,i}}{R_{t,i}} T_i(i) - F_{veh,i}(v_i(t))\right) \\ T_i(t) - \frac{1}{\tau_i} T_i(t)\Delta t \end{bmatrix} \in R^{3\times 1}.$$

Merging states, inputs and outputs of all nodes as vectors $X(t) = [x_1(t)^T, x_2(t)^T, ..., x_N(t)^T]^T \in R^{3N\times 1}$, $U(t) = [u_1(t)^T, u_2(t)^T, ..., u_N(t)^T]^T \in R^{N\times 1}$ and $Y(t) = [y_1(t)^T, y_2(t)^T, ..., y_N(t)^T]^T \in R^{2N\times 1}$ will give the overall discrete-time platoon dynamics

$$X(t+1) = \Phi X(t) + \Psi U(t) \tag{2.7}$$

$$Y(t+1) = \Gamma X(t+1) \tag{2.8}$$

where $\Phi = [\phi_1(x_1(t))^T, \phi_2(x_2(t))^T, ..., \phi_N(x_N(t))^T]^T \in R^{3N\times 1}$, $\Psi = diag\{\psi_1, ..., \psi_N\} \in R^{3N\times N}$ $\Gamma = I_N \otimes \gamma \in R^{2N\times 3N}$, where $\otimes$ represents the Kronecker product.

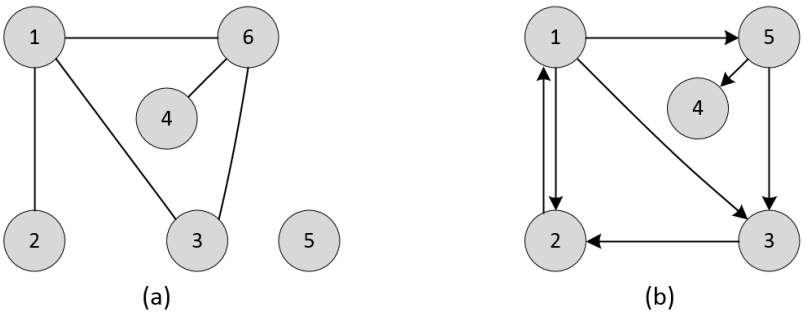

Figure 2.1. Graph Theoretic Definitions (a) Undirected Graph (b) Directed Graph

2.2. Communication Topology

The information flow topology defines which data to be received by each distributed controller and makes the impact on the overall platoon. As a consequence of that, the stability of the platoon is significantly dependent on the communication topology along with the distributed controllers. Here, directed graphs are used to represent the model for information flow that connects the vehicles in a platoon.

2.2.1. Graph Theoretic Definitions

A graph G consists of nodes and edges that may be connected with a pair of nodes and can be represented as $G = \{V, E\}$ where V is the set of nodes and E is the set of edges. An *undirected graph* is a set of nodes and a set of undirected edges that each connects a sequence of nodes as shown in Figure 2.1(a). Similarly, a *directed graph* (or digraph) contains the directed edges connecting a sequence of nodes as in Figure 2.1(b). A *directed edge* points from the first node to the second node in the pair. A graph is considered to be connected if there is a path from every node to all other nodes.

A *path* is defined as the sequence of edges (i_1, i_2),(i_2, i_3)...,(i_{k-1}, i_k) which connects a sequence of nodes in a graph. A graph is a *tree* if there are no cycles and is a path from any node to every other node which defines a connected graph. A *spanning*

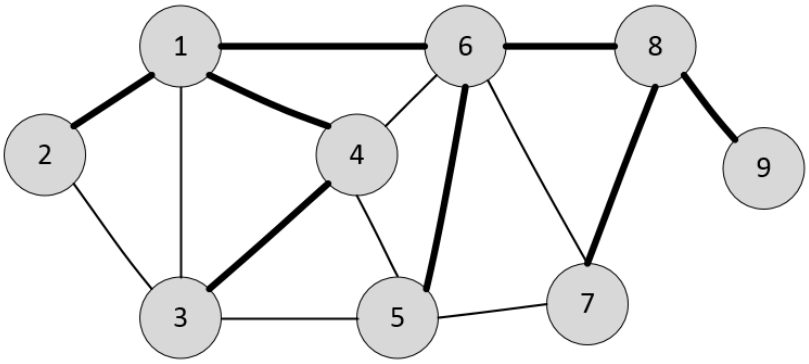

Figure 2.2. Example of a Spanning Tree

tree exists if a graph G is connected with the minimum possible number of edges as represented in Figure 2.2 with bold edges [20], [21].

In graph G, two nodes i, j of G are *neighbors*, if (i, j) is an edge of G. The neighbor set of node can be defined with $N_i = \{j|(i, j) \in E, j \in V\}$. The set N_i means that node i can receive information of any $j \in N_i$.

2.2.2. Communication Model of the Platoon

Considering the above definitions, communication between the nodes is defined with a directed graph formulated by an adjacency matrix. This matrix contains the information interchange of the leader and the following vehicles and can be expressed as $A = [a_{ij}] \in R^{N \times N}$ such that each element determined is via the rule

$$a_{ij} = \begin{cases} 1, \text{ if } (j, i) \in E \\ 0, \text{ if } (j, i) \notin E \end{cases} \tag{2.9}$$

where $i, j \in V, j \neq 0$ and $\{j, i\} \in E$. This notation denotes that there is a directional edge from node j to node i, i.e., node j sends its information to node i (or simply

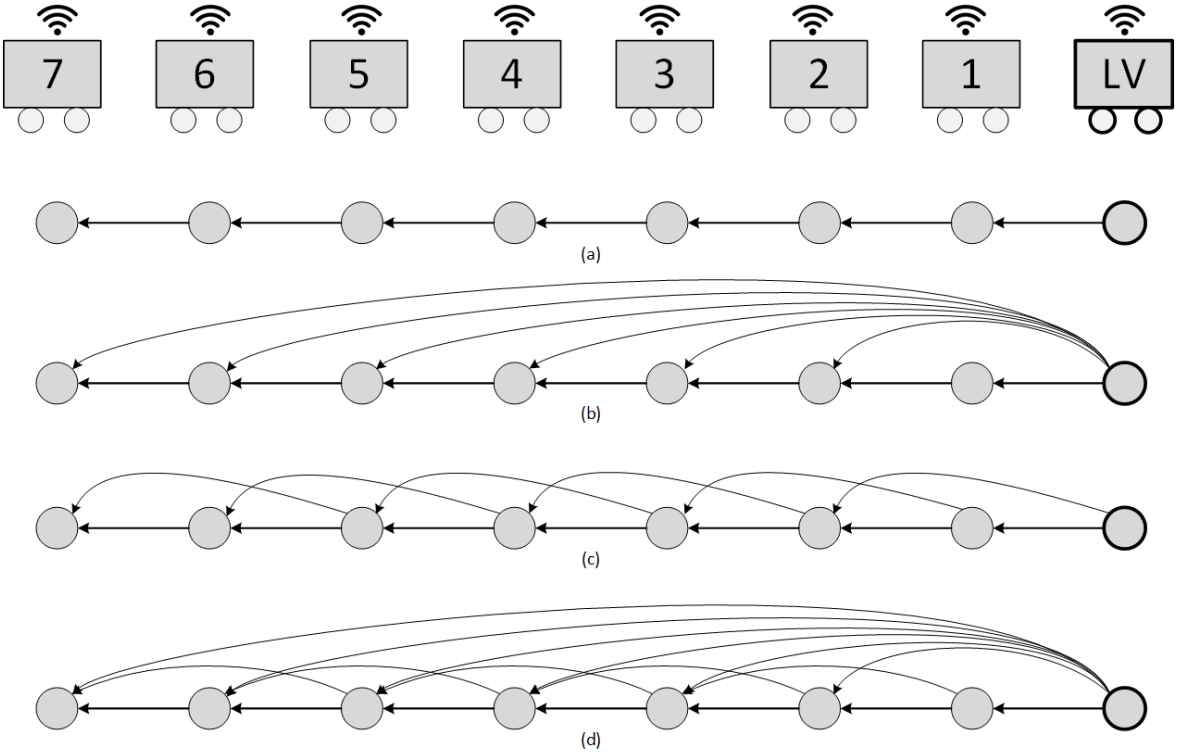

Figure 2.3. Communication Topologies (a) Predecessor Following (b) Predecessor Leader Following (b) Two-Predecessor Following (d) Two-Predecessor Leader Following

$j \rightarrow i$). Similarly leader information is stored with $\vec{l} = [l_i] \in R^{N\times 1}$ expressed as

$$l_i = \begin{cases} 1, \text{ if } (j,i) \in E \\ 0, \text{ if } (j,i) \notin E \end{cases} \tag{2.10}$$

where $i, j \in V$ and $j = 0$. The overall communication matrix can be represented in the compact form as

$$\overline{A} = \begin{bmatrix} \vec{l} & A \end{bmatrix} \in R^{N\times N+1}. \tag{2.11}$$

On a certain path where boundaries are identified, there will be a number of vehicles entering a communication range. Figure 2.3 shows the directed graphs that describe the information flow between vehicles in a platoon either via radar-based

sensors or V2V communications in that range. Four communication topologies are presented, Predecessor Following, Predecessor Leader Following, Two-Predecessor Following, Two-Predecessor Leader Following, to describe the connectivity of the vehicles.

2.2.3. Predecessor Following (PF) Communication Topology

The information flow for the predecessor following (PF) communication topology is given in Figure 2.3(a). This topology indirectly provides leader vehicle pre-defined desired trajectory information to the whole platoon by transferring through each vehicle. The first vehicle in the platoon obtains the desired set point information and shares it with its following neighboring vehicle and the vehicle that receives the shared information from the first vehicle transmits its own states to its follower. The following neighbor also receives the transmitted information by its preceding vehicle and shares its own states with the next vehicle in the platoon and this flow continues until the last vehicle. The communication matrix for the PF topology is given by

$$A_{PF} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.12}$$

The first column of the communication matrix introduces the information whether the following car has the leader trajectory information or not. Each row in A_{PF} represents the node information indicating who the corresponding follower information is from. As can be seen from (2.12), only the first vehicle in the platoon has the leader trajectory information.

2.2.4. Predecessor Leader Following (PLF) Communication Topology

The information flow for the predecessor leader following (PLF) communication topology is shown in Figure 2.3(b). This topology directly provides leader vehicle predefined desired trajectory information to the whole platoon. Each vehicle in the platoon uses both leader states and their predecessor vehicle states in solving the distributed optimization problem except for the first vehicle in the platoon. It only considers leader vehicle states to calculate its optimum torque value. The communication matrix for the PLF topology is given by

$$A_{PLF} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.13}$$

The first column of A_{PLF} matrix shows that all the vehicles in the platoon have the leader vehicle trajectory information. Except for the first vehicle in the platoon, each follower receives two predecessor vehicle trajectory information including the leader.

2.2.5. Two Predecessor Following (TPF) Communication Topology

The information flow for the two-predecessor following (TPF) communication topology is represented in Figure 2.3(c). Similar to the PF topology, TPF topology indirectly provides leader vehicle predefined trajectory information to the platoon. Differently, in TPF topology two of the closest predecessor vehicle states are obtained

by the ego vehicles. The communication matrix for the TPF topology is given by

$$A_{TPF} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{2.14}$$

As can be seen from (2.14), only the first and second vehicles in the platoon have the leader trajectory information. Except for the first vehicle in the platoon, each follower receives two of its predecessor vehicle's trajectory information.

2.2.6. Two Predecessor Leader Following (TPLF) Communication Topology

The information flow for the two-predecessor leader following (TPLF) communication topology is given in Figure 2.3(d). Among all, this topology is the one that provides the most information about the vehicle states in the platoon. Each vehicle directly has access to the leader vehicle states along with two of the closest predecessor vehicles. The communication matrix for the TPLF topology is given by

$$A_{TPLF} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{2.15}$$

The first column of A_{TPLF} matrix shows that all the vehicles in the platoon have the leader vehicle trajectory information. Except for the first and second vehicles in

the platoon, each follower receives two predecessor vehicle trajectory information in addition to the leader.

2.3. Summary of the Chapter

Nonlinear vehicle equations to be embedded in the prediction and plant model are presented. As done in the previous studies, simplifying assumptions will be applied in the vehicle model to keep the main focus on the solution of the information flow related topics. Communication topologies to be used in this study and their matrix representations are shown.

The next chapter demonstrates the DMPC algorithm for unidirectional PF, PLF, TPF and TPLF communication topologies by using the given platoon model explained in Section 2.1.

3. DISTRIBUTED MODEL PREDICTIVE CONTROL FOR VEHICULAR PLATOON

This chapter is devoted to explaining distributed platooning algorithm developed by Zheng et al. [5] and re-implementation of it by adhering to the main algorithm, the numerical values used for the platoon parameters and the cost function weights. This will enable to fit the investigations on the bidirectional communication topology and communication delay into the same frame for comparison purposes.

3.1. DMPC Algorithm for Unidirectional Topologies

The objective of platooning is the stabilization of each vehicle's dynamics in the platoon and the minimization of spacing errors in between while keeping the control effort low with the constraint of collision avoidance.

The powerful feature of the DMPC algorithm is its ability to provide a solution independent from the platoon size by assigning each node a local optimization problem. This feature especially gains importance in medium to large sized platoons where the computational complexity increases with the number of vehicles added to the platoon.

The algorithm followed to iterate the DMPC through the simulation time is given in Figure 3.1. In Step 1, all the following vehicles are assumed to be moving at a constant speed of e.g., 20m/s with no spacing error. The initial values for the assumed states are calculated based on the predefined initial position and velocity inputs, while the torque value as a function of vehicle speed and vehicle parameters that are given in Section 3.3. In Step 2, the optimization problem varies depending on the ego vehicle whether it has access to the trajectory of the leader vehicle or not. The same cost function and weights are used for all nodes and only the cost function related to receiving the leader trajectory is weighted zero if the node does not have that connection. The cost function components and its weights are explained in Section 3.2

along with the problem solver in a detailed way. Step 3 calculates the optimal states from the first value of the optimized control signal. Step 4 is to obtain the assumed control input signal for being able to compute assumed outputs through the prediction horizon in Step 5. Finally, in Step 6 the assumed outputs of the ego vehicle are sent and the neighbor vehicle states to be used in the next time step are received and the iteration starts over from Step 2.

Step 1: Assume that all vehicles are moving at constant speed without position error and initialize assumed states of each node based on

$$u_i^a(k|0) = h_i(v_i(0)),$$
$$y_i^a(k|0) = y_i^p(k|0),$$

for $k = 0, 1, \ldots, N_p - 1$

Step 2: Calculate the control signal as follows:

if leader information available

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$, $y_{i-}^a(k|t)$ and $y_{des,i}(k|t)$ by using the average values of neighbor and leader outputs as constraints for the states of node i

else

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$ and $y_{i-}^a(k|t)$ by using the average values of neighbor outputs as constraints

end if

Step 3: Compute the optimal states for the node using the first element of the optimized control signal

Step 4: Calculate the assumed control input with additional last value to ensure that the node moves at a constant speed

$$u_i^a(k|t+1) = \begin{cases} u_i^*(k+1|t), & k = 0, 1, ..N_p - 2 \\ h_i(v_i{}^*(N_p|t)), & k = N_p - 1 \end{cases}$$

Step 5: Calculate assumed outputs $y_i^a(k|t+1)$ using the nonlinear vehicle model given in 2.6

Step 6: Transmit $y_i^a(k|t+1)$ and receive $y_{i-}^a(k|t+1)$ for the next time step

Step 7: Increase t and go to Step 2.

Figure 3.1. Distributed MPC Algorithm

3.2. Optimization Problem Definition for Unidirectional Topologies

3.2.1. String Stability Notion

For a given safe distance δ, let

$$e_1(t) = s_{LV}(t) - s_1(t) - \delta \tag{3.1}$$

$$e_i(t) = s_{i-1}(t) - s_i(t) - \delta, \quad i = 2, ..., N \tag{3.2}$$

where $s_{LV}(t)$ and $s_i(t)$, $i = 2, ..., N$, denote the position of the leader vehicle and the ith follower, respectively. The main objective of platooning is to maintain a pre-defined constant spacing between each vehicle and its successor under different speed profiles of the leader. In other words, the goal is to achieve $e_i(t) \to 0$ as $t \to \infty$. Even though the prominent motivation is to obtain $e_i(t) \to 0$, secondary purpose is to ensure that the error towards the tail of the platoon does not escalate as the number of vehicles increases. Error, e_i, may amplitude when

$$|e_i| \leq c.|e_{i-1}|, \quad c > 1 \tag{3.3}$$

as $N \to \infty$ where c represents the ratio between two successive errors. Hence, it is required for all errors to be bounded as N increases. This notation is derived from the string stability definition in the literature [22]. As a more general expression,

$$\frac{\|e_i\|_\infty}{||e_{i-1}||_\infty} \leq 1 \tag{3.4}$$

should be secured to be able to ensure string stability. This implies that the $N - th$ error of the follower does not exceed that of the $(N-1)st$ follower, therefore

$$||e_N|| \leq ||e_{N-1}|| \leq ... \leq ||e_1|| \tag{3.5}$$

should be satisfied.

