Optimal Control and Resource Allocation over Wireless Networks with Applications in Automotive Systems

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)
genehmigte

Dissertation

von
Herrn
M.Sc. Shaban Masaud Sifau Guma
geb. in Tanzegat

Tag der mündlichen Prüfung: 2. Mai 2023

Prüfungskommission:
Prof. Dr.-Ing. Sergiy Antonyuk (Vorsitzender)
Prof. Dr.-Ing. Naim Bajcinca (1. Berichterstatter)
Prof. Dr.-Ing. Daniel Görges (2. Berichterstatter)

Kaiserslautern, 2023
D 386

"In order to have a friend, you must be a friend",

- Elbert Hubbard.

Abstract

In the context of distributed networked control systems, many issues affect the performance and functionality of the connected subsystems, mainly raised because of the communication medium imposed into the system structure. The communication functionality must generally cope with the data exchange requirements between system entities. Therefore, due to the limited communication resources, especially in wireless networks, an optimal algorithm for the assignment of the communication resources and proper selection of the right Medium Access Control (MAC) protocol are highly needed.

In this dissertation, we studied several problems raised by communication networks in wireless networked control systems, with a particular focus on the effect of standard Medium Access Control (MAC) protocols on the overall control system performance. We examined the effect of both the Time Division Multiple Access (TDMA) and the Orthogonal Frequency Division Multiple Access (OFDMA) protocols and developed a set of distributed algorithms that suit their specification requirements.

As a benchmark, we used a vehicle dynamics optimal control problem where the objective of the optimization problem is to penalize the maximal utilization of the tire's adhesion forces for a given driving maneuver. The problem was decomposed into a distributed form using primal and dual decomposition techniques, and solving algorithms were derived using both primal and dual subgradient methods. The problem solver was tested with respect to a wireless networked system structure and evaluated for different communication typologies, such as uni-directional, bidirectional, and broadcasting topology.

Later, the setup of the solution algorithms was extended concerning the specification of the TDMA and OFDMA protocols, and we introduced an event-triggered scheme into the solver algorithm. The proposed event-triggered scheme is mainly utilized to reduce communication between concurrent computation subsystems, which is primarily intended to facilitate real-time efficiency. Next, we investigated the effect of the data exchange between subsystems on the overall solver performance and adapted the sensitivity analysis concept within the event-based communication scheme. An adaptive sensitivity-based TDMA algorithm was developed to manage the extensive communication resource requests, and channel utilization was adapted for the optimal solution behavior.

In the last part of the thesis, we extended our research direction to the multi-vehicle concept and investigated the communication resource allocation problem in the context of the OFDMA protocol. We developed an adaptive sensitivity-based OFDMA protocol based on linking the evolution of the application layer to the communication layer and assigning the communication resources concerning the sensitivity analysis of the optimization problem at the application layer.

Zusammenfassung

Im Kontext von verteilten vernetzten Steuerungssystemen gibt es viele Probleme, die die Leistung und Funktionalität solcher Systeme beeinflussen, welche hauptsächlich durch die Einführung des Kommunikationsnetzwerks in die Systemstruktur entstehen. Im Allgemeinen muss die Kommunikationsfunktionalität den Anforderungen an den Datenaustausch zwischen den Systementitäten gerecht werden. Aufgrund der begrenzten Kommunikationsressourcen, insbesondere in drahtlosen Netzwerken, sind ein optimaler Algorithmus für die Zuweisung der Kommunikationsressourcen und die passende Auswahl des richtigen Medium Access Control (MAC) -Protokolls dringend erforderlich.

Im Rahmen dieser Dissertation wurden mehrere Probleme untersucht, die durch Kommunikationsnetzwerke im Kontext von drahtlosen vernetzten Steuerungssystemen aufgeworfen werden, mit besonderem Fokus auf den Einfluss von Standard-Multiple-MAC-Protokollen auf die Gesamtleistung des Steuerungssystems. Wir haben den sowohl den Einfluss des Time Division Multiple Access (TDMA)- als auch des Orthogonal Frequency Division Multiple Access (OFDMA)-Protokolls untersucht und eine Reihe von verteilten Algorithmen entwickelt, die ihren Spezifikationen entsprechen.

Als Benchmark wurde ein Fahrdynamik-Optimierungsproblem herangezogen, bei dem das Ziel des Optimierungsproblems darin besteht, die maximale Nutzung der Reifenhaftkräfte für ein bestimmtes Fahrmanöver zu bestrafen. Das Problem wurde mit Hilfe von primären und dualen Zerlegungstechniken in eine verteilte Form zerlegt, und es wurden Lösungsalgorithmen abgeleitet, die sowohl primäre als auch duale Subgradientenmethoden verwenden. Der Lösungsansatz wurde auf eine drahtlos vernetzte Systemstruktur verteilt und für verschiedene Kommunikationstypologien wie unidirektionale, bidirektionale und Broadcasting-Topologie getestet und bewertet.

Danach wurde der Aufbau des Lösungsansatzes im Hinblick auf die Spezifikation der TDMA- und OFDMA-Protokolle durch die Einführung eines ereignisgesteuertes Schemas erweitert. Das vorgeschlagene ereignisgesteuerte Schema wird hauptsächlich dazu verwendet, die Kommunikation zwischen gleichzeitig rechnenden Knoten zu reduzieren, was in erster Linie die Echtzeiteffizienz verbessern soll. Als Nächstes untersuchten wir die Auswirkungen des Datenaustauschs zwischen den Teilsystemen auf die Gesamtleistung des Solvers und passten das Konzept der Empfindlichkeitsanalyse innerhalb des ereignisbasierten Kommunikationsschemas an. Es wurde ein adaptiver empfindlichkeitsbasierter TDMA-Algorithmus entwickelt, um die umfangreichen Anforderungen an die Kommunikationsressourcen zu verwalten, und die Kanalauslastung wurde an das optimale Lösungsverhalten angepasst.

Im letzten Teil dieser Arbeit haben wir unsere Forschungsrichtung auf das Multi-Fahrzeug-Konzept erweitert und das Problem der Kommunikationsressourcenallokation im Kontext des OFDMA-Protokolls untersucht. Wir haben ein adaptives Empfindlichkeitsbasiertes OFDMA-Protokoll entwickelt, das auf der Verknüpfung der Evolution

der Anwendungsschicht mit der Kommunikationsschicht und der Zuweisung der Kommunikationsressourcen in Bezug auf die Sensitivitätsanalyse des Optimierungsproblems auf der Anwendungsschicht basiert.

Acknowledgements

Some people make a difference and have a high impact on each period of our life. We highly acknowledge their presence and appreciate their support.
First, I would also like to thank Prof. Dr.-Ing. Naim Bajcinca for his major contribution to my research, guidance, and support, both academically and personally. Second, I would like to thank Prof. Dr.-Ing. Jörg Raisch for accommodating me in the first period of my research career in his group. Third, I would like to express my deep appreciation to all members of my family, my wife, my kids Mohamed, Salma, and Abdualhai, my brothers, my sisters, and not forgetting my nephew (my youngest friend) Abdu Alrhman. Finally, my Friends, I will not write any names because I thank all of you for your honest friendship, and for enjoying my espresso with you.

Publications

[1] Y. Kouhi, S. Guma, and N. Bajcinca. Real-time allocation of tire adhesion forces for electric vehicles. In Control Applications (CCA), 2012 IEEE International Conference on, pages. 172-177, Oct 2012.