3.2.2. Cost

The cost function is basically constructed based on 4 components with their own weighting factors designed considering both normalization and prioritization purposes given in Table 3.1:

$$J(z) = J_y(z) + J_u(z) + J_y^a(z) + J_{y_{i-}}^a(z). \tag{3.6}$$

Here, $J(z)$ is the overall cost function, $J_y(z)$ and $J_u(z)$ are to penalize the output error and the input error, respectively; $J_y^a(z)$ is to maintain its assumed output, $J_{y_{i-}}^a(z)$ is to keep node's states close to its neighboring vehicle states. The first component of the overall cost function

$$J_y(z) = \big(y_i^p(k|t) - y_{des,i}(k|t)\big)^T Q_i \big(y_i^p(k|t) - y_{des,i}(k|t)\big) \tag{3.7}$$

targets to minimize the error between the current states of the node with the trajectory of the leader, if the leader information is available for that specific node. If it is not available, the weighting factor Q is set to 0, and thus the leader information will not be included in the optimization problem. The second component of the cost function

$$J_u(z) = \big(u_i^p(k|t) - h_i(v_i^p(k|t)\big)^T R_i \big((u_i^p(k|t) - h_i(v_i^p(k|t)\big) \tag{3.8}$$

aims to minimize the input signal that deviates from the equilibrium to achieve constant speed. The third component of the cost function

$$J_y^a(z) = \big(y_i^p(k|t) - y_i^a(k|t)\big)^T F_i \big(y_i^p(k|t) - y_i^a(k|t)\big) \tag{3.9}$$

tries to minimize the difference between its assumed trajectory and predicted trajectory. The assumed output trajectory is calculated by shifting the last-step optimal trajectory

Table 3.1. Weights in the cost functions [5]

Weights	**PF**	**PLF**	**TPF**	**TPLF**
F_i	F_i=$10I_2$, $i \in N$	F_i=$10I_2$, $i \in N$	F_i=$10I_2$, $i \in N$	F_i=$10I_2$, $i \in N$
G_i	G_1=0, G_i=$5I_2$, $i \in N\backslash\{1\}$	G_1=0, G_i=$5I_2$, $i \in N\backslash\{1\}$	G_1=0, G_i=$5I_2$, $i \in N\backslash\{1\}$	G_1=0, G_i=$5I_2$, $i \in N\backslash\{1\}$
Q_i	Q_1=$10I_2$, Q_i=0, $i \in N\backslash\{1\}$	Q_i=$10I_2$, $i \in N$	Q_1=$10I_2$, Q_2=$10I_2$, Q_i=0, $i \in N\backslash\{1,2\}$	Q_i=$10I_2$, $i \in N$
R_i	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$

of the same node. The last component of the cost function

$$J^a_{y_{i-}}(z) = \sum_{j \in N_i} \left(y^p_i(k|t) - y^a_{i-}(k|t) - \tilde{d}_{i,i-}\right)^T G_i \left(y^p_i(k|t) - y^a_{i-}(k|t) - \tilde{d}_{i,i-}\right) \tag{3.10}$$

is added in order to stay close to the assumed trajectories of the neighboring vehicles. $\tilde{d}_{i,i-}$ term is included to define the desired distance vector between node i and previous nodes. By setting $\tilde{d}_{i,i-} = [d_{i,i-}, 0]^T$ and $d_{i,i-} = 20m$, constant spacing policy is preferred to protect the distance between nodes. Cost function weights that are used in the simulations are given in Table 3.1.

3.2.3. Constraints

It is possible to add both linear and nonlinear equality or non-equality constraints to the optimization problem. In this application, only nonlinear equality constraints are being considered.

3.2.3.1. Input Constraints. Upper and lower limits for the control signal are calculated by using the below formula for each node based on related vehicle parameters:

$$a_i = \frac{\frac{\eta T_i}{R_i} - C_{A,i} v_0^2 - m_{veh,i} g f}{m_{veh,i}}. \tag{3.11}$$

The lower and upper limits for the braking and driving torque are set to the minimum and maximum acceleration values equal to -6 m/s^2 and 6 m/s^2, respectively.

There is no constraint added for the input rate as it does not make a significant difference in the solution, however, it might be taken into consideration for different test scenarios as follows:

$$U_{min,i} = T_{lowerbound,i}, \tag{3.12}$$

$$U_{max,i} = T_{upperbound,i}. \tag{3.13}$$

3.2.3.2. Output Constraints. Output constraints are calculated dynamically during the simulation based on the received information from the neighbor vehicles. That calculation also depends on the communication topology and changes with whether the leader information is available or not and the number of neighboring vehicles.

It focuses on minimizing the difference between the node's states with the assumed output of the neighbor vehicle/s at the end of the prediction horizon described by

$$y_i^p(N_p|t) = \frac{1}{|N_i|} \sum y_{i-}^a(N_p|t) + d_{i,i-}. \tag{3.14}$$

3.2.4. Problem Solver

To solve the constrained nonlinear programming problems, *fmincon* function provided by the Optimization Tool Box in MATLAB is used. *fmincon* solves the

problems having the following form:

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & Ax \leq b \\ & A_{eq}x = b_{eq} \\ & x_{lb} \leq x \leq x_{ub} \\ & c(x) \leq 0 \\ & c_{eq}(x) = 0 \end{aligned}$$

3.3. Test Scenarios and Simulation Results for Unidirectional Topologies

The DMPC algorithm is tested against the rapid acceleration and rapid deceleration scenarios assuming the algorithm would easily achieve good results with a smoother increase/decrease in the acceleration if it is already providing good results with the rapid change. Another reason to use these scenarios is already having an example study to be taken as reference for the performance comparison.

As already mentioned, a heterogeneous platoon is aimed to better reflect the real world use case since not all the vehicles in a platoon could have identical drivetrain characteristics. Parameters used to create different vehicle dynamics are given in Table 3.2. Below fmincon settings are applied for this simulation:

- Algorithm: Interior-point algorithm
- Optimality Tolerance: 10^{-5}
- Function Tolerance: 10^{-5}
- Step Tolerance: 10^{-6}

The solver settings are optimized after a series of trials and kept as the same for all of the investigations throughout this study. In addition to these, prediction horizon (N_p) and control horizon (N_c) are chosen as 20 and 1, respectively.

Table 3.2. Parameters for the Platoon [5]

Vehicle Index	$m_{veh,i}$ **(kg)**	τ_i	$C_{A,i}(N.s^2.m^{-2})$	R_i
1	1035.7	0.51	0.99	0.30
2	1849.1	0.75	1.15	0.38
3	1934.0	0.78	1.17	0.39
4	1678.7	0.70	1.12	0.37
5	1757.7	0.73	1.13	0.38
6	1743.1	0.72	1.13	0.37
7	1392.2	0.62	1.06	0.34

3.3.1. Test Scenario 1: Rapid Acceleration

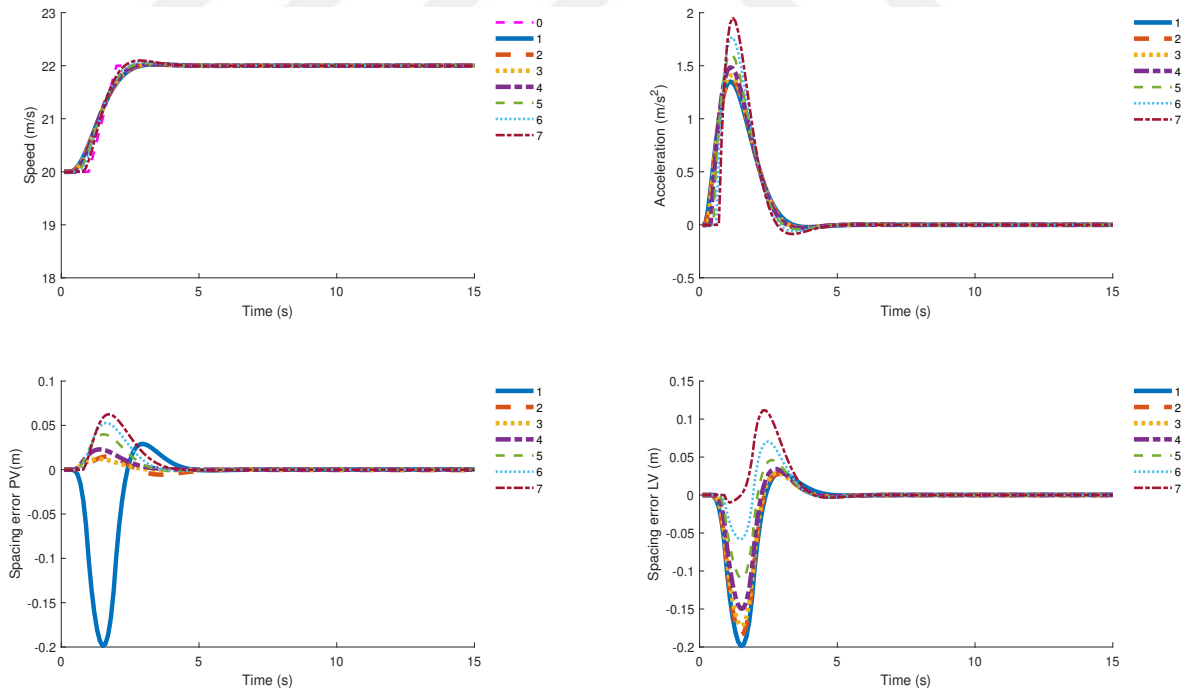

Figure 3.2. Rapid Acceleration under PF

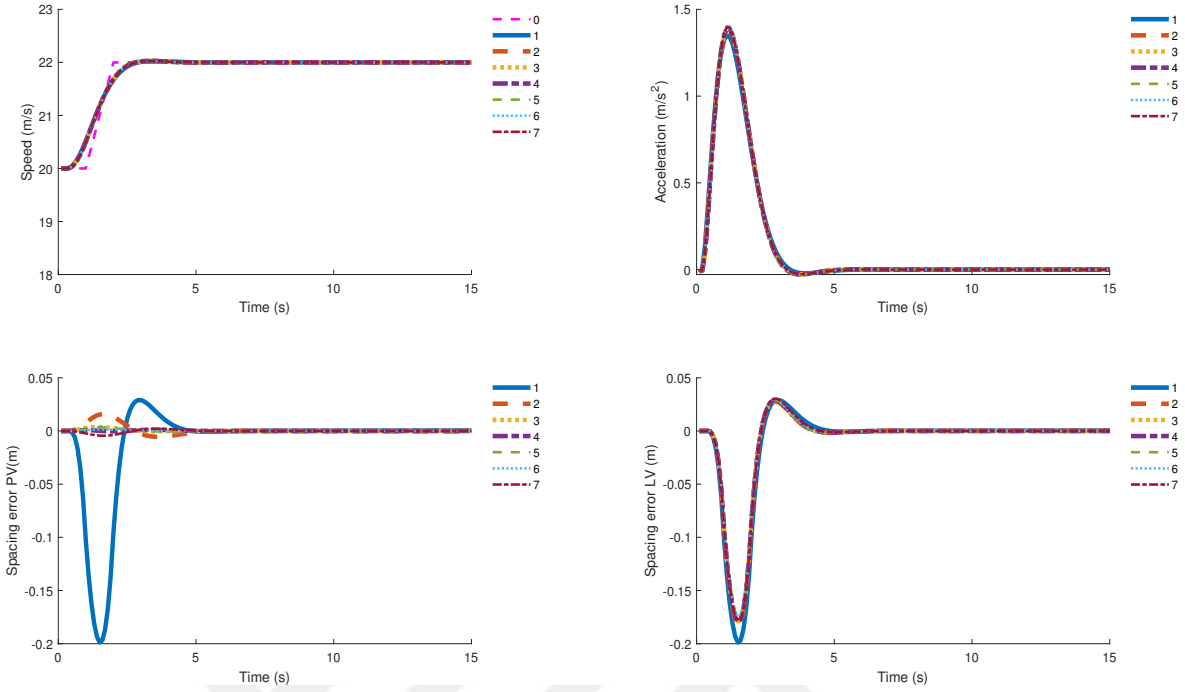

Figure 3.3. Rapid Acceleration under PLF

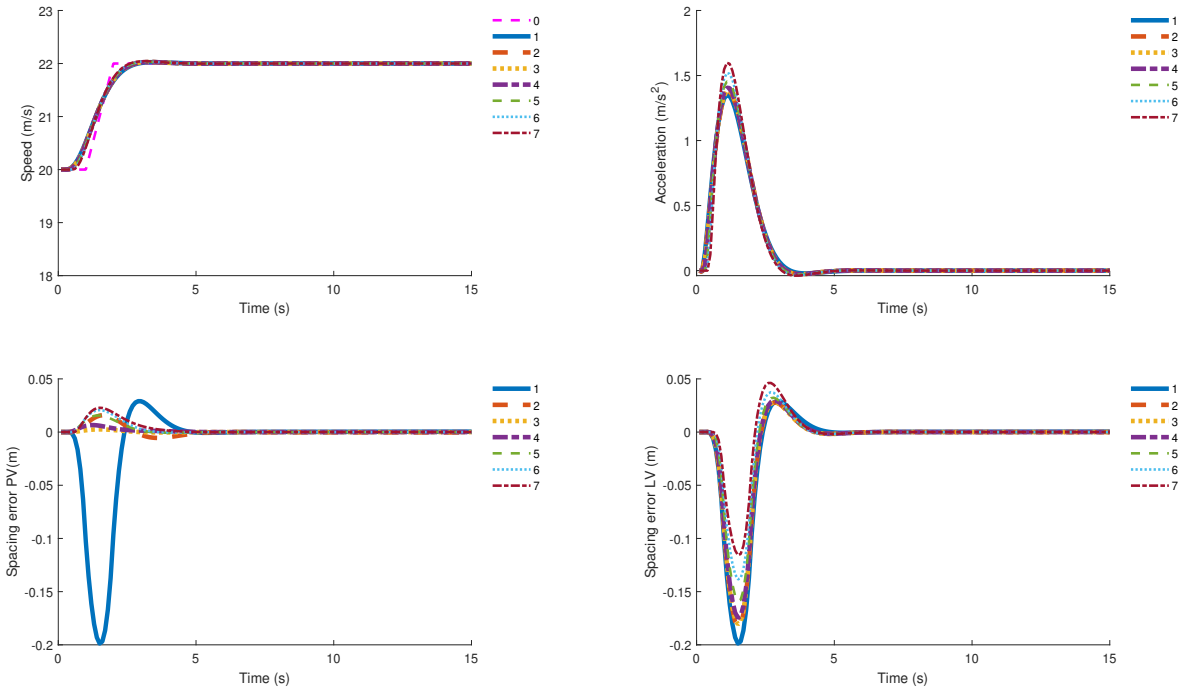

Figure 3.4. Rapid Acceleration under TPF

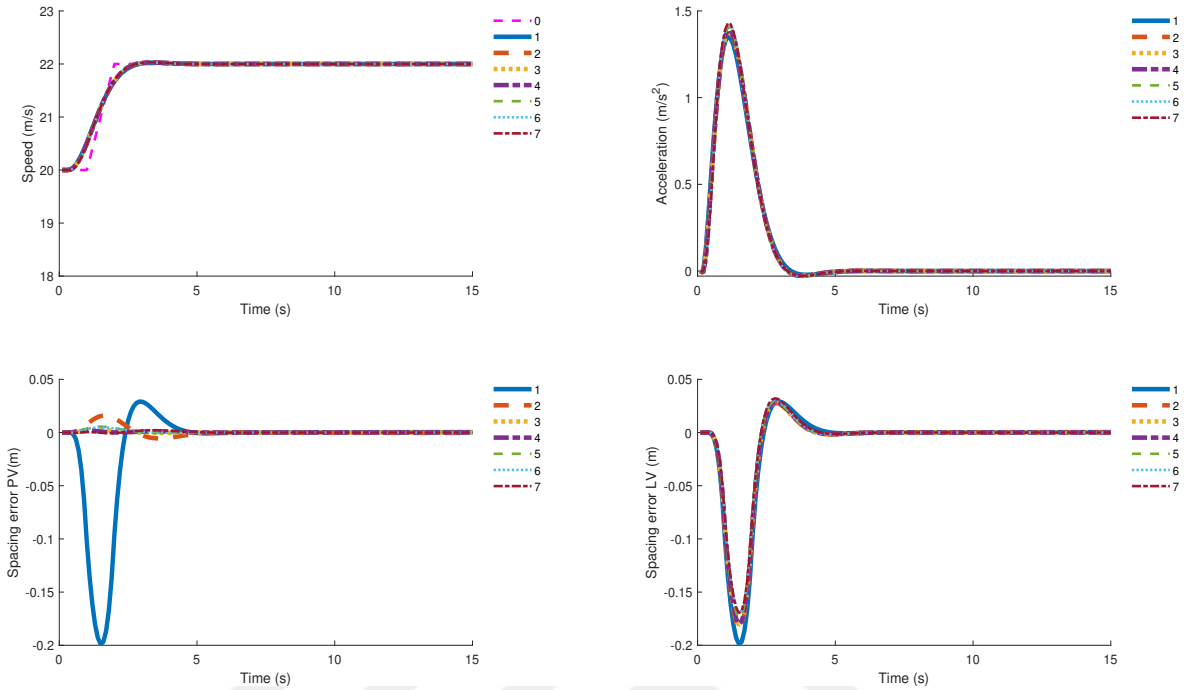

Figure 3.5. Rapid Acceleration under TPLF

Figures 3.2 - 3.5 show the platoon dynamics for $2m/s^2$ acceleration scenario under PF, PLF, TPF and TPLF topologies. The bottom left figure shows the spacing error to the preceding vehicle while the bottom right one presents the accumulated spacing error from the leader to the respective vehicle.

It can be claimed that transmitting the leader information to the follower vehicles clearly has a positive impact on the string stability as can be seen from the plots that are showing the spacing error to the preceding vehicle. Another benefit of transmitting leader information is, it enhances the transient performance of the platoon by reducing the deviation from the desired acceleration. However, it does not have any advantage on the steady-state error which is equal to zero for all of the topologies.

3.3.2. Test Scenario 2: Rapid Deceleration

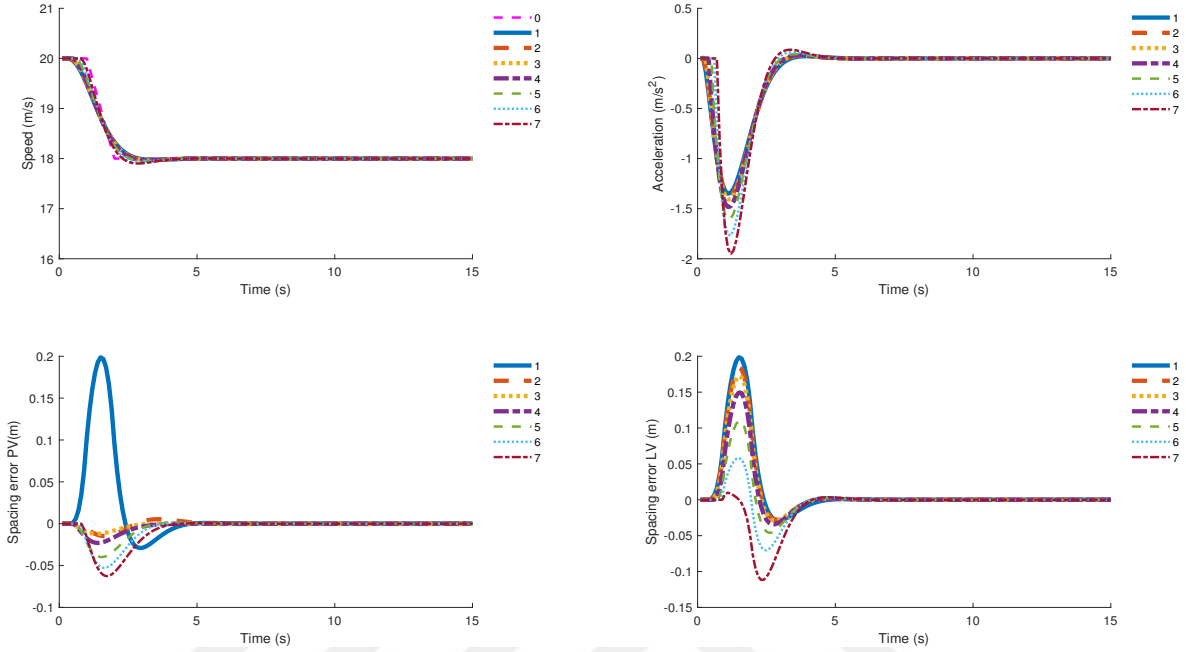

Figure 3.6. Rapid Deceleration under PF

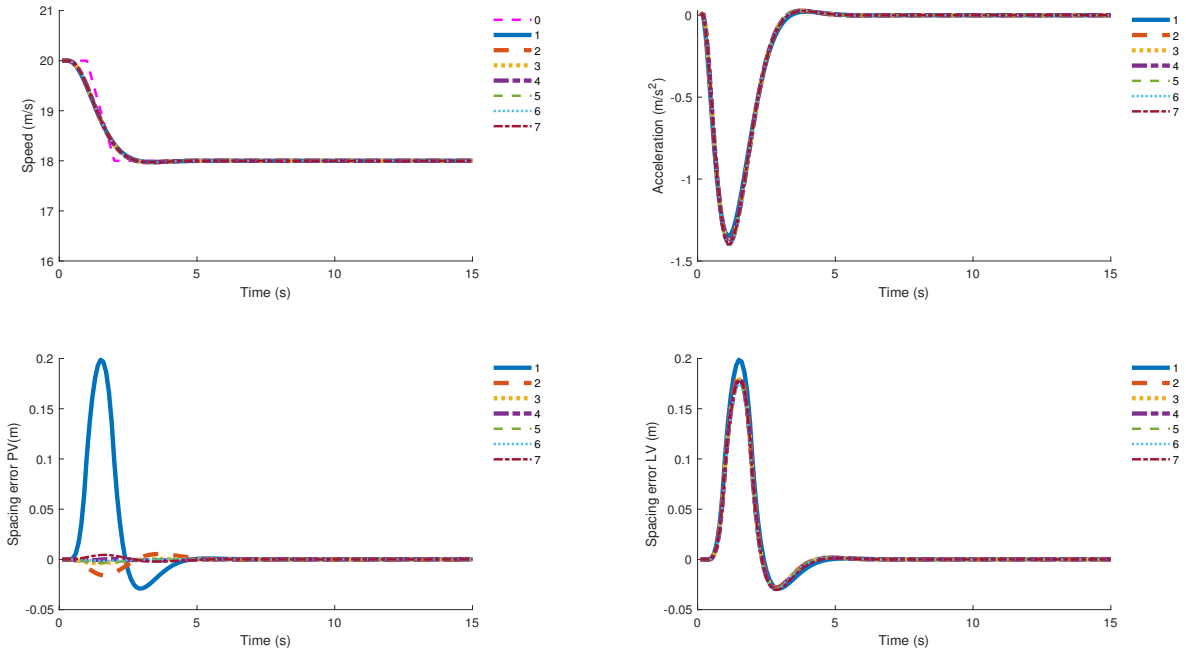

Figure 3.7. Rapid Deceleration under PLF

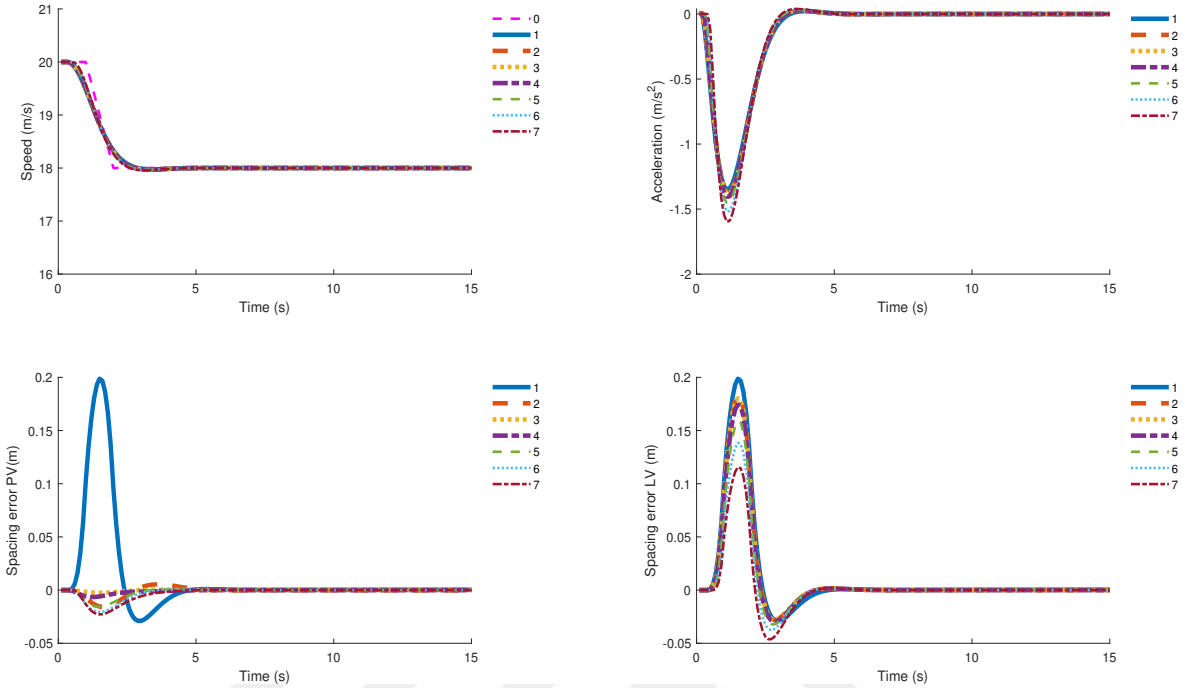

Figure 3.8. Rapid Deceleration under TPF

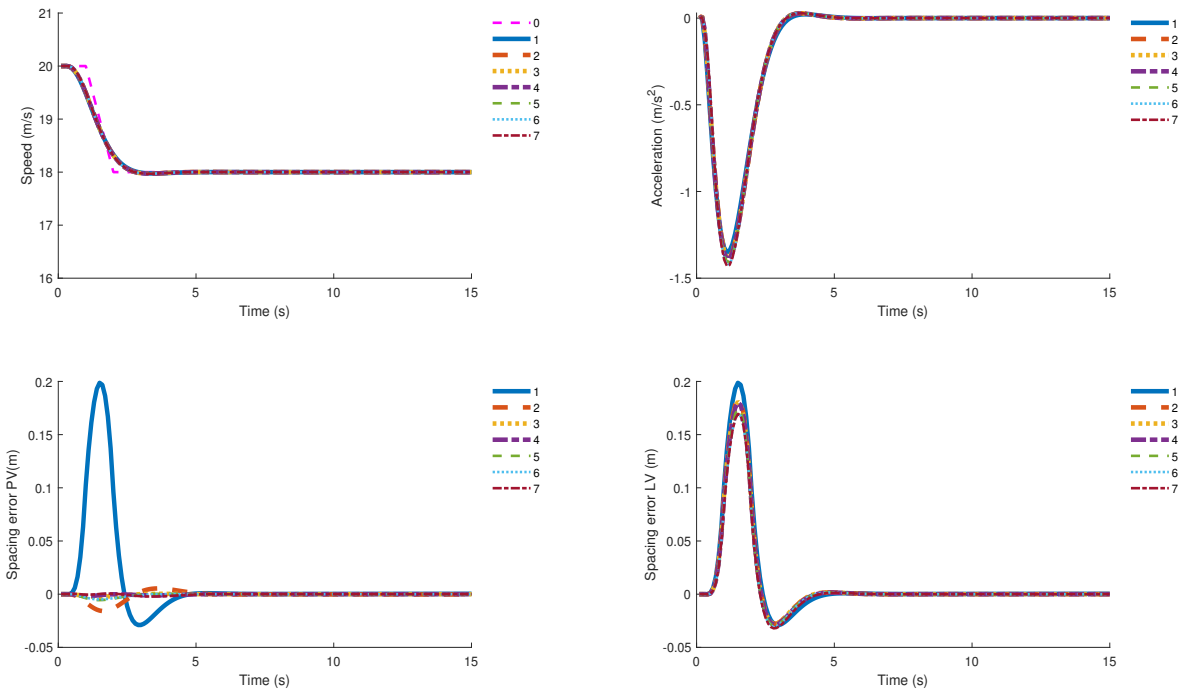

Figure 3.9. Rapid Deceleration under TPLF

Figures 3.6 - 3.9 show the platoon dynamics for $-2m/s^2$ deceleration scenario under PF, PLF, TPF and TPLF topologies. Similar to the results obtained in the previous test scenario, the leader information has the maximal impact on the last vehicle in the platoon to reduce its spacing error both to its preceding vehicle and the leader vehicle. No steady-state error is observed for all four topologies for the rapid deceleration scenario, either.

3.4. Summary of the Chapter

The optimization problem is discussed by including the details of the controller algorithm and the cost constraint formulations dependent on the unidirectional communication topology. Numerical values used on the simulation work are given and the simulation results under different test scenarios are shared.

In order to increase the level of complexity, bidirectional communications are investigated in Chapter 4. The communication topologies are extended as PF-bi, PLF-bi, TPF-bi, and TPLF-bi to also include the closest follower vehicle information. The bidirectional algorithm is tested under the same test scenarios defined in this chapter.

4. VEHICULAR PLATOON CONTROL WITH BIDIRECTIONAL COMMUNICATION TOPOLOGY

Another topic that draws a lot of attention in the vehicle platooning area is the bidirectional communication topology. It can be appraised as a quite powerful feature to have as it looks more promising for safety-related concerns by also having the follower vehicle states and does not necessarily require wireless communication for the topologies that do not have the leader vehicle states.

This chapter will present a DMPC algorithm for application of bidirectional links to PF, PLF, TPF and TPLF topologies for an eight-vehicle platoon including the leader and try to find a solution to unscalability identified by Seiler et al. in [12].

The bidirectional topology has been implemented under PF and PLF communication topologies and investigated for scalability and internal stability in [16]. It has proven that the addition of the leader states enhances the scalability of the platoon via distributed linear controllers under CDG policy but the string stability issue has not been addressed. The simulation results of the proposed DMPC algorithm will show that string stability can be achieved under four communication topologies with bidirectional links.

4.1. Bidirectional Communication Topology

The layout for the bidirectional communication topology that is going to be examined is shown in Figure 4.1. For simplicity and taking into consideration the possible use case of implementing a bidirectional topology, only the nearest following vehicle is assumed to be connected to the ego vehicle.

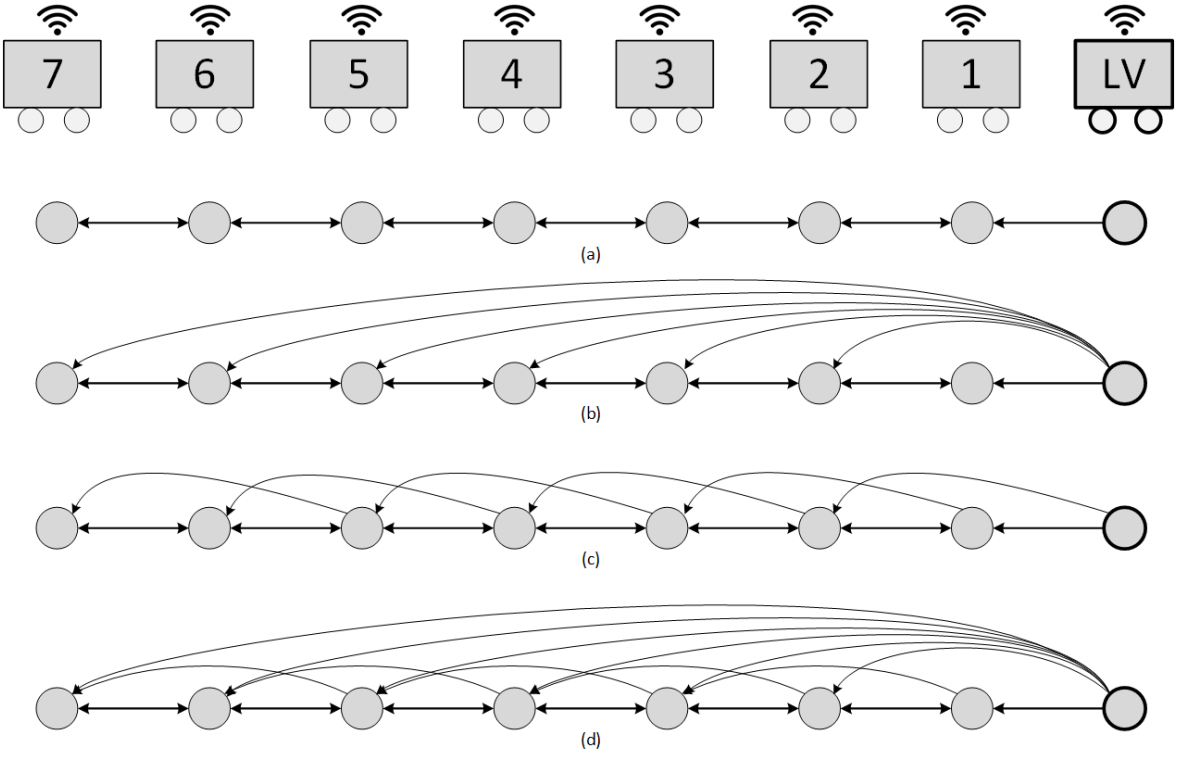

Figure 4.1. Bidirectional Communication Topology a) Predecessor Following (b) Predecessor Leader Following (b) Two-Predecessor Following (d) Two-Predecessor Leader Following

4.2. DMPC Algorithm for Bidirectional Topologies

A new configuration in MPC components has been examined in order to include following vehicle states into the equation without a need to compromise from the robustness that is achieved in Chapter 3. Figure 4.2 shows the updated algorithm with additional following neighbor outputs. The cost function components specific to the bidirectional topology and cost function weights are given in Section 4.3.

The cost function weights or the constraints are not designed for the best performance and are kept as the same for all topologies to be able to make a meaningful comparison between the previous and upcoming chapters.

Step 1: Assume that all the vehicles are moving at constant speed without position error and initialize assumed states of each node based on

$$u_i^a(k|0) = h_i(v_i(0)),$$
$$y_i^a(k|0) = y_i^p(k|0),$$

for $k = 0, 1, \ldots, N_p - 1$

Step 2: Calculate the control signal as follows:

if leader information available

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$, $y_{i-}^a(k|t)$, $y_{i+}^a(k|t)$ and $y_{des,i}(k|t)$ by using the average values of neighbor and leader outputs as constraints for the states of node i

else

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$, $y_{i-}^a(k|t)$ and $y_{i+}^a(k|t)$ by using the average values of neighbor outputs as constraints

end if

Step 3: Compute the optimal states for the node using the first element of the optimized control signal

Step 4: Calculate the assumed control input with additional last value to ensure that the node moves at a constant speed

$$u_i^a(k|t+1) = \begin{cases} u_i^*(k+1|t), & k = 0, 1, ..N_p - 2 \\ h_i(v_i{}^*(N_p|t)), & k = N_p - 1 \end{cases}$$

Step 5: Calculate assumed outputs $y_i^a(k|t+1)$ using the nonlinear vehicle model given in (2.6)

Step 6: Transmit $y_i^a(k|t+1)$, receive $y_{i+}^a(k|t)$ and $y_{i-}^a(k|t+1)$ for the next time step

Step 7: Increase t and go to Step 2.

Figure 4.2. Distributed MPC Algorithm for Platoon with Bidirectional Topology

4.3. Optimization Problem Definition for Bidirectional Topologies

The data originating from the following vehicle is embedded into the MPC as described in the following sections. The basis structure that was proposed in Chapter

3 is remarkably conserved and necessary modifications applied to the optimization problem are explained here.

4.3.1. Cost

The cost function given in (3.1) is updated to also include a component for the bidirectional topology. The cost function will have the following final form after addition of the following neighbor outputs:

$$J(z) = J_y(z) + J_u(z) + J_y^a(z) + J_{y_{i-}}^a(z) + J_{y_{i+}}^a(z) \tag{4.1}$$

where $J_{y_{i+}}^a(z)$ represents the assumed trajectory of the nearest following vehicle and can be described by

$$J_{y_{i+}}^a(z) = \sum_{j \in N_i} \left(y_i^p(k|t) - y_{i+}^a(k|t) - \tilde{d}_{i,i+}\right)^T P_i \left(y_i^p(k|t) - y_{i+}^a(k|t) - \tilde{d}_{i,i+}\right) \tag{4.2}$$

Similar to (3.10), $\tilde{d}_{i,i+}$ term is devoted for the desired distance vector between node i and $i+$ to satisfy the constant spacing among each vehicle and its successor. The remaining part of the cost function is preserved as defined in (3.7)-(3.9) with weights given in Table 4.1.

4.3.2. Constraints

4.3.2.1. Input Constraints. There is no change on the input constraints regarding the bidirectional topology as it does not necessarily affect the magnitude of the control signal; the configuration is kept the same for this topology, as well.