[2] S. Guma and N. Bajcinca. Distributed feedforward control of vehicle dynamics based on event-triggered optimization. In 2013 XXIV International Conference on Information, Communication and Automation Technologies, pages. 1-6, Oct 2013.

[3] N. Bajcinca, G. Dartmann, K. Sun, V. Handzinski, S. Guma, and G. Ascheid. Towards a novel drive-by-wireless vehicular technology. In VDI 2015, Jun 2015.

[4] S. Guma and N. Bajcinca. Event-triggered adaptive OFDMA protocol for multi-vehicle clusters. In 2018 European Control Conference (ECC), pages. 1779–1784, Jun 2018.

[5] S. Guma and N. Bajcinca. Sensitivity based event-triggered TDMA protocol for distributed optimization. In 2018 26th Mediterranean Conference on Control and Automation (MED), pages. 473–480, Jun 2018.

Contents

List of Figures

List of Tables

Abbreviations

WLAN	**W**ireless **L**ocal **A**rea **N**etwork
NCS	**N**etworked **C**ontrol **S**ystem
WNCS	**W**ireless **N**etworked **C**ontrol **S**ystem
OFDMA	**O**rthogonal **F**requency **D**ivision **M**ultiple **A**ccess
QoS	**Q**uality **o**f **S**ervice
MAC	**M**edium **A**ccess **C**control
TDMA	**T**ime **D**ivision **M**ultiple **A**ccess
FDMA	**F**requency **D**ivision **M**ultiple **A**ccess
CDMA	**C**ode **D**ivision **M**ultiple **A**ccess
SDMA	**S**pace **D**ivision **M**ultiple **A**ccess
CSMA	**C**arrier **S**ence **M**ultiple **A**ccess
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
BER	**B**it **E**rror **R**ate
SNR	**S**ignal **t**o **N**oise **R**atio
SINR	**S**ignal **t**o **I**nterference plus **N**oise **R**atio
CG	**C**enter **o**f **G**ravity
FDD	**F**requency **D**ivision **D**uplex
TDD	**T**ime **D**ivision **D**uplex
AP	**A**ccess **P**oint
BS	**B**ase **S**tation
64QAM	**64 Q**uadrature **A**mplitude **M**odulation
16QAM	**16 Q**uadrature **A**mplitude **M**odulation
QPSK	**Q**uadrature **P**hase **S**hift **K**eying
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers

To my mother and my father, may Allah rest their souls in heaven

...

Chapter 1

Introduction

With the rapid improvements in communication technology, mobility, flexibility, bandwidth, coverage range, hardware, and the availability of wireless communication infrastructure, wireless communication has become the central element in most modern technology [3], including control systems engineering and its applications such as Wireless Networked Control System (WNCS) [4, 5]. Basically, WNCS defines a class of systems where the system components (subsystems/-agents) are connected via wireless links, and the control input signals and sensor measurements are transmitted through the wireless communication medium [6]. Using the wireless channel as the communication medium in the control loop has the advantages of mobility, spatially distribution, connection flexibility, ease of installation, and reduces cabling costs [7–9]. On the other hand, the WNCS inherits wireless communication problems such as delay and jitters, bandwidth limitation, [9–11] time delay [12], packet loss [10, 12], channel access and limited wireless communication resources [13, 14], data rate [15, 16], and energy consumption constraints [17, 18].

Basically, the fundamental question in wireless networked control systems is related to the trade-off between the wireless communication characteristics in terms of available wireless communication resources and the control system requirements that stabilize the system and guarantee its performance. The performance of a wireless networked control system heavily depends on the packet delivery between the control system entities, which is limited by the quantity and quality of the data exchanged over the wireless communication channel, and also depends on the availability of the communication resources. Therefore, wireless communication constraints such as limited data rate, short bandwidth, availability, and the Quality of wireless Service (QoS) have a big impact on the data exchange between the system entities [19].

In view of the limitations of commercial off-the-shelf wireless communication technologies, an adaption of the design methods of control and communication algorithms at the protocol level is needed. It should be noted, though, that the added value of the distributed WNCS requires providing a reliable, secured wireless communication channel with a sufficient data rate and bandwidth to cope with the control system requirements [20]. We thus formulate the first research question of this work as follows:

For a Distributed Optimal Wireless Networked Control System consisting of a set of distributed wireless nodes solving an optimization problem and communicating over wireless channel, what is the effect of the limited communication resources on the system performance? And how is the convergence rate of the optimal control problem solution affected by the limited wireless communication resources?

Wireless communication systems face issues with limited resources, which can affect the Medium Access Control (MAC) protocol and cause disruptions in the exchange of data packets. This can lead to shortages in packet delivery and delays within the control system loop. Under these circumstances and also due to the increased demands of the communication resources within a distributed wireless network, it is necessary to adapt the data exchange mechanisms within the networked control system loop in order to assign the communication resources to a subsystem that has a higher effect on the system performance. Therefore, we formulate the second research question as follows:

For a Distributed Optimal Wireless Networked Control System, how to regulate the communication sequence for a distributed optimization problem solved over a wireless network in order to converge to the optimal solution while reducing the communication demands within the wireless network at the same time?

The concept of solving the distributed optimization problem within the WNCS requires extensive collaboration between the distributed subsystems. Collaboration means the systematic exchange of information between the subsystems over the wireless channel. Therefore, the convergence of the distributed optimization problem to an optimal solution depends on the reliability of the communication and on the packet delivery ratio[21]. All methods used to solve the distributed optimization problem are based on iterative algorithms, which require extensive iterative computation, and on heavy communication between the system's entities. In order to circumvent the extensive communication requirement in solving the distributed optimization problem, an event-triggered distributed optimization algorithm has been developed that guarantees a given convergence rate of the distributed optimization problem with respect to the system performance. The implementation of this distributed event-triggered optimization algorithm requires each subsystem to independently reduce the communication resource requirements based on its internal state evolution. The results show that this is an efficient way for limiting the usage of communication resources and guarantees the message flow among the subsystem, which preserves the performance, stability, and convergence of the overall system.

In particular, we investigate how to share the wireless channel resources between interconnected agents in such a way that the convergence rate of the optimization problem reaches an acceptable value. The main focus of this thesis is on developing a set of event-based communication algorithms in conjunction with solution algorithms of the distributed optimization problem under the communication constraints and comply with the standard MAC protocols specification. Finally, the vehicle dynamics problem will be used as a benchmark for our simulation study and proof of concept of the developed algorithms. This thesis is organized as follows:

Chapter 2 introduces the theoretical background of wireless communication considered in this work, followed by an introduction to the concept of wireless networked control systems, and provides a detailed model of vehicle dynamics, which is used as a benchmark for the proof of concept simulation.

Chapter 3 presents the theoretical background of the optimal control problem, the convex optimization problem, and various decomposition methods, and defines the framework of distributed optimal control over a wireless network. It also introduces the solver algorithm with a focus on the subgradient method and average consensus. At the end, the vehicle dynamics optimization problem is formulated and the optimization problem solver is presented, named "Consensus-based projected primal subgradient algorithm".

Chapter 4 introduces the event-triggered concept that is implemented in the distributed primal subgradient algorithm, and presents the adaptation of the solution algorithm for a TDMA-MAC-based wireless network. Also, the algorithm implementation is reformulated with respect to the concept of distributed event-based communication.

Chapter 5 focuses on the dual subgradient algorithm of the optimal control problem, and introduces the sensitivity concept to the event-triggered scheme, with the sensitivity of the data exchange being utilized in the TDMA scheduler to develop the sensitivity-based adaptive TDMA protocol.