4.3.2.2. Output Constraints. The nearest following vehicle's assumed states are also included as an additional output constraint both to minimize the spacing error and to

Table 4.1. Weights in the cost functions

Weights	**PF**	**PLF**	**TPF**	**TPLF**
F_i	F_i=10I_2, $i \in N$	F_i=10I_2, $i \in N$	F_i=10I_2, $i \in N$	F_i=10I_2, $i \in N$
G_i	G_1=0, G_i=5l_2, $i \in N\backslash\{1\}$	G_1=0, G_i=5I_2, $i \in N\backslash\{1\}$	G_1=0, G_i=5I_2, $i \in N\backslash\{1\}$	G_1=0, G_i=5I_2, $i \in N\backslash\{1\}$
Q_i	Q_1=10I_2, Q_i=0, $i \in N\backslash\{1\}$	Q_i=10I_2, $i \in N$	Q_1=10I_2, Q_2=10I_2, Q_i=0, $i \in N\backslash\{1,2\}$	Q_i=10I_2, $i \in N$
R_i	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$	R_i=I_2, $i \in N$
P_i	P_7=0, P_i=5I_2, $i \in N\backslash\{7\}$	P_7=0, P_i=5I_2, $i \in N\backslash\{7\}$	P_7=0, P_i=5I_2, $i \in N\backslash\{7\}$	P_7=0, P_i=5I_2, $i \in N\backslash\{7\}$

avoid any possible collision during rapid deceleration scenarios as follows:

$$y_i^p(N_p|t) = \frac{1}{|N_i|}\sum y_{i+}^a(N_p|t) + d_{i,i+} \tag{4.3}$$

4.3.3. Problem Solver

For the solution of the problem, *fmincon* function is used with the same configuration as in Chapter 3:

- Algorithm: Interior-point algorithm
- Optimality Tolerance: 10^{-5}
- Function Tolerance: 10^{-5}

- Step Tolerance: 10^{-6}

4.4. Test Scenarios and Simulation Results for Bidirectional Topologies

4.4.1. Test Scenario 1: Rapid Acceleration

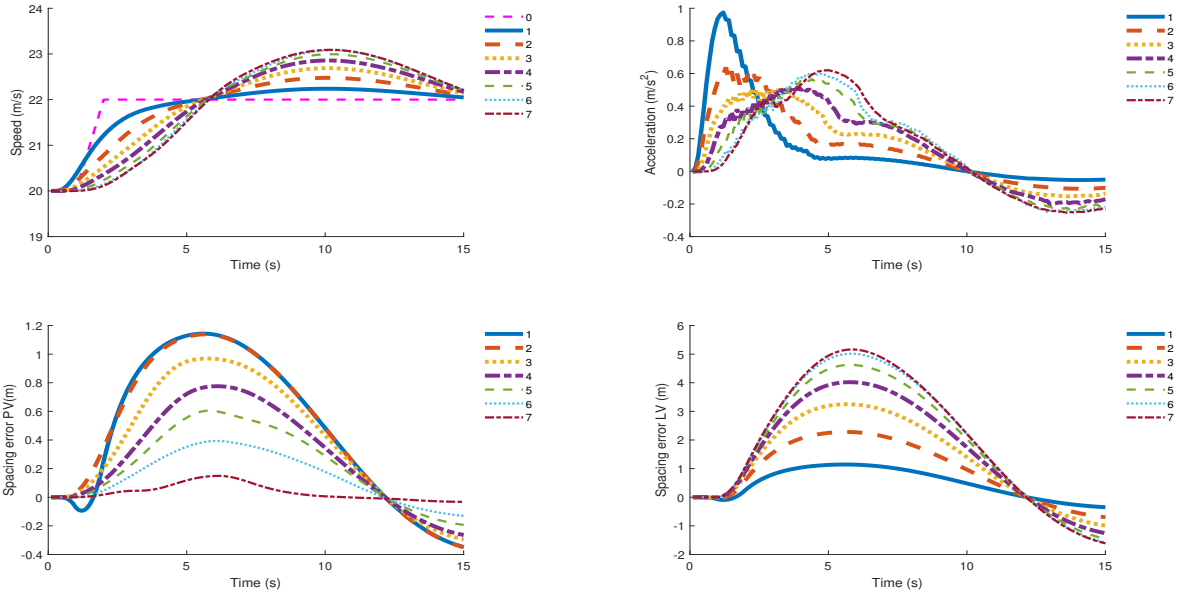

Figure 4.3. Rapid Acceleration under PF-bi

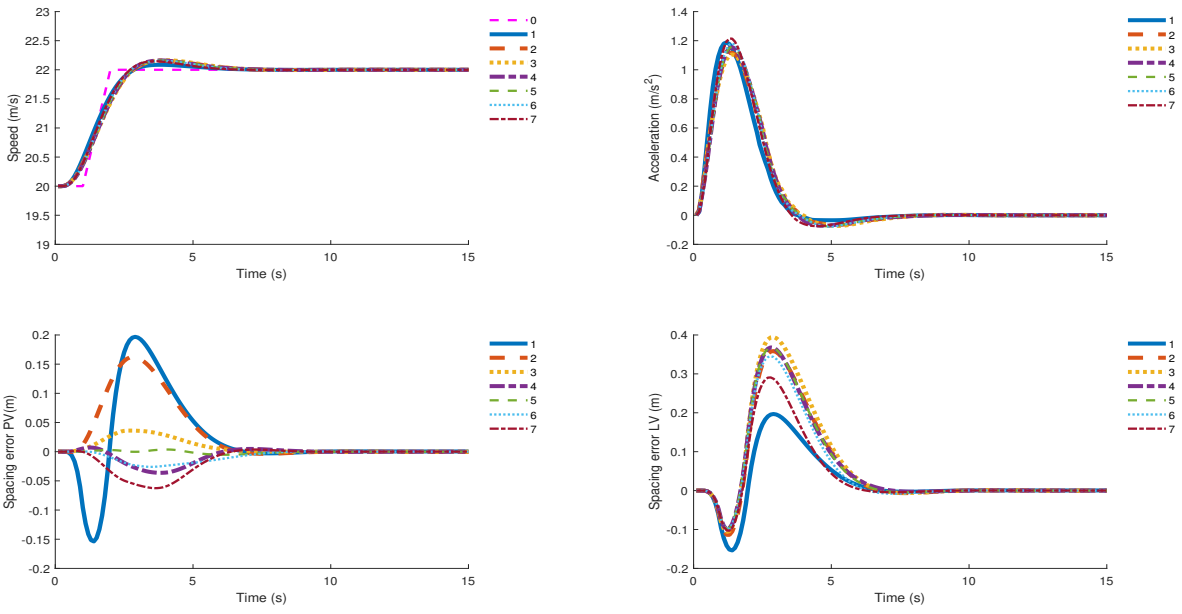

Figure 4.4. Rapid Acceleration under PLF-bi

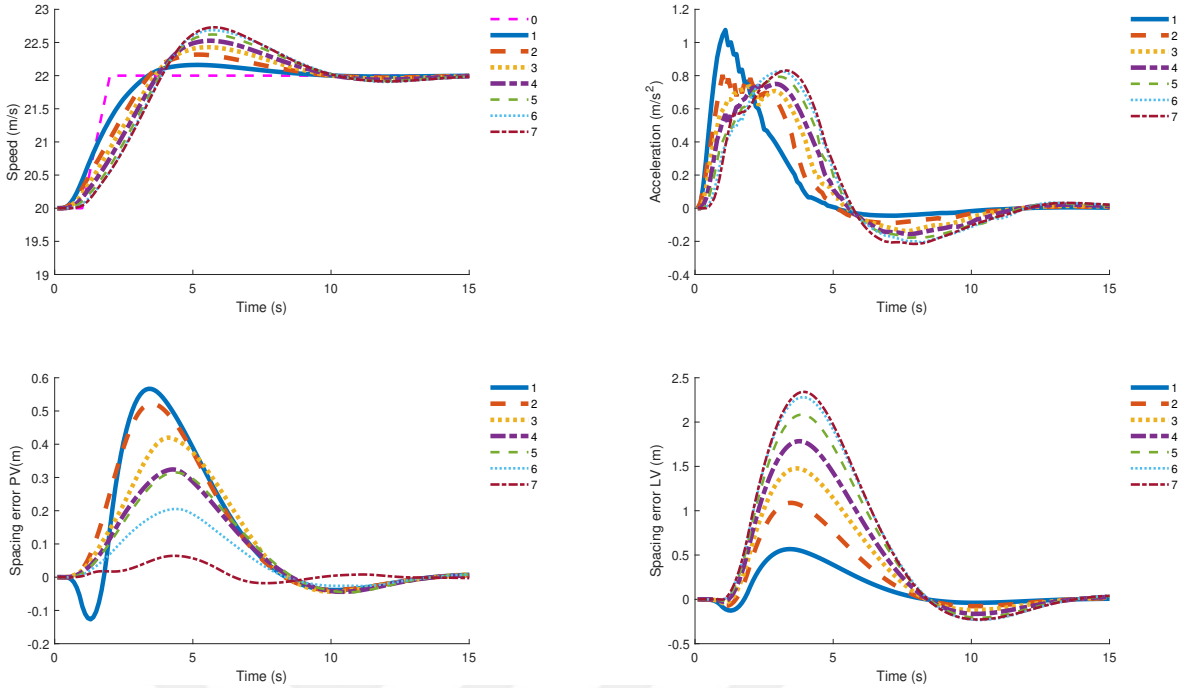

Figure 4.5. Rapid Acceleration under TPF-bi

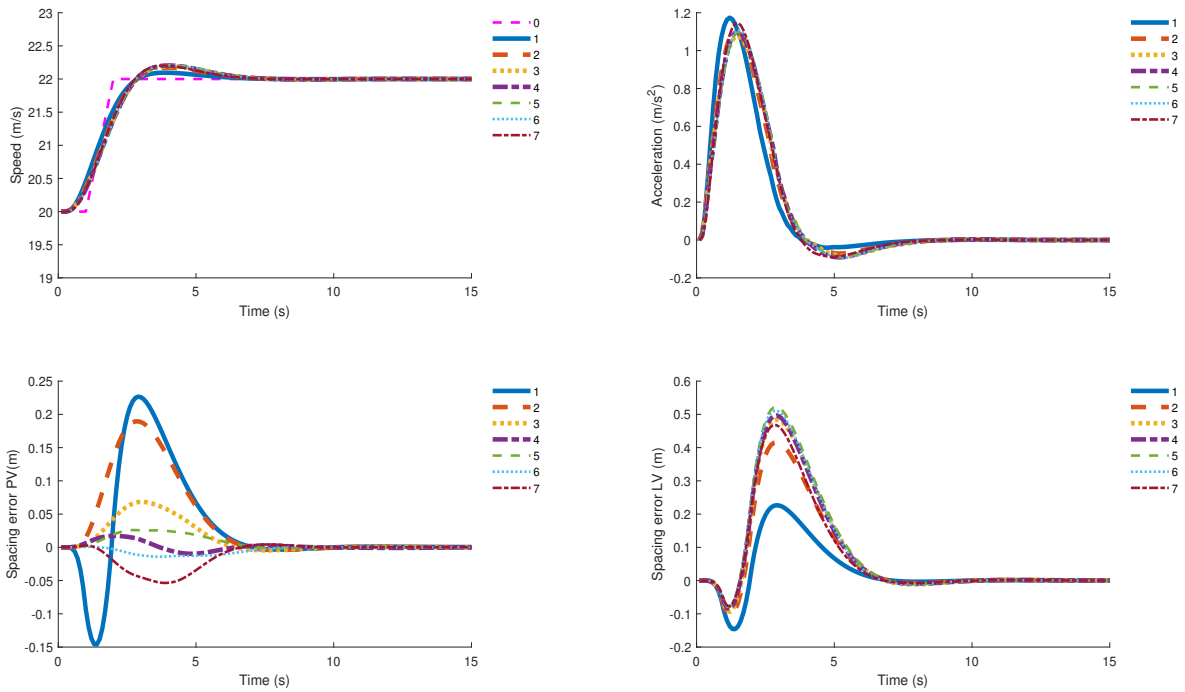

Figure 4.6. Rapid Acceleration under TPLF-bi

Figures 4.3 - 4.6 show the platoon dynamics under $2m/s^2$ acceleration scenario with bidirectional topology. It is surprising to see that the bidirectional topology satisfies the string stability even without having the leader set points, yet with increased error margin. Receiving the leader states has had major impact for improving the transient response of the overall platoon. Inclusion of follower vehicle states increased the complexity of the optimization problem as especially can be seen from Figures 4.3 and 4.5. In these, increased spacing error and non-smooth control signal are drawing attention. PLF-bi and TPLF-bi topologies display that having the leader states in the algorithm helped to reduce the magnitude of the spacing error.