Chapter 6 focuses on the allocation of the communication resources in the OFDMA protocol, and uses the sensitivity of exchanging the state information on the optimal solution of the problem. It mainly expands the system structure setup to a distributed multi-vehicle system based on an OFDMA-based wireless network, and investigates the connection between the application layer and the communication layer. It also presents a sensitivity-based resource allocation algorithm for the OFDMA scheduler.

Chapter 7 presents the conclusion of this work, lists unresolved issues, and provides some directions for future study.

Chapter 2

Theoretical background

This chapter reviews wireless communication terminology related to wireless networked control systems and provides the theoretical background for the issues addressed in this dissertation. Section 2.1 introduces the wireless network standards, and the set of standard medium access control protocols used in wireless networks. Section 2.2 introduces wireless networked control systems and lists theoretical works developed in this area, and also describes the effect of using a wireless channel as a communication medium in the control loop. The third section 2.3 presents vehicle dynamics and planar motion, which will be used as a benchmark for the simulation study.

2.1 Wireless communication standards

Wireless communication has become common technology in most modern applications, and plays a key role especially for such applications that require data exchange without cables or cords, such as remote sensing, cell phones, computer networks, and wireless sensor networks. The basic technical specifications of wireless communication mainly depend on the frequency band and the standard specification of the wireless communication layers. The wireless frequency band defines the frequency range that wireless devices operate on, and the wireless standards define the set of characteristics of the physical layer (PHY) and the Medium Access (MAC) layer of the wireless technology [1]. In order to standardize wireless communication usage, reduce frequency overlapping, define wireless communication standards and protocols for general use and industrial technology, special organizations like the Internet Engineering Task Force (IETF), the Institute of Electrical and Electronics Engineers (IEEE), the Wireless Fidelity Alliance (Wi-Fi), and the ZigBee Alliance for low-power WLAN networks have defined the standards used for software and hardware comparability, as well as the operating frequency range of each standard. The defined standards and protocols are designed to ensure interoperability between communications' equipment from different vendors by allocating different parts of the frequency band to specific systems and applications. Here we will summarize the specifications of the Wireless Local Area Network (WLAN) standard (IEEE 802.11), the Wireless Personal Area Networks (WPAN) standard (IEEE 802.14), and the World-wide Interoperability for Microwave Access (WiMax) standard (IEEE 802.16), which are considered as the wireless communication platform in this study.

1. **IEEE 802.11 (WLAN)**

   The IEEE 802.11 standard defines the characteristics of a Wireless Local Area Network (WLAN). It defines the implementation specifications of the physical layer and media access control (MAC) for local area wireless network products. There are different generations of the IEEE 802.11 standard, which have been developed over time since 1997 when the first version was launched. IEEE 802.11 applies to 2.4 GHz with data rates starting from 1.2 Mbps [1, 3, 22]. Table 2.1 presents the key characteristics of the IEEE 802.11 standards.

	802.11	802.11a	802.11b	802.11g
Bandwidth (MHz)	300	83.5	83.5	83.5
Frequency range (GHz)	2.4-2.4835	5.15-5.825	2.4-2.4835	2.4-2.4835
Number of channels	3	12	3	3
Max data rate (Mbps)	1.2	54	11	54

TABLE 2.1: General specifications of the IEEE 802.11 (WLAN) standards -from [1].

2. **IEEE 802.15 (WPAN)**

   The low-rate Wireless Personal Area Network (WPAN) standard, IEEE 802.15, defines the fundamental characteristics of this low-cost, low-speed, low-power, and short-distance communication technology. Based on the application requirements such as wireless sensors and short-range data transmission, there are many standards that have been generated from IEEE 802.15, such as Bluetooth IEEE 802.15.1, ultra-band IEEE 802.15.3, and Zigbee IEEE 802.15.4 [3]. Table 2.2 presents the key characteristics of these standards derived from IEEE 802.15 [1].

	Zigbee (802.15.4)	Bluetooth (802.15.1)	UWB(802.15.3)
Frequency range (GHz)	2.4-2.4835	2.4-2.4835	3.1-10.6
Bandwidth (MHz)	83.5	83.5	7500
Max data rate (Mbps)	0.25	1	100
Range (m)	30	10	10
Channel access method	CSMA/CA (TDMA)	TDMA	Undefined
Power consumption (mW)	5-20	40-100	80-150
Networking	Mesh/Star/Tree	Sub-net Clusters(8 nodes)	Undefined

TABLE 2.2: General specifications of the 802.15 (WPAN) standards - from [1].

3. **IEEE 802.16 (WiMax)**

   The use of a multiple access scheme in order to serve multiple users has been integrated into many wireless communication applications in the last decade. In 2004, the Worldwide Interoperability for Microwave Access (WiMax) standard

IEEE 802.16 was introduced as an improvement over IEEE 802.11. The concept of multiple access control uses techniques such as Orthogonal Frequency Division Multiple Access (OFDMA), which operates based on Orthogonal Frequency Division Multiplexing (OFDM). IEEE 802.16 applies to 3.5-5.8 GHz for fixed WiMax, and 2.3-3.5 GHz for mobile WiMax. Table 2.3 summarizes the main specifications of the IEEE 802.16 (WiMax) standards [2].

	Fixed WiMax	Mobile WiMax
IEEE standards	802.16-2004	802.16e-2005
Frequency range (GHz)	3.5-5.8	2.3-3.5
Channel band width (MHz)	3.5, 7 in 3.5 GHz; 10 in 5.8 GHz	3.5, 7, 5, 10, and 8.75
Transmission subcarriers scheme	256 or 2048	128, 256, 512, 1024, or 2048
Max data rate (Mbps)	1-75	1-75
Medium Access Control (MAC)	OFDMA	OFDMA

TABLE 2.3: The general specification of IEEE 802.16 (WiMax) standards - from [2].

2.1.1 Medium Access Control (MAC) protocol

In general, a wireless communication network operates as a shared medium system, where a set of connected nodes use the same frequency band to transmit their data. The increasing number of interconnected nodes that communicate over a wireless channel introduces the problems of limited communication resources and high demands on wireless channel utilization. There are many standard multiple Medium Access Control protocols (MAC), such as Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Orthogonal Frequency-Division Multiple Access (OFDMA), Carrier Sense Multiple Access (CSMA), and Space Division Multiple Access (SDMA) [1, 3, 23–25]. In the following, we will describe the most frequently implemented MAC protocols in wireless communication systems:

2.1.1.1 Frequency Division Medium Access (FDMA) protocol

In the Frequency Division Medium Access (FDMA) protocol, the entire frequency bandwidth of the wireless channel is divided into non-overlapping sub-frequency bands called sub-channels. Each user (node/-agent) is assigned one sub-channel for its transmission process. For example, when a channel operates on a frequency band with a bandwidth of B GHz and serves N nodes, the total channel frequency band is divided into N sub-frequencies, each of which is $b_i = \frac{B}{N}$ GHz, and each node i is assigned a sub-frequency b_i GHz. To prevent sub-channels from overlapping, a small part of the frequency band called (guard band Guard band) is usually not assigned between each two successive sub-channels to both users [26]. For example, Fig. 2.1 shows that the frequency band is divided into 7 sub channel, and the guard bands separate each two successive sub-channels to not overlap and transmit in a close frequency band. Hence, the allocation of the sub-frequency is continuous over time, and the node uses the specified sub-frequency during the entire time of the transmission process. The relocation of the sub-frequency to another node is mainly difficult; it requires frequency-agile radios and the transceiver must be able to deal with different sorts of modulation methods [1, 23].

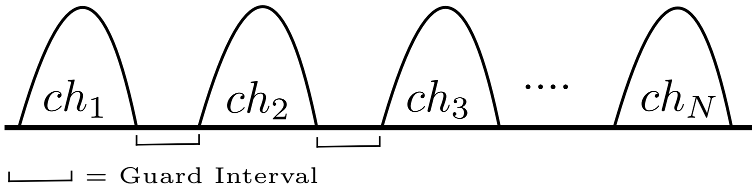

FIGURE 2.1: FDMA protocol (overlapping guard): The frequency spectrum is divided into N sub-frequencies and an overlapping guard was unsigned between each two successive users. Here ch_1, ch_2, and ch_N indicates the sub-frequency

2.1.1.2 Time Division Medium Access (TDMA)

The basic concept of the Time Division Medium Access protocol is that the channel utilization time is divided into fixed time frames, each consists of number of time slots. Where each interconnected node is assigned a time slot within each frame periodically, and it will be directly allocated the entire frequency band in order to transmit its information with a full data rate. This allows the node to exploit the frequency diversity available within the bandwidth allocated to the channel. Furthermore, the sensitivity to random frequency modulation is reduced [3].

Since we are considering the TDMA protocol for further investigation, we integrate the TDMA scheduler into the wireless network model introduced in [8], where the channel state is modelled as a switch S_i that opens and closes with respect to a certain rate. The data is transmitted if the channel switch is closed $S_i = 1$ and not delivered if the switch is open $S_i = 0$. Here, the TDMA protocol assigns the channel utilization for a fixed time period to one node by closing the channel switch $S_i = 1$ and allowing its transmitted data to pass over the channel frequency to the other nodes. Therefore, TDMA protocol model consists of a set $\mathcal{N} = \{i = 1, \ldots, N\}$ nodes connected to a TDMA-based wireless network, $\mathcal{N}_i$ defines the set of node i neighbors, N is the number of connected nodes, and x_i presents the information transmitted by node i to its neighbor j, and $j \in \mathcal{N}_i$. Basically, channel time is divided into frames of duration f. Each frame f is divided into N time slots of duration $\Delta_t = \frac{f}{N}$ length. Also, a guard time slot is inserted between two successive time slots to reduce signal interference or synchronization errors [1, 23]. The TDMA scheduler controls the time slots allocation to the interconnected nodes, and if the time slot is assigned to a node i, its data will be transmitted to its neighbors j;

otherwise, it will keep the last transmitted value as follows:

$$x_i^j[k+1] = \begin{cases} x_i^j[k+1], & if(S_i{=}1) \\ x_i^j[k], & if(S_i{=}0) \end{cases} \tag{2.1}$$

where, x_i^j is the information transmitted by i received at neighbor j. This implies that if node i is assigned a time slot $T_{s,i}$, the information $x_i^j[k+1]$ received at node j. Fig. 2.2 illustrates the TDMA protocol mechanism operating on a systematic transmission order $1, \ldots N$, where the TDMA scheduler assigns the time slots to the connected nodes in a fixed order. Within each time frame, the scheduler assigns the node i with the time slot

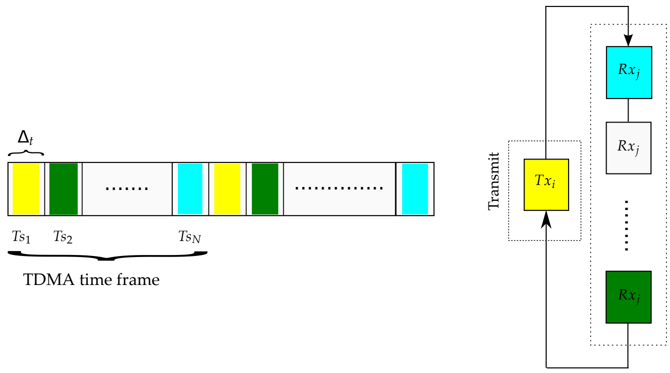

FIGURE 2.2: TDMA protocol: channel time division into N time slots T_s assigned to N nodes, and node switching between transmitting state Tx_i and receiving state Rx_i.

T_{si}, where it changes its state into transmission mode T_{xi}, and it starts transmission of its data x_i over the channel frequency band. In the meanwhile, the other nodes are in receiving mode R_{xi} and receive the data x_i^j that is transmitted by the node i. When the time slot T_{si} elapsed, the TDMA scheduler assigns the channel to the next node in the sequence until the frame ends, and the next frame will start in the same order.

2.1.1.3 Orthogonal Frequency Division Multiple Access (OFDMA)

Orthogonal Frequency Division Multiple Access (OFDMA) protocol is based on splitting the frequency bandwidth into a large number of closely orthogonal sub-carrier signals, each of which is assigned to a different user to carry on their parallel data transmission. Conceptually, the advantages of the orthogonality of sub-carrier frequency is to increase the usability of the available spectrum, increase the number of users, and enhance channel frequency selection, resistance to interference, and robustness against multi-path fading channels [27, 28]. The OFDMA protocol provides the opportunity to accommodate large sets of wireless nodes and offers functionality for adaptive resource allocation techniques, which is greatly needed in large wireless networks. Due to its multiple access technologies, it has great potential for application in multi-user wireless communication systems [23]. OFDMA is also suitable for broadcasting and for multi-user applications, where the communication mechanism and the resource allocation techniques are designed to serve many nodes within the network's coverage area. where, it has the ability to dynamically assign a subset of subcarriers to individual nodes by combining the time division duplex (TDD), frequency division duplex (FDD), and orthogonal frequency division OFDM protocols [29].

2.1.1.4 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

The concept of the carrier sense technique is that the connected nodes randomly get access to the channel resources and use the channel frequency band to transmit their data. The basic procedure of the CSMA protocol is controlled by the Distributed Coordination Function (DCF), which consists of two functionalities, the carrier sense functionality, and back-off functionality. In the carrier sense functionality, the node that needs to transmit data continuously checks the channel status if idle or busy for a duration of a DCF Interference Space (DIFS). If the channel is busy, the node waits for a random back-off time and then rechecks the channel state. The back-off functionality computes the back-off time which is a uniformly chosen timer from [0, CW]. In general, the minimum and maximum values of the Contention Window (CW) are defined based on the protocol standards, and the back-off time exponentially increases with respect to the number of collisions in the network. If the channel is idle for the given back-off time, the back-off timer will be decreased by one, and the node transmits its data. The node waits for the receiver to transmit the Acknowledgment signal (ACK) if it received the data, otherwise, the data will be considered lost, and the node increases the CW, and it repeats the procedure until the retry time is exceeded, and the packet is dropped [1, 30, 31].