4.4.2. Test Scenario 2: Rapid Deceleration

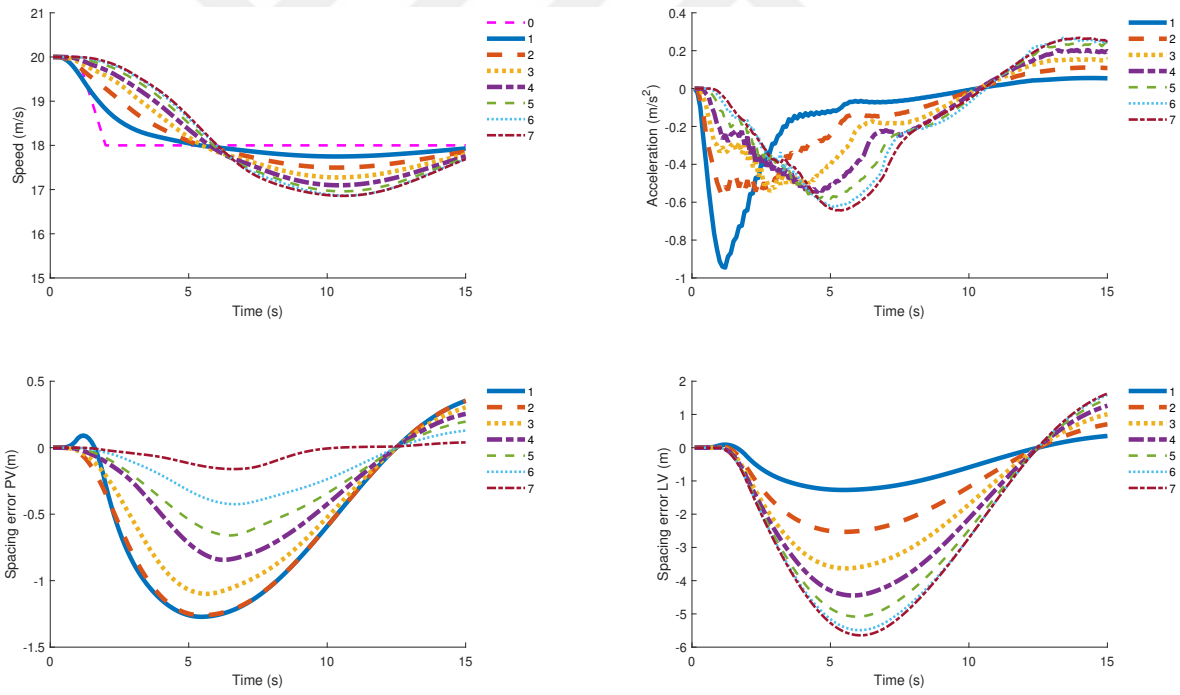

Figure 4.7. Rapid Deceleration under PF-bi

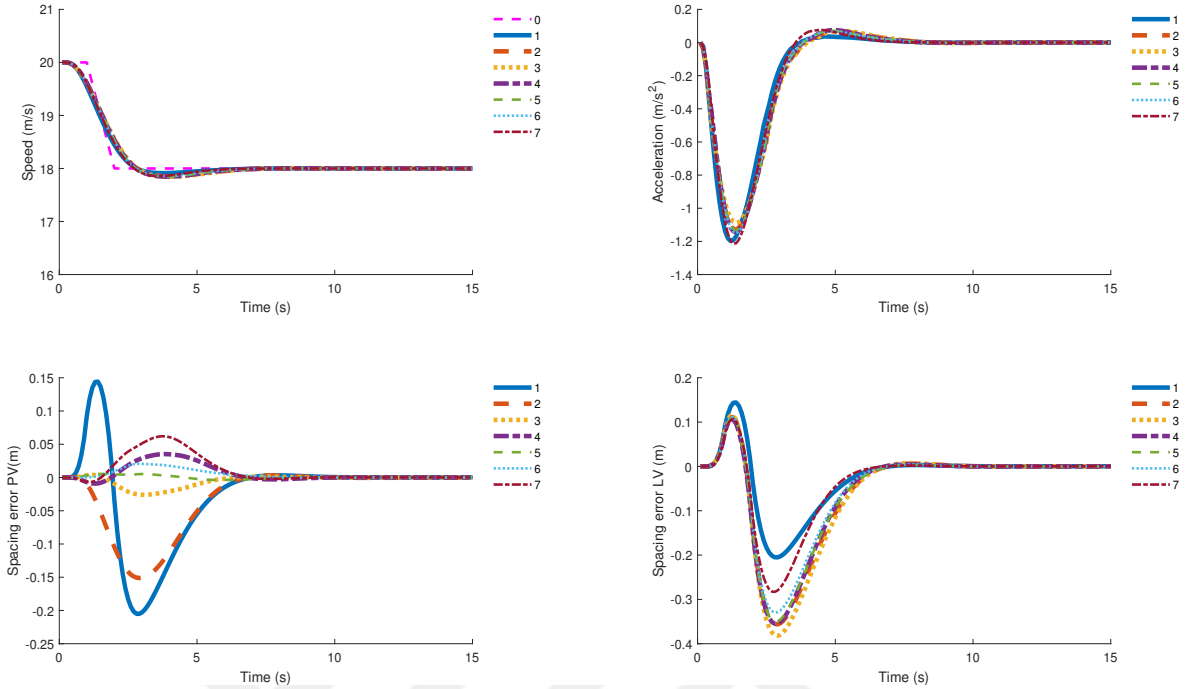

Figure 4.8. Rapid Deceleration under PLF-bi

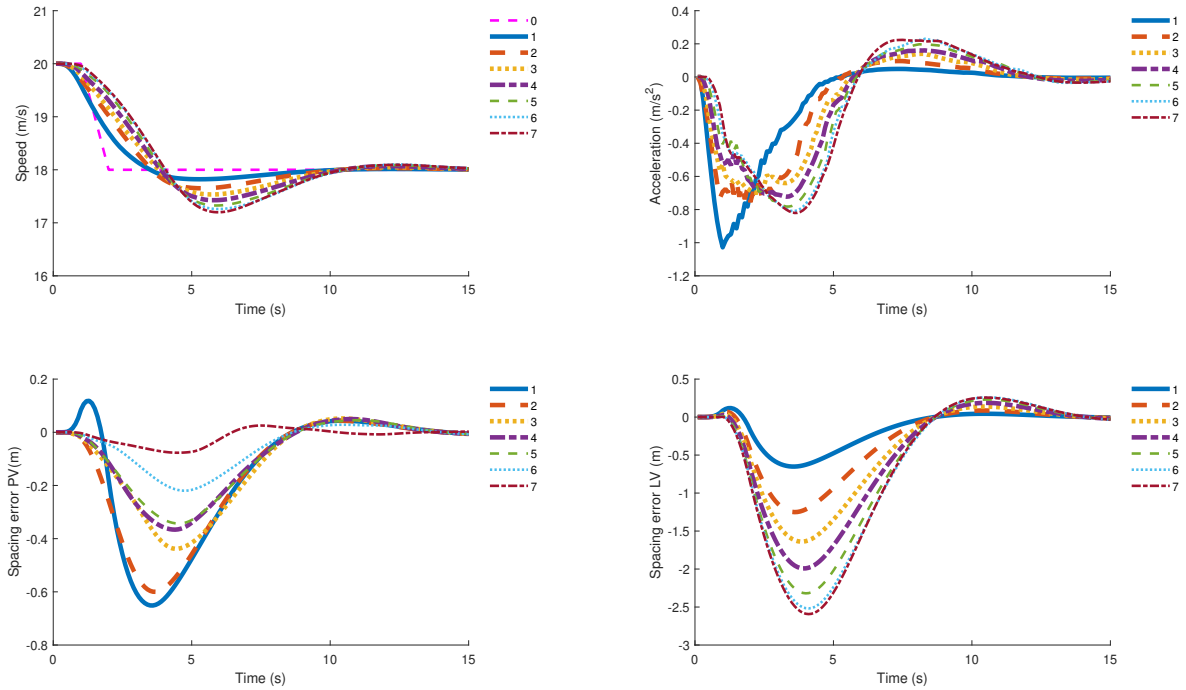

Figure 4.9. Rapid Deceleration under TPF-bi

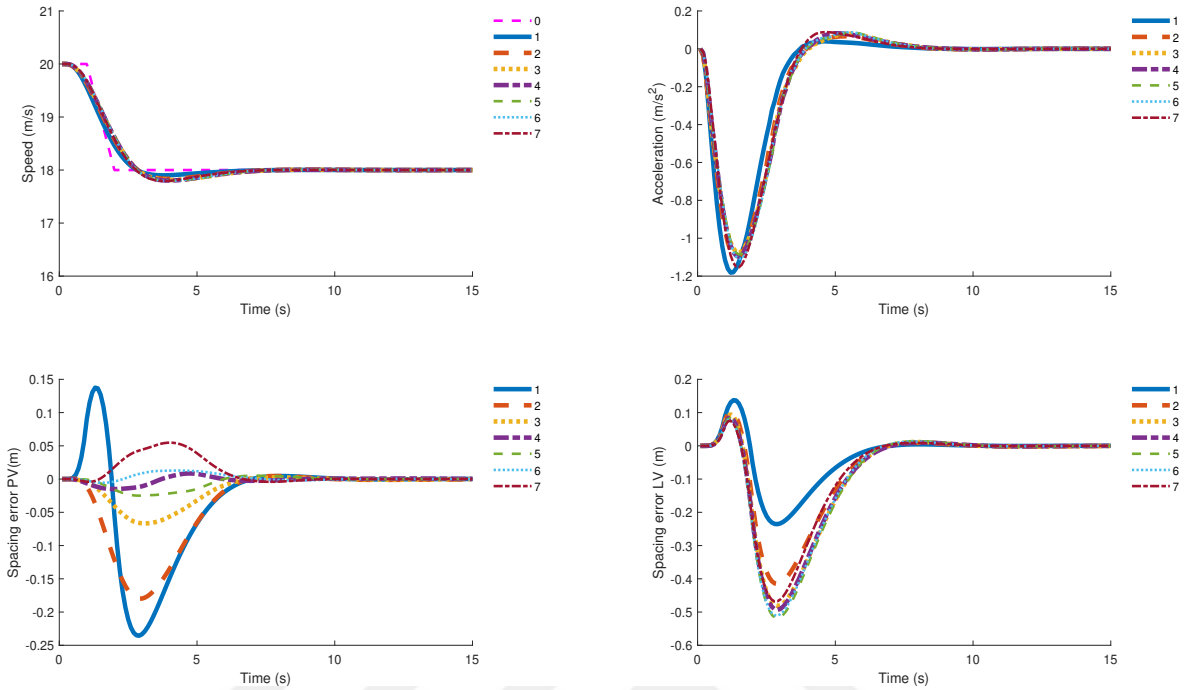

Figure 4.10. Rapid Deceleration under TPLF-bi

Figures 4.7 - 4.10 show the platoon dynamics under $-2m/s^2$ deceleration scenario with bidirectional topology. From Figures 4.7 and 4.9, it is observed that the bidirectional topologies without the leader information have adversely affected the settling time even though no spacing error occurs in none of the investigated topologies. The simulation results are in harmony with the rapid acceleration scenario, and the string stability is preserved under all bidirectional topologies.

Only the spacing error of the last vehicle in the platoon is not quite positioned in the right order according to the string stability definition. However, it should be highlighted that the last vehicle has no follower to limit its states and thus it has larger error margin comparing to its closest two predecessor neighbors and it still does not exceed the absolute spacing error of the first two vehicles in the platoon.

4.5. Summary of the Chapter

In this chapter, bidirectional topology is examined based on the previously defined PF, PLF, TPF and TPLF topologies. Bidirectional topology provided string stable results even without having the leader vehicle set points transmitted but the leader vehicle information helped to reduce the spacing errors under all topologies. The weights used in the cost function assisted to achieve the string stability but it is also noted that the same weights increased the error margin in comparison to the previous implementations.

After adding the follower vehicle information to the optimization problem, it has become more complex to solve as can be seen from the plots for PF topology under both of the test scenarios and the results have become smoother once the leader states are included. Taking this into consideration, even though string stability is achieved for all the topologies, further modification might be needed in order to improve the transient response for the bidirectional topologies where the leader information is not available. Increased settling time can be limiting and unpractical considering the platoon systems that necessitate fast responses.

It should also be noted that the TPF topology performs better than PF topology even though it has not received the leader states. The second predecessor information seems to be helping to reduce the spacing error and the settling time in a significant way.

In Chapter 1, the communication delay is defined as another challenge of the platooning. This topic is investigated in the next chapter by taking the unidirectional DMPC algorithm as the basis. In this manner, the next chapter will be an extension to Chapter 3 with the application of communication delay to PF, PLF, TPF and TPLF topologies.

5. VEHICULAR PLATOON CONTROL WITH COMMUNICATION DELAY

Communication delay and packet loss are the most important issues likely to be encountered in real-life implementations of the platoon systems. Vehicles that are equipped with V2V technologies are able to receive and transmit data but this wireless communication protocol is subject to different kind of failures and delays as any other technology.

Considering above, this chapter is devoted to addressing communication delays between vehicles that are connected with wireless communication. It is worth mentioning that, in this investigation, the focus is limited to uniform communication delay as discussed in earlier studies [18], [23]. Random delays which will more likely to occur in real-life applications are left as a potential research topic for future.

Wireless communication dropout is another network malfunction that may be needed to consider in real-life applications. Demonstrating the effects of this network malfunction is the second focus of this chapter. Transmitted leader vehicle states may not be delivered to the following vehicles when the network reaches to its maximum bandwidth capacity. In this scenario, each follower would need to use the on-board sensor information which only contains its neighboring vehicle states to calculate its own trajectory. In order to simulate this use case, this chapter also examines the effects of topology change in the platoon while steering.

5.1. Communication Topology with Uniform Delay

It is assumed that the communication delay is affected by the distance to the leader vehicle. In other words, for the follower vehicles, getting farther away from the leader vehicle causes receiving data with cumulatively increased delay. Signal distortion is formulated as the first vehicle in the platoon will be receiving the leader states with

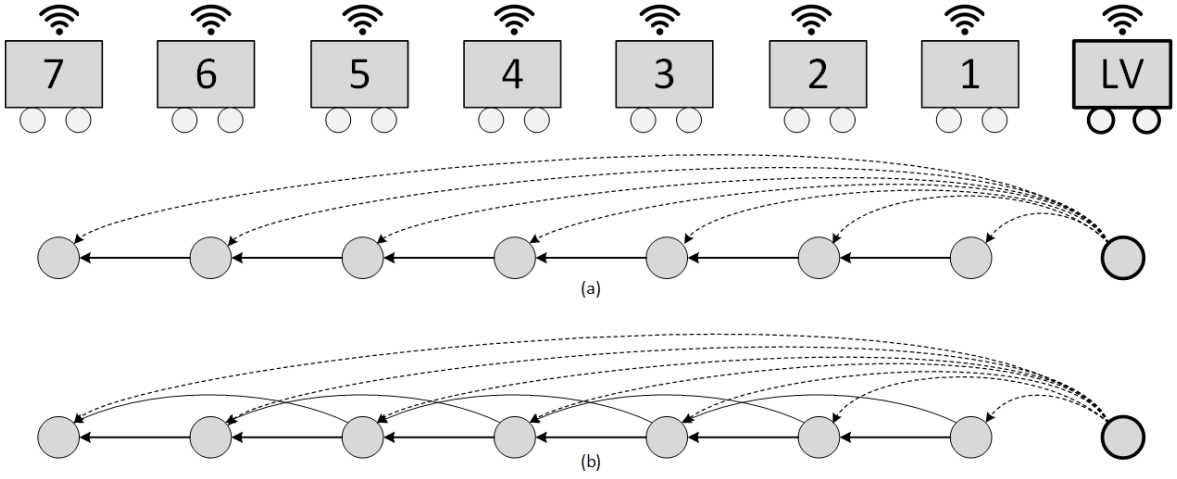

Figure 5.1. Communication Topology with Delay a) Predecessor Leader Following (b)Two-Predecessor Leader Following

1 sample time delay, the second vehicle is with 2 sample time delay and so on till the last vehicle.

Considering the assumption of leader states are being delayed during wireless communication, only PLF and TPLF topologies are further investigated. Firstly, the DMPC response is checked under PLF and TPLF topologies without any modification in the algorithm other than the addition of the delay to the leader broadcast data.

By examining Figures 5.2 and 5.3, it can be obviously claimed that communication delay causes string unstable response as the spacing error to the preceding vehicle keeps increasing towards the end of the platoon. Besides the steady-state error that is observed in the comparative position plots, the major speed undershoot caused by the lagged information of the leader position is determined. Responses for both PLF and TPLF topologies are affected in a similar way except that in TPLF, the second predecessor states appear to have helped to smooth the peaks a bit more.

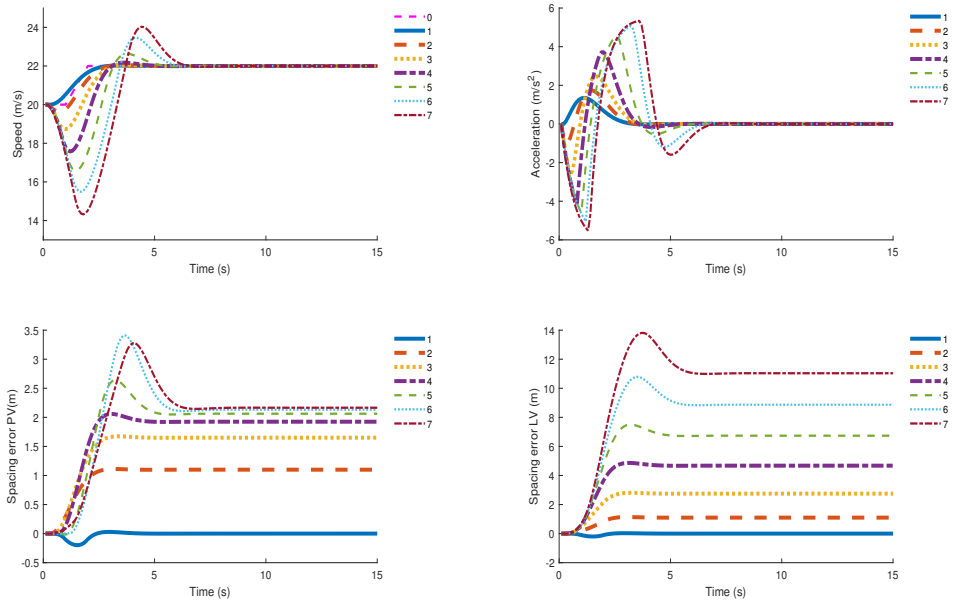

Figure 5.2. Rapid Acceleration under PLF with Uniform Communication Delay

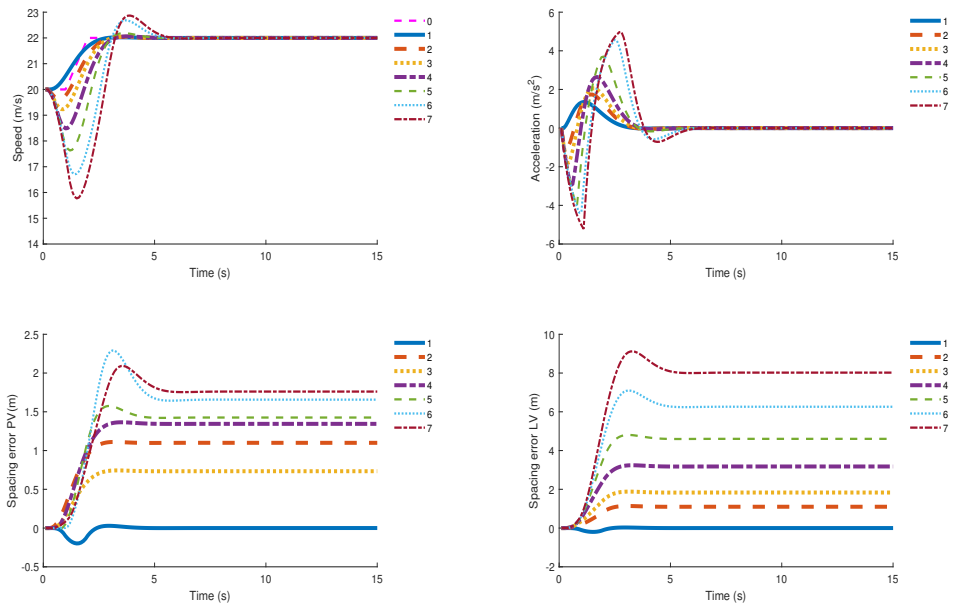

Figure 5.3. Rapid Acceleration under TPLF with Uniform Communication Delay

Without any doubt, it can be said that the controllers can not tolerate the communication delay, as the nodes do not have the recent information for solving the control problem. Thus a further modification is needed.

5.2. DMPC Algorithm for Unidirectional Topologies with Uniform Communication Delay

In order to cope with the delay in the leader output $y_{des,i}$ which contains the desired position ($P_{des,i}$) and velocity trajectories ($V_{des,i}$), an additional change is needed in the receiver node. The desired position data that belongs to a former time step limits the optimization problem to avoid an illusory collision.

To overcome this, a new output $y'_{des,i}(k|t) = [P'_{des,i}(k|t) \quad V'_{des,i}(k|t)]$ is constructed in the receiver node to ignore the past data and complete the trajectory matrix with the generated position and the velocity. The generated position is calculated by using the last sample of vehicle velocity received from the leader based on own MPC model of each vehicle. Even though the vehicle parameters are different from the leader vehicle, this does not lead to a big issue; in contrary, it is extremely helpful to set the desired set points and more importantly keep the platoon stable. The vehicle velocity is assumed to be constant during the generated steps. The visual representation of the necessary changes made in trajectory matrices is given in Figure 5.4. Figure 5.5 represents the algorithm updates in the DMPC to compensate for the communication delay.

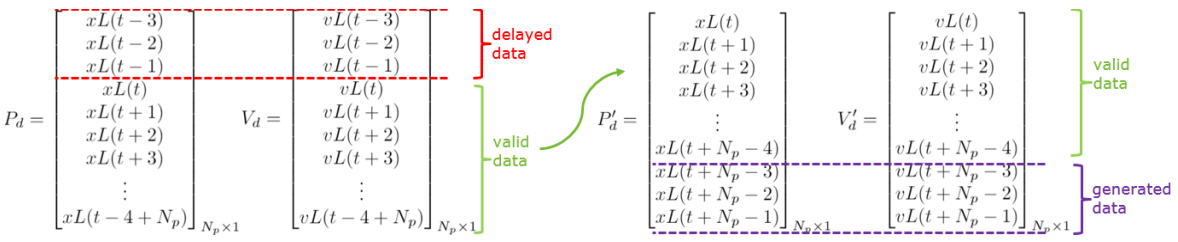

Figure 5.4. Desired Trajectory Updates in Matrices for Uniform Delay Handling

Step 1: Assume that all the vehicles are moving at constant speed without position error and initialize assumed states of each node based on

$$u_i^a(k|0) = h_i(v_i(0)),$$
$$y_i^a(k|0) = y_i^p(k|0),$$

for $k = 0, 1, \ldots, N_p - 1$

Step 2: Calculate the control signal as follows:

if leader information available

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$, $y_{i-}^a(k|t)$, $y'_{des,i}(k|t)$ and using the average values of neighbor and leader outputs as constraints for the states of node i

else

Optimize node i according to $x_i(t)$, $y_i^a(k|t)$ and $y_{i-}^a(k|t)$ by using the average values of neighbor outputs as constraints

end if

Step 3: Compute the optimal states for the node using the first element of the optimized control signal

Step 4: Calculate the assumed control input with additional last value to ensure that the node moves at a constant speed

$$u_i^a(k|t+1) = \begin{cases} u_i^*(k+1|t), & k = 0, 1, ..N_p - 2 \\ h_i(v_i{}^*(N_p|t)), & k = N_p - 1 \end{cases}$$

Step 5: Calculate assumed outputs $y_i^a(k|t+1)$ using the nonlinear vehicle model given in (2.6)

Step 6: Transmit $y_i^a(k|t+1)$, receive and $y_{i-}^a(k|t+1)$ for the next time step

Step 7: Increase t and go to Step 2.