2.2 Wireless Networked Control Systems (WNCS)

In the last decade, there has been a dramatic push towards the decentralization and distribution of the control system structure, which implies considering the communication protocol within the control system structure. For instance, such a wireless distributed networked control system consists of an aggregated control and communication system, where the control system components are connected via shared wireless channel [32, 33]. This implies that sensor measurements and control actions are transmitted over a wireless communication channel, such as the one depicted in Fig 2.3-(left), where the control system consists of n sub-processes (SP) and m sub-controller(SC) distributed and connected to a wireless network, and in Fig 2.3-(right) shows that the centralized controller and the plant (sensors and actuators) are connected via wireless channel link. In practice, the advantages of the distributed WNCS that it provides flexibility, ease of installation and maintenance, less wiring and cables, and it provides an efficient way

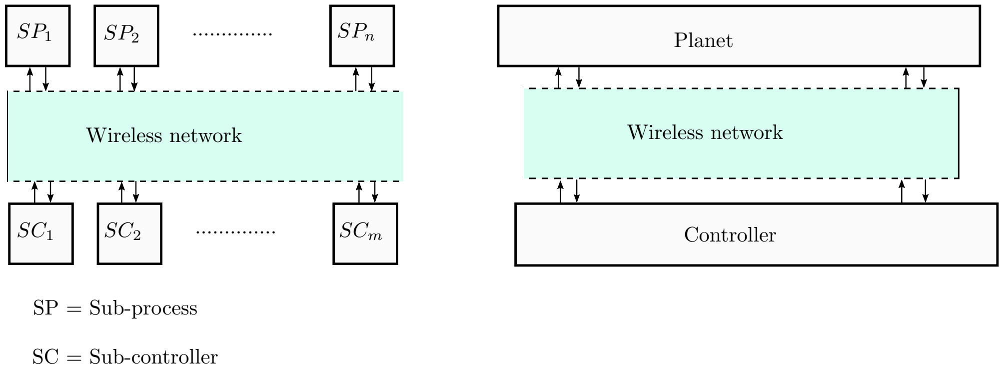

FIGURE 2.3: Wireless networked control system (WNCS), distributed system (left), centralized system (right).

of sharing data over the network. However, the presence of a wireless channel in the control loop introduces wireless communication issues into the system including, time delay, packet loss, communication latency, and channel congestion [34–36].

2.2.1 Effects of wireless networks in the control loop.

Conventional control system theory is basically based on a periodic sequence of system operations. The continuous-time signals are sampled at a fixed frequency and transmitted over a wired point-to-point connection [37]. Despite the wired cable characteristics, the presumption in conventional control system theory is the perfection of the transmission medium, which implies that the control signal and the sensor measurements are assumed to be received with no distortion, loss, delay, or attenuation. In contrast, the problems of the wireless transmission medium affect the packet transmitted over the wireless channel, which may get lost, delayed, or received with errors. The imperfection of the received signal influences the overall system performance or even destabilizes the process [3].

In general, from the perspective of control system theory, the problems induced by the wireless communication medium that have a high impact on system performance are: limited data rate, time delay, channel access, shared frequency band, and packet loss [13]. The main cause of such problems are obviously the characteristics of the wireless communication system and its limitations, such as limited coverage range, limited bandwidth, and limited data rate. The limited coverage range limits the mobility characteristics of the wireless nodes. The communication link will break down if the receiver node moves out of the maximum radio distance, which results in signal attenuation because of the existence of obstacles and moving sources of interference [25, 38]. It is obvious that a wireless communication network operates as a shared access medium in which multiple nodes are connected to one channel and share the same radio frequency. Therefore, the limited frequency bandwidth introduces the channel interference problem, where the operation in the limited wireless spectrum is the main source of wireless channel interference. The transmission of the radio frequency signal is mainly affected by different phenomena such as multi-path propagation, signal attenuation, fading, interference, or signal distortion. The difference between the strength of the transmitted signal and the received signal is known as signal attention, while signal fading occurs if the node receives a variant of the signal strength [39, 40]. The multi-path fading problem introduces problems with time delay and latency time, as the transmitted signal arrives at the receiver node through different angles, at a different time, or on a different frequency because of the electromagnetic waves scattering in the environment. The fading effect on the signal power in space is due to the angle spread, and frequency fluctuation is due to delay spread or time delay through the Doppler effect [39]. The other cause of time delay is the wave propagation due to the speed of light and the distance between the transmitter and the receiver nodes [40]. In fact, the limited wireless channel throughput is another cause of time delay. Wireless networks with higher throughput will often have shorter time delay than networks with lower throughput [30]. Moreover, the distance between the nodes results in an increase of transmission delay and of the Bit Error Rate (BER) with regard to signal-to-noise ratio (SNR) and transmission power [3]. It is known that the packet loss problem in wireless communication systems is due to failures of physical links or to network congestion. Nevertheless, even if the signal strength is a fixed parameter, sending a packet to a distant neighbor in the border area of the transmission range results in a higher probability of packet loss due to signal attenuation and node mobility [25, 38].

To visualize and study the effects of wireless networks on data transmission, we consider the quality of the wireless channel as a measure of the data received at the destination node. The quality of the wireless channel is modeled with respect to the signal-to-noise ratio (SNR), which is measured by the transmitter and receiver nodes, and is computed as the ratio between signal power and noise power [41] as follows:

$$SNR = \frac{P_s}{P_n}, \tag{2.2}$$

where, SNR represents the signal-to-noise ratio, P_s is signal power, and P_n is the noise power.

2.3 Vehicle dynamics and planar motion

As stated in chapter 1, the vehicle dynamics will be used as a benchmark for the simulation study and proof of concept of the algorithms which are developed in the context of this thesis. The following sections provide a summary of the vehicle dynamics including single track model, and vehicle motion dynamics model of the four wheels vehicles [42–44]. The presented vehicle's dynamic models will be used to formulate an optimal control problem of the vehicle dynamics, and it will be utilized in testing and evaluation of the algorithms that will be presented in the next chapters.

2.3.1 Vehicle dynamics: single-track model

The single-track model (STM) is widely used in automotive simulation and control because it is relatively simple and it has been proven that it is a good approximation for vehicle dynamics when lateral acceleration is limited to 0.4g on normal dry asphalt roads [43]. Single-Track-Model is a linear kinematic model for longitudinal, lateral and yaw motions. It does not consider the suspension forces, since suspension forces are inertial to the vehicle system and have no effect on the vehicle motions on the horizontal plane. It is obtained by lumping each of two wheels of one axle into one wheel located at the center of the respective axle. The single-track model considers the front and rear wheel steering to define the vehicle's lateral dynamics. Hereafter, the steering angle and slip angle will be restricted to relatively small values, the brake forces will be neglected, and the body roll and pitch behavior will be assumed to be zero.

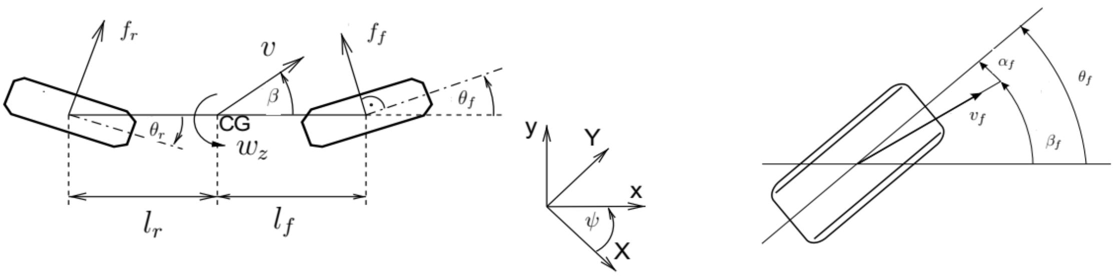

FIGURE 2.4: a) Vehicle single-track model. b) Wheel model variables

As depicted in Fig. 2.4-(a), θ_f and θ_r present the front and rear steering angles, respectively. The l_f presents the distance between the center of gravity (CG) and the front axel, l_r presents the distance between (CG) and the rear axel, and $l = l_f + l_r$ stands for the vehicle's wheelbase. The velocity vector v has the longitudinal and lateral projections v_x and, v_y on the vehicle coordination system (xyz), β is the vehicle side-slip angle, which is the angle between the longitudinal axis of the vehicle and the orientation of the vehicle velocity vector v.

The longitudinal and lateral velocity vectors v_x and v_y are computed as follows:

$$\begin{aligned} v_x &= v\cos\beta, \\ v_y &= v\sin\beta. \end{aligned} \tag{2.3}$$

Thus, the moment of inertia about the $z-$axis is denoted by I_z, and the mass of the vehicle is defined by m. The variables of the wheel dynamics model are depicted in Fig. 2.4-(b). Here, β_f presents the chassis sideslip angle, and α_f is the tire slip angle which is the angle between the velocity vector v_f and the tire center line. f_f and f_r are the front side force and the rear side force transmitted from the road surface via the wheels to the vehicle chassis. The yaw angle is ψ, and the yaw rate of the vehicle is defined $\omega_z = \dot{\psi}$. The equations of motion for the three degrees of freedom in the horizontal plane are with the vehicle mass m and the moment of inertia I_z with respect to the vertical axis through the center of gravity (CG) are defined as follows:

$$\text{Longitudinal motion: } F_x = -\, mv(\dot{\beta}+\omega_z)\sin(\beta) + m\dot{v}\cos(\beta), \tag{2.4}$$

$$\text{Lateral motion: } F_y = mv(\dot{\beta}+\omega_z)\cos(\beta) + m\dot{v}\sin(\beta), \tag{2.5}$$

$$\text{Yaw motion: } M_z = I_z\dot{w}_z. \tag{2.6}$$

From the equations (2.4), (2.5), and (2.6) the vehicle dynamic can be formulated as:

$$\begin{bmatrix} mv(\dot{\beta}+\omega_z) \\ m\dot{v} \\ I_z\dot{w}_z \end{bmatrix} = \begin{bmatrix} -\sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta) & \sin(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}, \tag{2.7}$$

Referring to Fig. 2.4-(b) and (2.3), the velocity components v_x and v_y in the longitudinal direction of the vehicle must be the same at the rear, the front wheels and at the center of gravity (CG) as follows:

$$v_r\cos\beta_r = v_f\cos\beta_f = v\cos\beta \tag{2.8}$$

The velocity components perpendicular to the centerline precede the yaw rate, according to:

$$\begin{aligned} v_f\sin(\beta_f) &= v\sin\beta + \ell_f\omega_z, \\ v_r\sin(\beta_r) &= v\sin\beta + \ell_r\omega_z. \end{aligned} \tag{2.9}$$

Finally, the variables v_f and v_r will be eliminated by dividing by the corresponding terms in (2.8). Therefore, the kinematic model as follows:

$$\tan(\beta_f) = \frac{v\sin(\beta) + \ell_f\omega_z}{v\cos(\beta)} = \tan(\beta) + \frac{l_f\omega_z}{v\cos(\beta)}, \tag{2.10}$$

$$\tan(\beta_r) = \frac{v\sin(\beta) + \ell_r\omega_z}{v\cos(\beta)} = \tan(\beta) - \frac{l_r w_z}{v\cos(\beta)}, \tag{2.11}$$

and the tire slip angles are:

$$\begin{aligned} \alpha_f &= \theta_f - \beta_f, \\ \alpha_r &= \theta_r - \beta_r. \end{aligned} \tag{2.12}$$

For the liner model, assume a very small of the sideslip angle $\beta << 1$, the model defined in (2.7) will :

$$\begin{bmatrix} mv(\dot{\beta} + \omega_z) \\ m\dot{v} \\ I_z\dot{w}_z \end{bmatrix} = \begin{bmatrix} -\beta & 1 & 0 \\ 1 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}, \tag{2.13}$$

the velocity v is assumed to be constant, $\dot{v} = 0$. From (2.13), $f_x = -\beta f y$ and $\beta << 1$

$$\begin{bmatrix} mv(\dot{\beta} + w_z) \\ I_z\dot{w}_z \end{bmatrix} = \begin{bmatrix} F_y \\ M_z \end{bmatrix}, \tag{2.14}$$

and considering the steering angles θ_f and θ_r are small, we get:

$$\begin{bmatrix} F_y \\ I_z\dot{w}_z \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ l_f & -l_r \end{bmatrix} \begin{bmatrix} f_f(\alpha_f) \\ f_r(\alpha_r) \end{bmatrix}, \tag{2.15}$$

Under normal driving conditions, the angles β, θ_f, θ_r, β_f, and β_r are relatively small, so equations (2.10) and (2.11) can be formulated as follows:

$$\begin{aligned} \beta_f &= \beta + \frac{l_f\omega_z}{v}. \\ \beta_r &= \beta - \frac{l_r\omega_z}{v}. \end{aligned}$$

Due to the small values of the small side-slip angles, the side forces F_f and f_r are proportional to the tire side-slip angle and are modeled as:

$$\begin{aligned} f_f &= c_f\mu\alpha_f, \quad \alpha_f = \theta_f - \beta_f, \\ f_r &= c_r\mu\alpha_r, \quad \alpha_r = \theta_r - \beta_r, \end{aligned} \tag{2.16}$$

where μ is the friction coefficient parameter of the tire and the road interface; the c_f, c_r parameters are the tire-cornering stiffness of the front and rear axle, respectively. The vehicle dynamics can be approximated by a linear state space model as follows:

$$\begin{bmatrix} \dot{\beta} \\ \dot{w}_z \end{bmatrix} = \begin{bmatrix} \frac{-(c_r+c_f)\mu}{m_v v} & \frac{-1+(c_r l_r - c_f l_f)\mu}{m_v v^2} \\ \frac{(c_r l_r - c_f l_f)\mu}{I_z} & \frac{-(c_r l_r^2 + c_f l_f^2)\mu}{I_z v} \end{bmatrix} \begin{bmatrix} \beta \\ w_z \end{bmatrix} + \begin{bmatrix} \frac{c_f\mu}{m_v v} & \frac{c_r\mu}{m_v v} \\ \frac{c_f l_f\mu}{I_z} & \frac{-c_r l_r\mu}{I_z} \end{bmatrix} \begin{bmatrix} \theta_f \\ \theta_r \end{bmatrix}. \tag{2.17}$$

2.3.2 Vehicle motion dynamics

Fig. 2.5 depicts the planar motion of a vehicle with 4 wheels. Independent torque τ_i and steering angle θ_i are the inputs applied to each wheel. The resulting planar motion is described by the vehicle's body state $\xi^T = [v_x, v_y, \omega_v]$, where v_x is the longitudinal speed, v_y is the lateral speed, and ω_v is the yaw rate, while the pitch and roll motions are neglected. A chassis reference frame xyz is mounted at the center of gravity CG of the vehicle in accordance with the ISO convention. This reference frame is used for the description of the evolution of the state variables v_x, v_y and ω_v with respect to the ground. Furthermore, a reference wheel frame $\varphi_i \gamma_i z_i$ is attached to the center of each