Figure 5.5. Distributed MPC Algorithm for Platoon with Uniform Communication Delay

5.3. Optimization Problem Definition for Bidirectional Topologies with Uniform Communication Delay

The main objective of the platoon is to keep the outputs close to the original responses shown in Chapter 3 by aiming small spacing error with string stable behavior for both PLF and TPLF topologies even with the applied communication delay to the leader states.

5.3.1. Cost

The cost function formulation in Chapter 3 is reused with the same weights defined in Table 3.1 but the data to be inputted to the cost function is treated on the receiver node.

5.3.2. Constraints

5.3.2.1. Input Constraints. Braking and driving torque lower and upper limits are set to have the acceleration values equal to $-6m/s^2$ and $6m/s^2$, respectively. These values are calculated based on (3.11).

5.3.2.2. Output Constraints. Output constraints are directly affected by any modification made on $y_{des,i}$, so it is supremely critical that the leader position is updated to compensate the communication delay using the MPC model and the vehicle velocity obtained in the last step.

5.3.3. Problem Solver

For the solution of the problem $fmincon$ function is used with the same configuration as in Chapters 4 and 5:

- Algorithm: Interior-point algorithm

- Optimality Tolerance: 10^{-5}
- Function Tolerance: 10^{-5}
- Step Tolerance: 10^{-6}

5.4. Test Scenarios and Simulation Results for Bidirectional Topologies with Uniform Communication Delay

5.4.1. Test Scenario 1: Rapid Acceleration

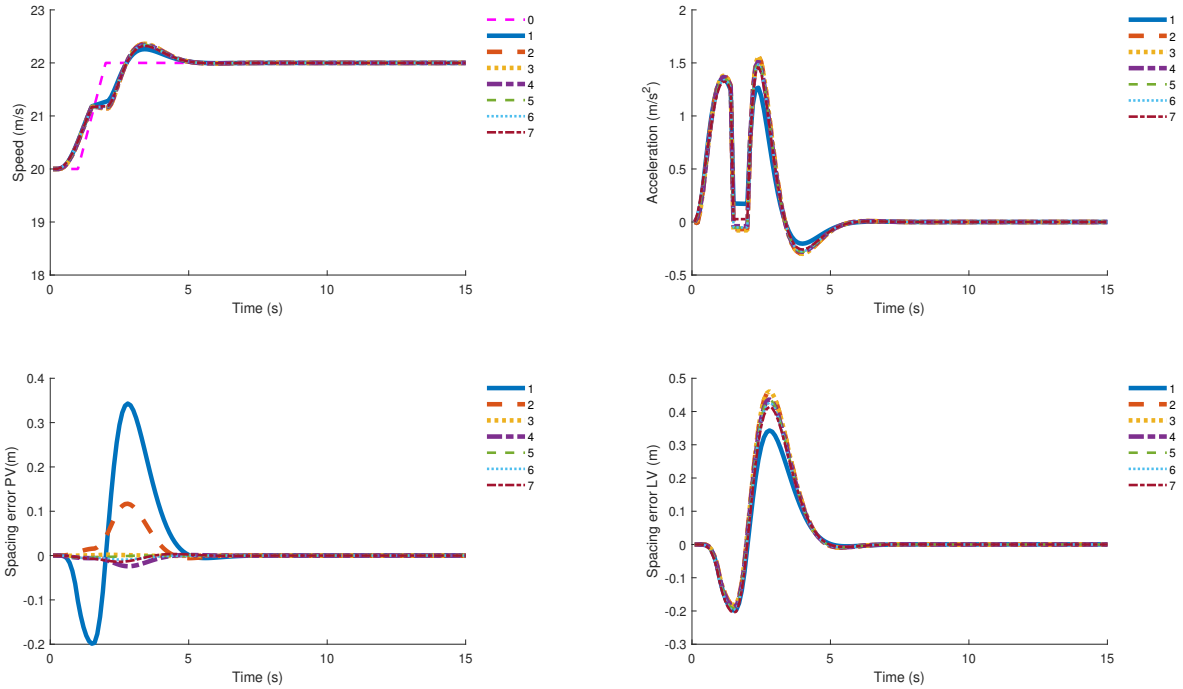

Figure 5.6. Rapid Acceleration under PLF with Uniform Communication Delay

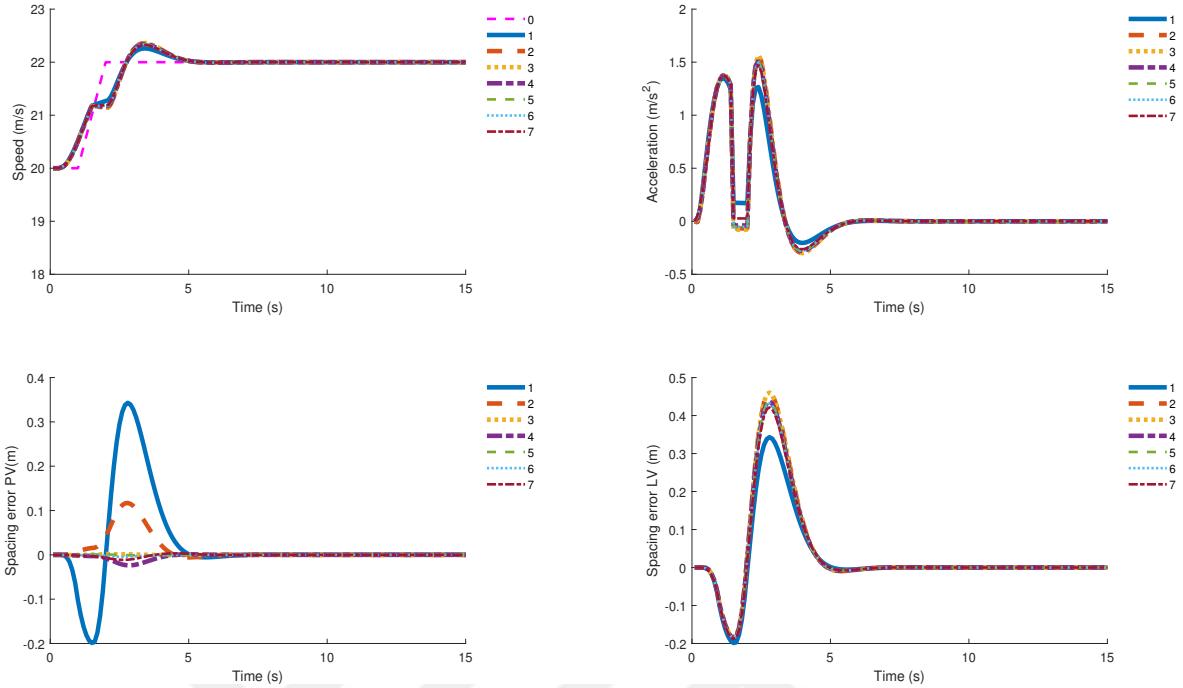

Figure 5.7. Rapid Acceleration under TPLF with Uniform Communication Delay

5.4.2. Test Scenario 2: Rapid Deceleration

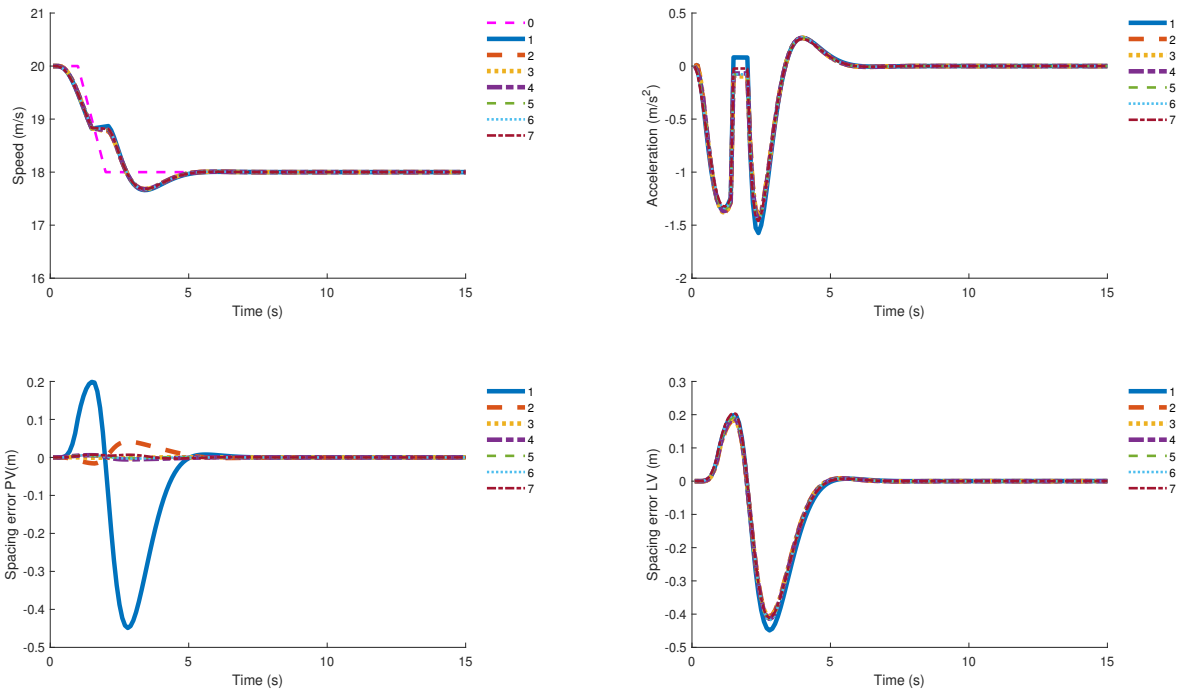

Figure 5.8. Rapid Deceleration under PLF with Uniform Communication Delay

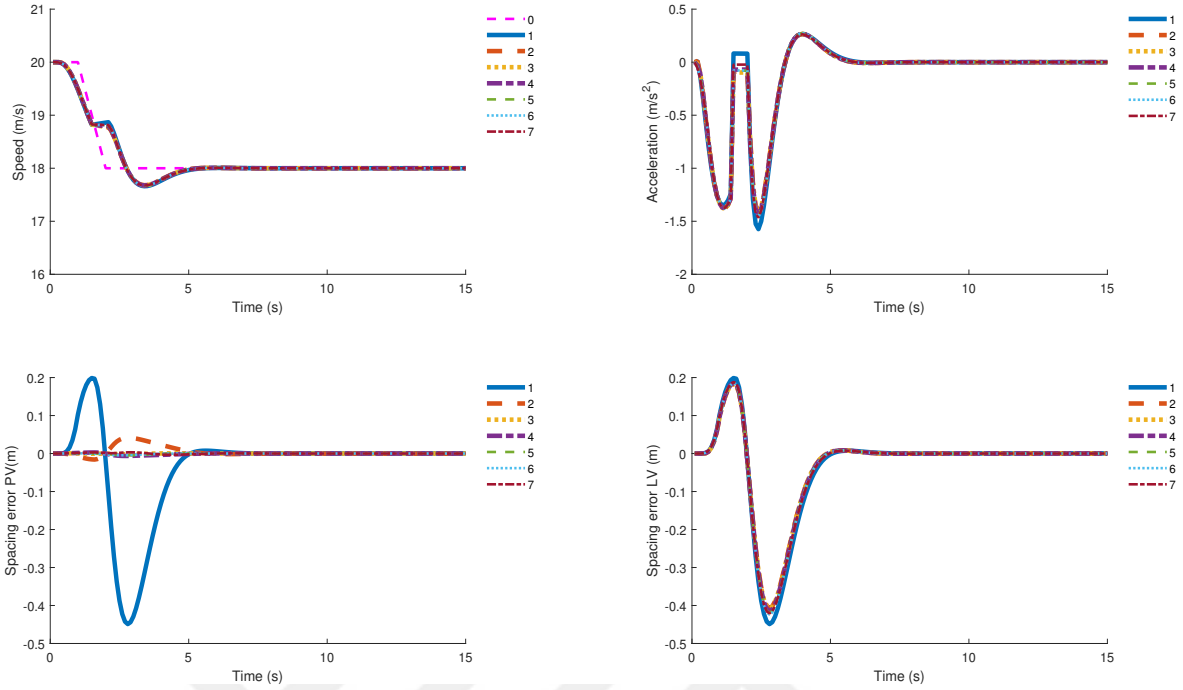

Figure 5.9. Rapid Deceleration under TPLF with Uniform Communication Delay

As shown in Figures 5.6 - 5.9, string stability is restored in both topologies with a lot less overshoot in the desired speed and no steady-state error in the desired spacing. Although the steady-state performance of the DMPC algorithm for communication delay is satisfying, more effort needs to be put to improve the transient response since rapid changes in the acceleration will not be acceptable considering the driver comfort and the fuel consumption.

Further investigation was made to avoid the short pause in the velocity plot and high changes in the acceleration plot. The very first thing to try was increasing the prediction horizon that is being shrunk by the communication delay indirectly. The prediction horizon should be larger than the delay horizon for MPC to compensate for the negative effects of delay [24].

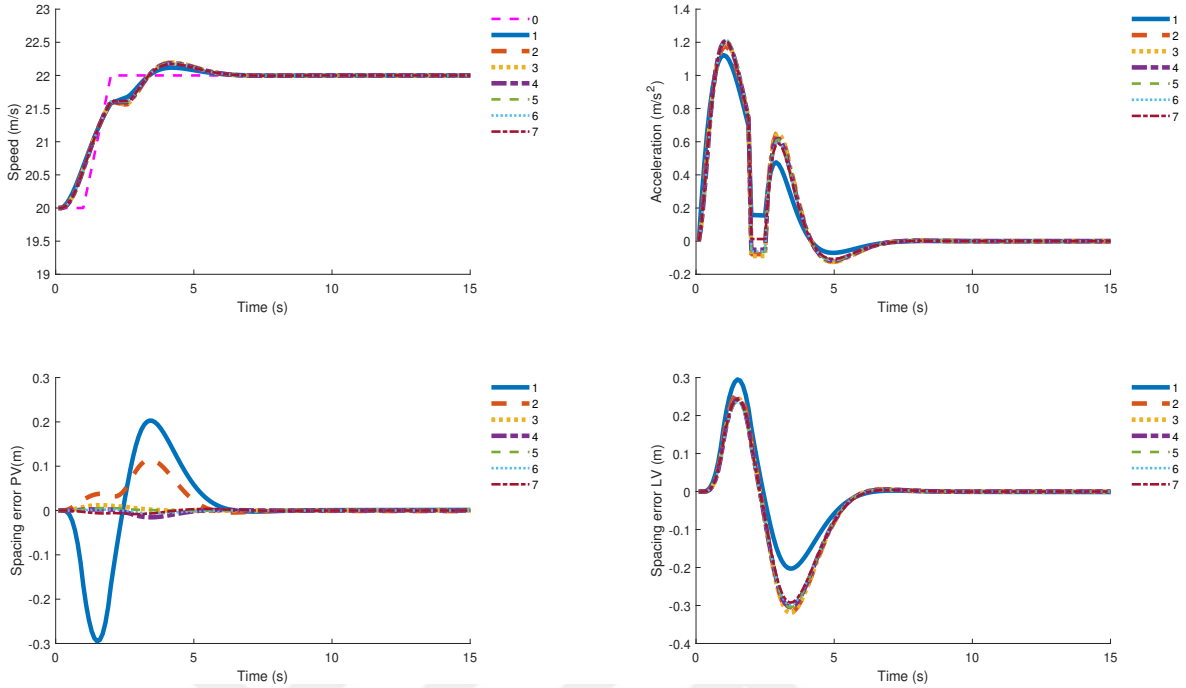

Figure 5.10. Rapid Acceleration under PLF with Communication Delay with $N_p = 25$

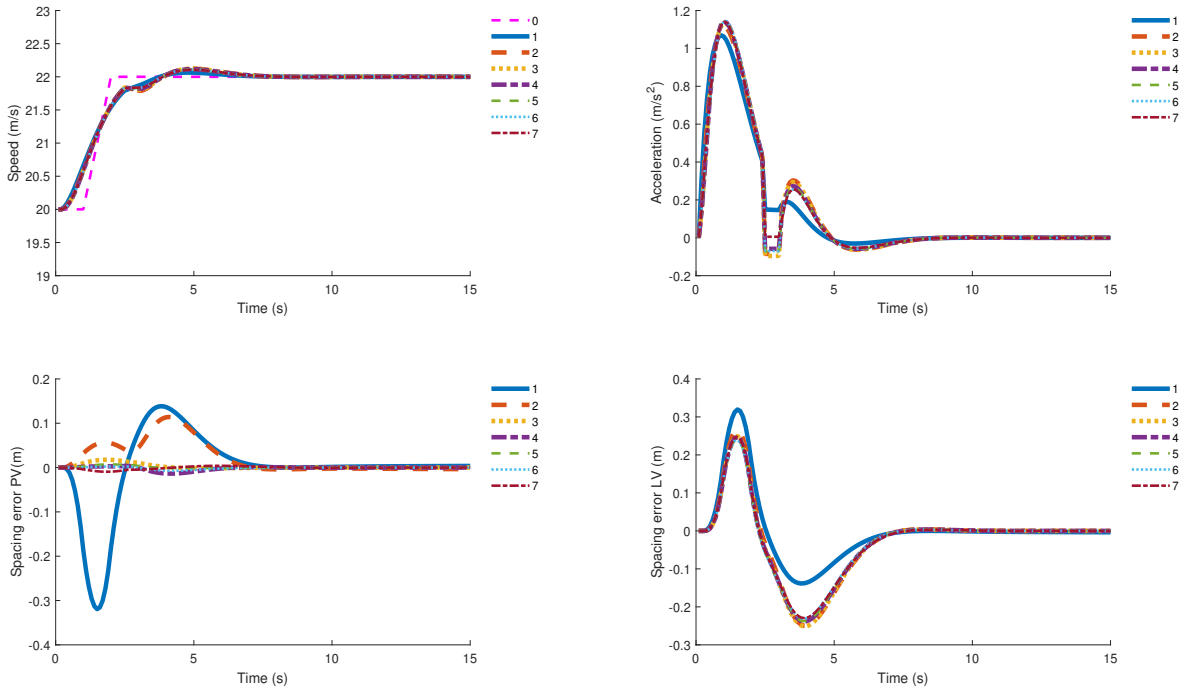

Figure 5.11. Rapid Acceleration under TPLF with Communication Delay with $N_p = 30$

Figures 5.10 and 5.11 show that increasing the prediction horizon smoothened the response in both speed and acceleration plots. All the results look acceptable in terms of string stability and having no steady-state error however supplementary revisions on the cost function weights are possible to be evaluated considering driver comfort and fuel efficiency. Addition of another cost function component for the control signal rate can also be considered in order to overcome the reverse peak in the acceleration.