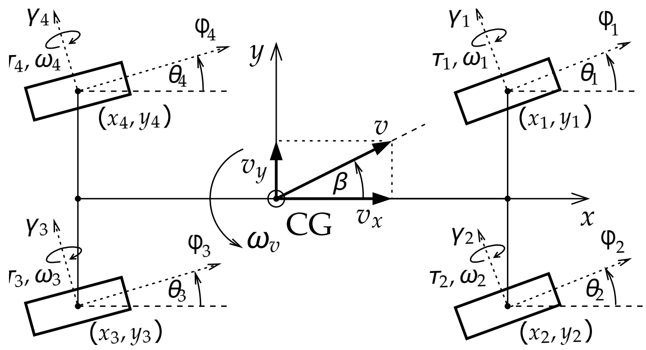

FIGURE 2.5: Inputs and states of the planar vehicle motion

wheel to model the wheel motion and tire friction forces. The transnational and yaw motion of the vehicle in the xyz reference frame is modeled as follows:

$$\text{rigid chassis: } M\dot{\xi} = g(\xi) + A_\varphi(\theta)F_\varphi + A_\gamma(\theta)F_\gamma \tag{2.18}$$

$$\text{four wheels: } I_w\dot{\omega}_i = r_w F_{i\varphi} + \tau_i \tag{2.19}$$

$$\text{tire model: } F_{i\varphi} = f_\varphi(\zeta_i, N_i), F_{i\gamma} = f_\gamma(\zeta_i, N_i), \tag{2.20}$$

where, $F_\varphi = [F_{1\varphi}, F_{2\varphi}, F_{3\varphi}, F_{4\varphi}]$ are the lateral tire friction forces, $F_\gamma = [F_{1\gamma}, F_{2\gamma}, F_{3\gamma}, F_{3\gamma}]$ are the longitudinal tire friction forces, $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]$ is the steering angles vector, $M = \text{diag}[m_v, m_v, I_v]$ is the mass matrix, m_v is the vehicle mass, and I_v is the chassis moment of inertia about CG, ω_i is the rotational speed of the wheel around the wheel axis γ_i, I_w is the wheel moment of inertia, and r_w is the effective wheel radius.

The lateral and longitudinal forces in (2.18) are transformed into the xyz frame by means of the matrices $A_\varphi(\theta)$ and $A_\gamma(\theta)$, which are determined by the geometrical parameters (wheelbase and track width) of the vehicle [42, 45]. Also note that the balance between force and moment is expressed in the chassis coordinate frame xyz, with $\xi^T = [v_x, v_y, \omega_v]$ which is caused by the yaw motion and is given by:

$$g(\xi) = m_v[\omega_v v_y, -\omega_v v_x, 0]^T. \tag{2.21}$$

The rotational motion of the wheel i about its own axis γ_i is presented in (2.19), and the general static tire model is presented in (2.20), where the functions f_φ and f_γ depend on the specific tire model. Assuming that the roll and pitch angles of the vehicle remain identically zero, the normal tire forces N_f are uniquely determined by the planar forces F_φ and F_γ as follows:

$$N_f = c + P_\varphi(\theta)F_\varphi + P_\gamma(\theta)F_\gamma, \tag{2.22}$$

where $N_f = [N_{f_1}, \cdots, N_{f_4}]^T$, while the constant vector c and the matrices P_φ, P_γ are again determined by the width, length, and height of the vehicle [42]. More specifically, $F_{i\varphi}$ and $F_{i\gamma}$ depend on the normal force N_{f_i} and on the friction coefficients $\mu_{i\varphi}$ and $\mu_{i\gamma}$ according to:

$$\text{longitudinal tire friction force: } F_{i\varphi} = \mu_{i\varphi} \cdot N_{f_i}, \tag{2.23}$$

$$\text{lateral tire friction force: } F_{i\gamma} = \mu_{i\gamma} \cdot N_{f_i}, \tag{2.24}$$

where the friction coefficient $\mu_i^T = [\mu_{i\varphi}, \mu_{i\gamma}]$ is a dynamic variable that determines the transmission of the lateral and longitudinal tire forces to the vehicle body for a fixed slip ζ_i.

Chapter 3

Distributed Optimal Control Problem

This chapter presents the theoretical background on optimal control theory. It focuses on the formulation of the convex optimization problem and introduces the methods for decomposing the centralized optimization problem into a distributed form. The first section 3.1 introduces the general form of the convex optimization problem and lists some definitions of convexity and the optimality conditions of the general optimal problem. The next section 3.2 presents the decomposition methods that are used to decompose the centralized optimization problem into a distributed form. Section 3.3 presents the communication network model considered in solving the distributed optimization problem over the network and describes the communication scheme imposed on the distributed system as well as the communication typologies. In section 3.4, we present the algorithms and methods, including the subgradient method, the projection method, and the consensus algorithms, that are mainly used in the proposed solution algorithm of the considered optimization problem. In section 3.5, we formulate the vehicle dynamics optimal control problem and derive the consensus-based projected primal subgradient algorithm, implementing it over different communication schemes. In the section 3.6, we report on an extensive simulation study of the proposed solution algorithm.

3.1 Convex optimization problem

In general, the objective of optimal control theory defined in [46] is to compute the control signals that fulfill with the process's physical constraints, and minimize/-maximize a predefined performance criterion. Wherein, a convex optimization problem is a problem that the objective function is convex and the feasible set is a convex set [47]. In general, we consider the following optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & J = \sum_{i}^{N} f_0^i(x) \\ \text{subject to} \quad & f_p(x) \leq 0, \; p = 1, \ldots, P \\ & h_q(x) = 0, \; q = 1, \ldots, Q \end{aligned} \tag{3.1}$$

where $x \in \Re^n$ is the state variable and n is the state-space dimension of the state variable of cost function f_0^i. The cost function f_0^i and the inequality constraints f_p are convex, while the equality constraints h_q are affine functions. To enhance clarity and understanding, we list the following definitions of the convex optimization problem [47–50]:

Definition 3.1.1 (Convex set)**.** A set $C \subset \Re^n$ is said to be a *convex* set if the line segment of every two points in the set C lies entirely within the set C, $\forall x, y \in C$, $\alpha \in [0, 1]$, i.e., the set C is a convex set if:

$$\alpha x + (1 - \alpha)y \in C. \tag{3.2}$$

Definition 3.1.2 (Convex function)**.** Consider a function $f_0^i : C_i \subset \Re^n \to \Re$. The set C_i is called the domain of f_0^i by $\text{dom}(f_0^i)$. The function f_0^i is called a *convex* function, $\forall x, y \in C_i$ *and* $0 \leq \alpha \leq 1$, if:

$$f_0^i(\alpha x + (1 - \alpha)y) \leq \alpha f_0^i(x) + (1 - \alpha)f_0^i(y). \tag{3.3}$$

Definition 3.1.3 (Strictly convex function)**.** Consider a function $f_0^i : C_i \subset \Re^n \to \Re$. The function f_0^i is called a *strictly convex* function, if $\forall x, y, x \neq y$ *and* $\alpha \in (0, 1)$:

$$f_0^i(\alpha x + (1 - \alpha)y) < \alpha f_0^i(x) + (1 - \alpha)f_0^i(y). \tag{3.4}$$

Definition 3.1.4 (Affine function)**.** Function $h_q : \Re^n \to \Re$ is called an *affine* function if it has the form $h_q(x) = Ax + b$, where $A \in \Re^{1 \times n}$ and $b \in \Re$.

Definition 3.1.5 (Local minimizer)**.** A point x^* is a *local minimizer* if there is a neighborhood C_i of x^* such that $f_0^i(x^*) \leq f_0^i(x)$, $\forall x \in C_i$.

Definition 3.1.6 (Strict local minimizer)**.** A point x^* is called a *strict local* minimizer of the function f_0^i if there is a neighborhood C_i of x^* such that $f_0^i(x^*) < f_0^i(x)$, $\forall x \in C_i$ and $x \neq x^*$.

Definition 3.1.7 (Global minimizer)**.** A point x^* is called a *global minimizer* of the function f_0^i if $f_0^i(x^*) \leq f_0^i(x)$, $\forall\, x \in \text{dom}(f_0^i)$.