Since the DMPC algorithm for communication delay provided acceptable steady-state performance without exceeding the control signal limitations and the string stability is achieved under these conditions, no further modification is aimed within the frame of this chapter.

5.5. Communication Topology with Information Packet Loss

In addition to the investigation of delayed data in wireless communication, the information packet loss issue is examined, as well. Once the packet loss is detected in the follower nodes via additional software, the topology is updated to its no-leader version, for example, PLF to PF and TPLF to TPF. This change is examined for several points; before follower vehicle acceleration begins, during follower vehicle acceleration and during leader vehicle acceleration.

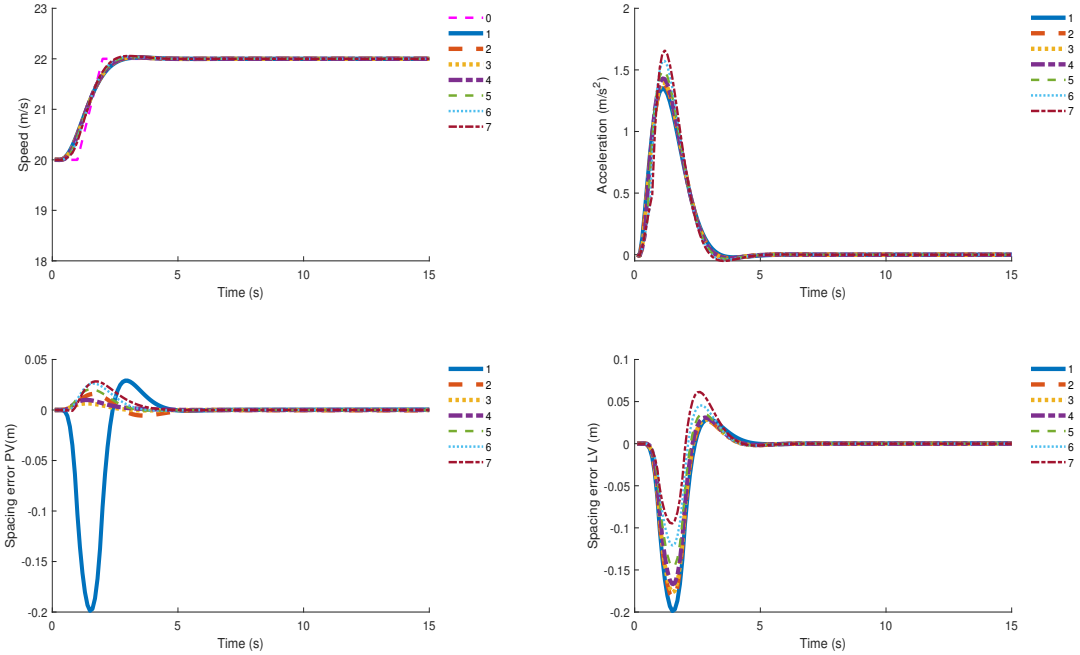

Figure 5.12. Topology Switch from PLF to PF at t=0.3s

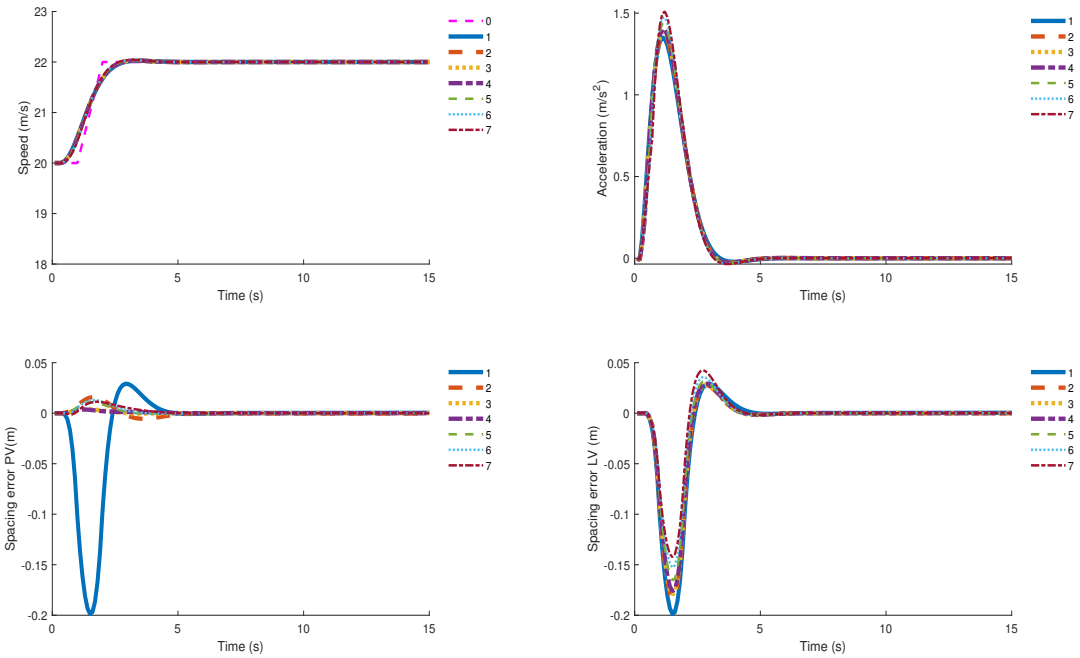

Figure 5.13. Topology Switch from PLF to PF at t=0.4s

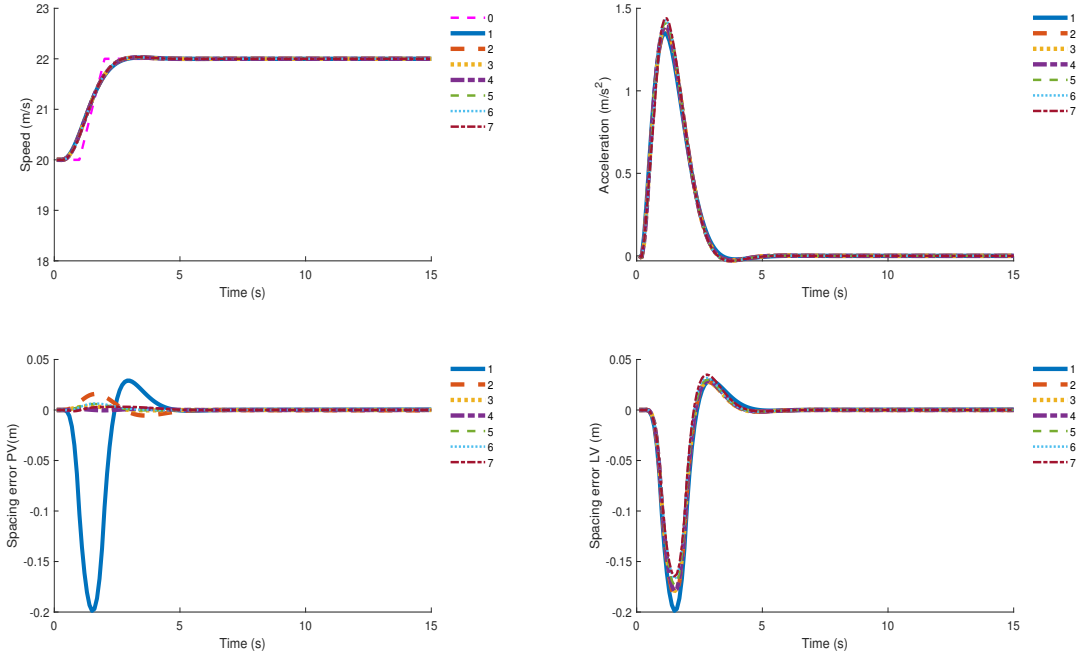

Figure 5.14. Topology Switch from PLF to PF at t=0.5s

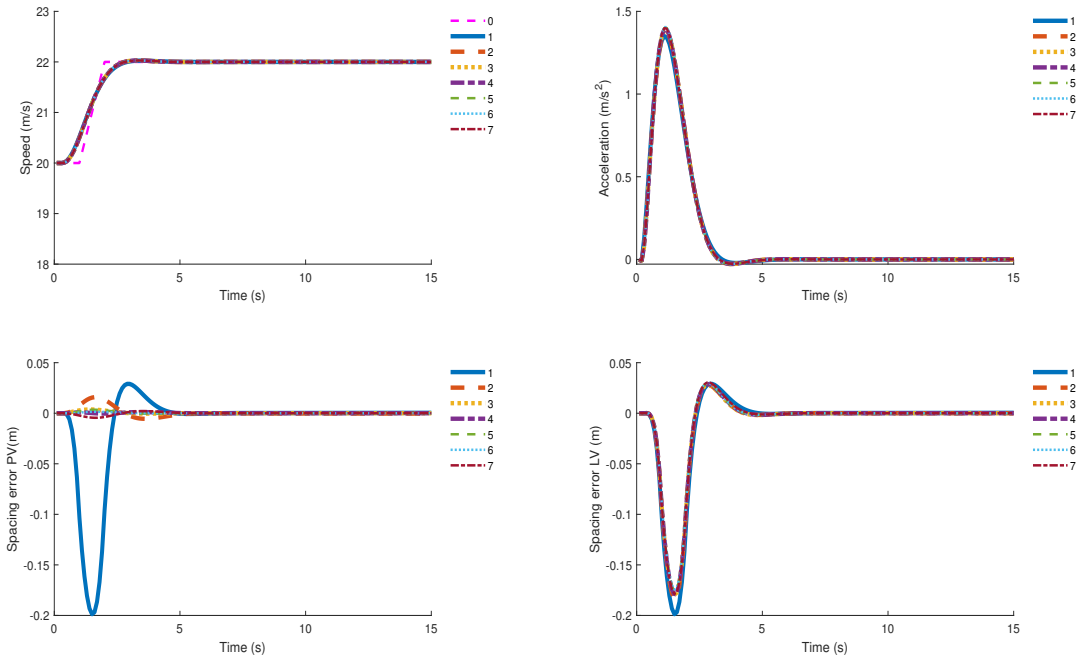

Figure 5.15. Topology Switch from PLF to PF at t=1s

By examining Figures 5.12 - 5.15, it can be claimed that if the packet loss happens before the following vehicle acceleration, then the platoon response is similar to PF response. If the packet loss occurs after the acceleration starts, then the magnitude of the spacing error tends to decrease and results become closer to PLF response.

Effects caused by the TPLF to TPF transition are also investigated considering the same conditions applied while working on the above-mentioned PLF to PF topology change.

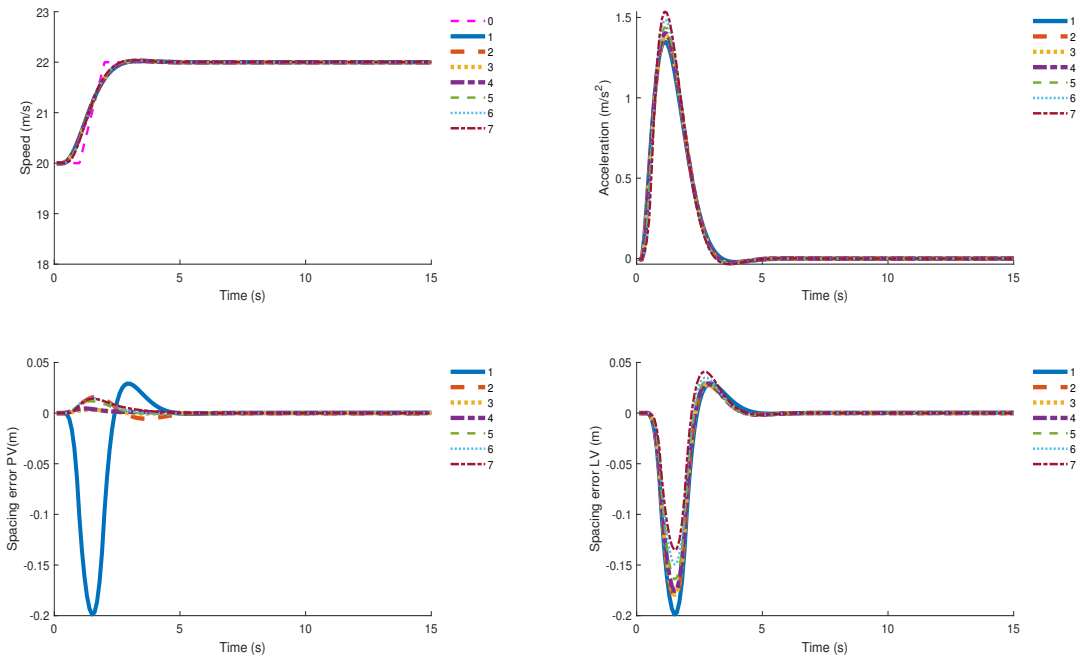

Figure 5.16. Topology Switch from TPLF to TPF at t=0.3s

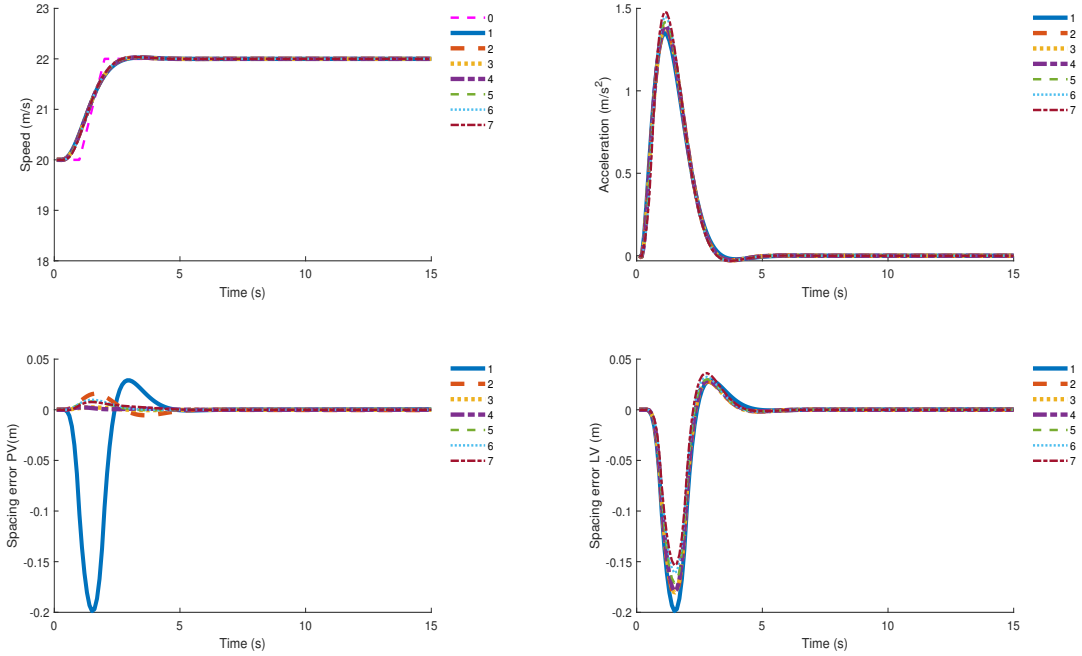

Figure 5.17. Topology Switch from TPLF to TPF at t=0.4s

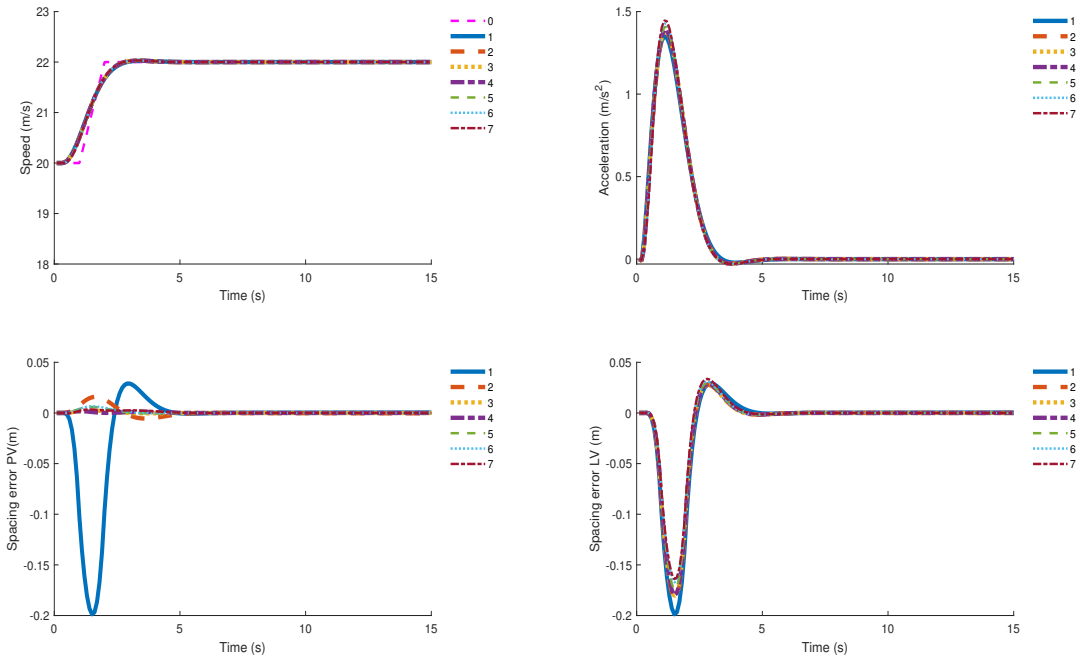

Figure 5.18. Topology Switch from TPLF to TPF at t=0.5s

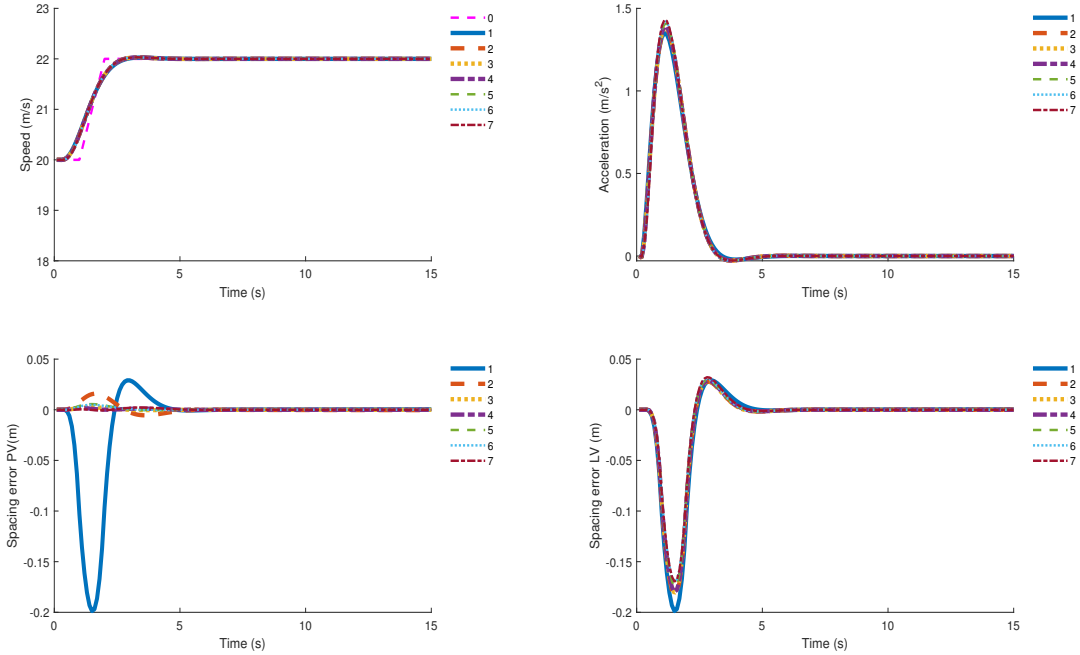

Figure 5.19. Topology Switch from TPLF to TPF at t=1s

Figures 5.16 - 5.19 show that the results are consistent with PLF case. When the results are compared with Figures 3.4 and 3.5 in Chapter 3, it can be said that if the packet loss happens before the acceleration of the follower vehicles, the response looks similar to TPF plots. If the packet loss occurs after the acceleration begins, then spacing error and acceleration plots are closer to the results of TPLF topology.

As a last package loss scenario application; the existence of the leader information is allowed and interrupted continually by manipulating the topology switching in order to simulate the weaknesses in communication quality.

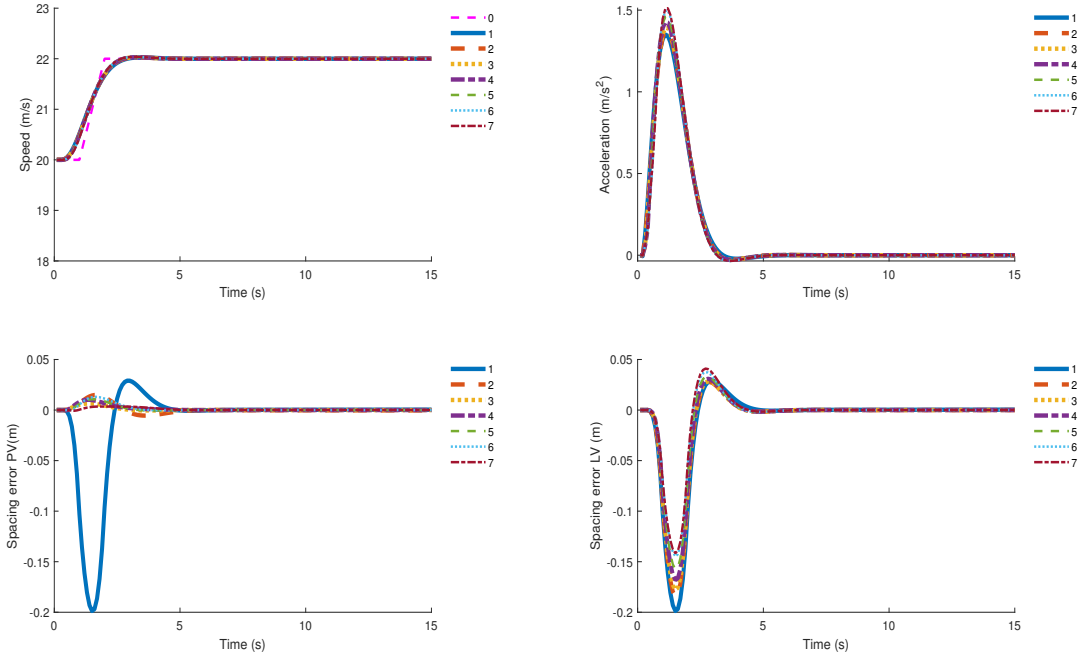

Figure 5.20. Continual Topology Switching between PLF and PF

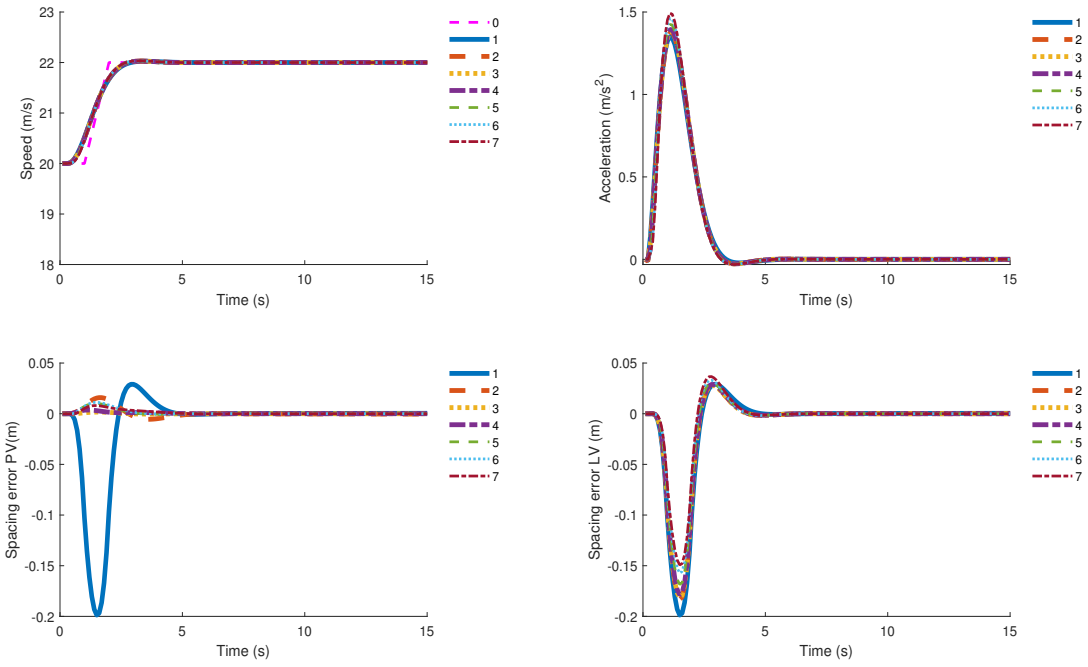

Figure 5.21. Continual Topology Switching between TPLF to TPF

Under the weak communication scenario where topology switching between PF and PLF (or; TPF and TPLF) occurs at each time step, the results are not string stable. However, the spacing errors become smaller than the results observed in Chapter 3 where the leader information is not available, namely PF and TPF. It can be claimed that DMPC is able to cope with short communication dropouts.

5.6. Summary of the Chapter

In this chapter, the DMPC communication delay handling capabilities are investigated under unidirectional PLF and TPLF topologies. The main advantage of the MPC controller is that it already includes the future states in its structure. In order to extract that information from the DMPC controller, a complementary approach to the algorithm proposed in Chapter 3 is presented. It is proven that the proposed approach is capable of removing the limitation on the distributed controllers caused by the pre-known uniform delayed leader information. Satisfactory results are achieved in terms of setpoint tracking and string stability under both PLF and TPLF configurations.

However, static and previously known communication delay has a very limited field of application in real life. The amount of delay arising out of hardware/software failures or by road, driving or weather conditions cannot always be precisely predicted. This algorithm is likely to fail in applications with random delay. Therefore, more research is needed to handle random delay which is beyond the focus of this thesis.

Furthermore, the effects of the information packet loss in wireless communication are examined under unidirectional PLF and TPLF topologies. It is shown that the moment that packet loss occurs has a direct effect on the platoon response. The simulation results are very close to PF/TPF topology results if the packet loss occurs before the acceleration starts, because the leader trajectory information is never transmitted to the follower vehicles. When the packet loss is shifted to a later time in the simulation, the results get closer to ones that are obtained under PLF/TPLF topologies.

6. CONCLUSION

Vehicle platoon control is an active control problem since achieving string stability with bidirectional topology is still a challenging task when the model contains nonlinear characteristics. Other unresolved problems encountered in real life applications are the communication delay and communication dropout that might be caused by the road curvature, the platoon length or the technical failure, etc.

The main contribution of this thesis has been to develop a DMPC algorithm for a nonlinear vehicular platoon under bidirectional topologies. The performance of the DMPC algorithm for the bidirectional links was compared with the results of the unidirectional topologies and it has been demonstrated via simulations that DMPC can process the additional information without violating string stability. Cost function components and their weighting factor kept the platoon moving in the desired trajectory while the constraint formulation prevented undesired actions such as front-end or rear-end collision. Including follower vehicle states to the optimization problem helped to generate a safer environment since the ego vehicle has the information of its closest neighbor vehicles in each direction to avoid the possible crashes that may occur under extreme conditions.

Another contribution of this thesis has been to investigate the effects of delay on the DMPC and the overall platoon and propose a method to overcome these adverse consequences. The delay was applied to the leader vehicle transmitted states with an increasing amount towards the end of the platoon. It is shown that the original DMPC algorithm proposed in Chapter 3 is not able to cope with the delay effects. The delayed information coming from the leader and the proper states received from the neighbor vehicle/s have easily driven DMPC to a region where the problem becomes infeasible. To resolve this, the valid data that already exists in the prediction horizon was extracted and used for the problem solution. Trajectory matrix that contains the future position and velocity states of the leader vehicle was processed to remove already past data points and generate new ones in order to match the matrix size. Steady-state

error caused by the communication delay has been fixed by this method.

Wireless communication dropout case is also simulated in order to examine its effects on the platoon. Communication topology was changed from PLF and TPLF to PF and TPF, respectively and the leader vehicle desired trajectory was removed from the optimization problem of the follower vehicles. It is observed that the communication dropout time also affects the response of the platoon and if it occurs before acceleration starts the plots are similar to PF. For later time points, the error margins are getting closer to the plots that are obtained with PLF topology.

As a conclusion, two major obstacles in the vehicle platooning were investigated in this study. The proposed algorithms have been implemented and tested in the simulation environment. The proposed DMPC algorithms are fast enough to be used in real time applications and provide satisfactory performance with zero steady-state error and small overshoots. The controller performances can be enhanced with additional calibration effort on the cost function weights, but the main focus of this study has been to construct the DMPC components in terms of costs and constraints to resolve the identified issues. It should be noted that this study is limited to the simulation development and theoretical work is left open for future investigation. In order to conduct such a practice, the previous study done by Zheng et al. [5] can be taken as reference.

Finally, in light of the work done so far, a few notes can be deducted to further improve this study. One suggestion is to develop an algorithm that can also reduce the effects of random delays. In this study, it was assumed that each vehicle knew its order in the platoon and as a result, knew the amount of delay in the information it receives. The modifications that could be made on the algorithm for random delay handling deserves further attention. Another improvement is the application of a dynamic communication topology that may depend on the acceleration trend in order to mimic the real-life driving behaviors. Under the assumption of driving in a single lane, while accelerating, the preceding vehicle states are more crucial than the follower vehicles. Similarly, for collision avoidance, follower vehicle states are as important as the

preceding vehicle states during deceleration. To better capture this tendency, a hybrid approach can be obtained to switch communication topology from unidirectional to bidirectional while decelerating.

REFERENCES

1. Alam, A., B. Besselink, V. Turri, J. Martensson and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency", *IEEE Control Systems*, Vol. 35, No. 6, pp. 34–56, 2015.

2. Swaroop, D. and S. Min, "The Design of a Controller for a Following Vehicle in an Emergency Lane Change Maneuver", UC Berkeley: California Partners for Advanced Transportation Technology, 1999.

3. McMahon, D. H., J. K. Hedrick and S. E. Shladover, "Vehicle Modeling And Control For Automated Highway Systems", UC Berkeley: California Partners for Advanced Transportation Technology, 1993.

4. Zegers, J. C., E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw and H. Nijmeijer, "Consensus Control for Vehicular Platooning With Velocity Constraints", *IEEE Transactions on Control Systems Technology*, Vol. 26, pp. 1592 – 1605, 2018.

5. Zheng, Y., S. E. Li, K. Li, F. Borrelli and J. K. Hedrick, "Distributed Model Predictive Control for Heterogeneous Vehicle Platoons Under Unidirectional Topologies", *IEEE Trans. Contr. Sys. Techn.*, Vol. 25, No. 3, pp. 899–910, 2017.

6. Yan, M., J. Song, P. Yang and L. Zuo, "Neural Adaptive Sliding-Mode Control of a Bidirectional Vehicle Platoon with Velocity Constraints and Input Saturation", *Complexity*, Vol. 2018, pp. 1–11, 2018.

7. Tom, R. and C. Erik., *Operating Platoons On Public Motorways: An Introduction To The SARTRE Platooning Programme*, 2013, `https://www.researchgate.net/publication/268300380_Operating_Platoons_On_Public_Motorways_An_Introduction_To_The_SARTRE_Platooning_Programme`, accessed at March 2018.

8. "Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey", NHTSA's National Center for Statistics and Analysis, 2015.

9. Nowakowski, C., S. E. Shladover, X.-Y. Lu, D. Thompson and A. Kailas, *Cooperative Adaptive Cruise Control (CACC) for Truck Platooning: Operational Concept Alternatives*, 2015, `https://escholarship.org/uc/item/7jf9n5wm`, accessed at April 2018.

10. Bemporad, A., "Model-based predictive control design: New trends and tools", *Proc. 45th IEEE Conf. Decision and Control*, pp. 6678–6683, 2006.

11. Li, S. E., K. Li, R. Rajamani and J. Wang, "Model Predictive Multi-Objective Vehicular Adaptive Cruise Control", *IEEE Trans. Contr. Sys. Techn.*, Vol. 19, No. 3, pp. 556–566, 2011.

12. Seiler, P., A. Pant and J. K. Hedrick, "Disturbance propagation in vehicle strings", *IEEE Trans. Automat. Contr.*, Vol. 49, No. 10, pp. 1835–1842, 2004.

13. Swaroop, D., J. Hedrick, C. C. Chien and P. Ioannou, "A Comparision of Spacing and Headway Control Laws for Automatically Controlled Vehicles", *Vehicle System Dynamics*, Vol. 23, pp. 597–625, 1994.

14. Darbha, S. and J. K. Hedrick, "Constant spacing strategies for platooning in automated highway systems", *Trans. ASME, J. Dyn. Syst. Meas. Control*, Vol. 121, No. 3, pp. 462–470, 1999.

15. Zegers, J. C., E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw and H. Nijmeijer, "Consensus-based bi-directional CACC for vehicular platooning", *American Control Conference (ACC 2016)*, pp. 2578–2584, 2016.

16. Zheng, Y., S. E. Li, J. Wang, D. Cao and K. Li, "Stability and Scalability of Homogeneous Vehicular Platoon: Study on the Influence of Information Flow Topologies", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 1,

pp. 14–26, 2016.

17. Gao, F., S. E. Li, Y. Zheng and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay", *Intelligent Transport Systems (IET)*, Vol. 10, p. 503–513, 2016.

18. Xiangheng, L., A. Goldsmith, S. Mahal and J. Hedrick, "Effects of communication delay on string stability in vehicle platoons", *Intelligent Transportation Systems 2001. Proceedings. 2001 IEEE*, pp. 625–630, 2001.

19. El-Ferik, S., B. A. Siddiqui and F. L. Lewis, "Distributed Nonlinear MPC of Multi-Agent Systems With Data Compression and Random Delays", *IEEE Transactions on Automatic Control*, Vol. 61, No. 3, pp. 817–822, 2016.

20. Erkan, O. F. and M. Akar, "Distributed consensus with multiple equilibria in continuous-time multi-agent networks under undirected topologies", *IFAC-PapersOnLine*, Vol. 48, 2015.

21. Sipser, M., *Introduction to the Theory of Computation*, Massachusetts Institute of Technology, 2006.

22. Seiler, P. J., "Coordinated Control of Unmanned Aerial Vehicles", University of California, 2001.

23. Gao, F., S. E. Li, Y. Zheng and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay", *IET Intelligent Transport Systems*, Vol. 10, No. 7, pp. 503 – 513, 2016.

24. Maciejowski, J., *Predictive Control with Constraints*, Prentice Hall, 2003.