3.2 Decomposition of centralized optimization problem into a distributed form

Decomposition of a centralized (original) optimization problem into a distributed form is usually done by dividing the original problem into N sub-problems. The decomposed sub-problems are mainly independent in the term of computation, and require coordination in order to converge to the optimal solution of the global optimization problem. Basically, coordination is usually done by introducing two levels of the optimization problem: a master problem and N distributed sub-problems. Fig. 3.1 illustrates the decomposition structure of the centralized optimization problem into a set of N sub-problems. We see that the communication network is introduced into the system structure to handle the coordination and exchange of variables between master problem and sub-problems.

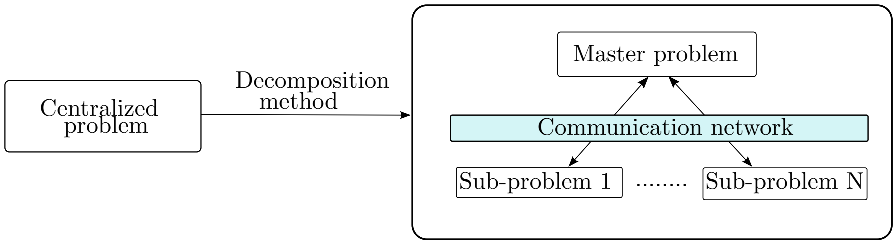

FIGURE 3.1: Structure of the decomposition of the centralized problem into N distributed sub-problems.

Dividing the centralized optimization problem into N distributed sub-problems produces a set of decision variables, such as local variables and global variables. The local variables are related internally to the sub-problem, while the global variables are shared variables between the sub-problems. Hence, the centralized optimal problem can be broken into smaller distributed sub-problems by using local copies x_i of the global decision variables x in all sub-problems. Then the decomposed distributed optimization problem is restated as:

$$\begin{aligned} &\underset{x_i}{\text{minimize}} \quad J = \sum_{i=1}^{N} f_0^i(x_i) \\ &\text{subject to} \\ &\qquad f_p^i(x_i) \leq 0,\ p = 1, \dots, P \\ &\qquad h_q^i(x_i) = 0,\ q = 1, \dots, Q \end{aligned} \tag{3.5}$$

where N is the number of sub-problems, $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ is the set of global state variables, and n is the state-space dimension, and it satisfies the condition $x_1 = x_2 = \cdots = x_N$. Despite the advantages of the distributed decomposition in terms of computation efficiency achieved by distributing the computation over the sub-problems, coordination and communication requirements between the master problem and the sub-problems and also between all sub-problems are imposed onto the system. The communication here allows sub-problems to exchange their decision variables with the master problem or with other sub-problems. In general, there are various decomposition techniques, such as primal decomposition, dual decomposition, indirect decomposition, and hierarchical decomposition [47, 51, 52], that are used to decompose the centralized problem into a distributed form. These will be described in detail in the following subsections.

3.2.1 Dual decomposition

Basically, the dual decomposition is based on deriving Lagrangian function of the optimization problem, which is defined by transferring the equality and inequality constraints to the objective function, and introducing a Lagrangian multiplier associated with each constraint. We applied dual decomposition on the decomposed optimization sub-problems (3.5) and formulated the Lagrangian function $L_i(x_i, \lambda_p^i, \nu_q^i)$ of each sub-problem i as follows:

$$L_i(x_i, \lambda_p^i, \nu_q^i) = f_0^i(x_i) + \sum_{p=1}^{P} \lambda_p^i f_p^i(x_i) + \sum_{q=1}^{Q} \nu_q^i h_q^i(x_i), \tag{3.6}$$

where $\lambda_p^i = \left\{\lambda_1^i, \lambda_2^i, ..., \lambda_P^i\right\}$ and $\nu_q^i = \left\{\nu_1^i, \nu_2^i, ..., \nu_Q^i\right\}$ are the Lagrange multipliers associated with inequality and equality constraints of the sub-problem i, respectively, and x_i is the set of primal variables. The corresponding dual function is given by:

$$g_i(\lambda_p^i, \nu_q^i) = \inf_{x_i} L_i(x_i, \lambda_p^i, \nu_q^i), \tag{3.7}$$

and the Lagrange dual function can be written as follows:

$$\begin{aligned} \text{minimize} \quad & g_i(\lambda_p^i, \nu_q^i) \\ \text{subject to} \quad & \lambda_p^i \geq 0, \end{aligned} \tag{3.8}$$

where, the inequality constraint $\lambda_p^i \geq 0$ must be satisfied, and we refer to the optimal value of the Lagrange dual problem (3.8) as g^*, and to the optimal solution of the primal problem (3.5) as f_0^*. The following definitions present the duality conditions between the dual and the primal problems [47].

Definition 3.2.1 (Weak duality)**.** *Weak duality* For the dual function g^* and the primal function f_0^*, if the condition $g^* \leq f_0^*$ holds, a weak duality exists even if the primal problem is not convex.

Definition 3.2.2 (Strong duality)**.** *Strong duality* For the dual function g^* and the primal function f_0^*, if the condition $g^* = f_0^*$ holds, a strong duality exists and $g^* = g(\lambda^*, \nu^*) = f_0^* = f_0(x^*)$.

In general, strong duality exists if there is a dual optimal λ^* and ν^* for the Lagrange dual problem (3.8) and there exists an optimal x^* for the primal problem (3.5). If strong duality does not exist and if the primal problem (3.5) is convex, i.e., if $f_0(x)$ and $f_p(x)$, $p = 1, \cdots, P$ are convex functions, we usually have weak duality.

3.2.2 Primal decomposition

The primal decomposition method is mainly used if the cost function is not separable and coupled with global variables shared with the sub-functions. A primal decomposition method is mainly used if there are some variables that are coupled with the aggregated general problem. For example, consider the following optimization problem where the cost function is coupled with one global variable ys:

$$\begin{aligned} \underset{x_i,y}{\text{minimize}} \quad & J = \sum_{i=1}^{N} f_i(x_i, y) \\ \text{subject to} \quad & x_i \in C_i, \qquad i = 1, \ldots, N \\ & y \in Y, \end{aligned} \tag{3.9}$$

the global variable y couples all functions f_i, and the set of variables $\{x_i, i = 1, \ldots N\}$ are local variables associated with each sub-function f_i. In primal decomposition, the general problem is divided into a master problem and a set of sub-problems. The primal master problem directly controls the assignment of the resources to the sub-problems using the global variable y. Specifically, the general optimization problem (3.9) is divided into N sub-problems as follows:

$$\begin{aligned} \text{sub-problem 1: } & \min_{x_1 \in C_1} f_1(x_1, y), \\ & \vdots \\ \text{sub-problem } N\text{: } & \min_{x_N \in C_N} f_N(x_N, y), \end{aligned} \tag{3.10}$$

and the master problem is defined as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} f_i^*(y) \\ \text{subject to } & y \in Y, \end{aligned} \tag{3.11}$$

with $f_i^*(y)$ being the optimal solution of the problems with respect to the global variable y, and where the master problem coordinates the sub-problems f_i by providing the optimal value of the global variable y.

3.2.3 Indirect decomposition

The basic technique used in indirect decomposition is to add an auxiliary variable to the general problem and to relax the problem by using the dual decomposition method. In general, the auxiliary variable consists of all coupling variables in the problem, wherein adding the auxiliary variable to all sub-problems provides more flexibility in applying either dual or primal decomposition. For example, the following optimization problem is coupled in the inequality constraint as follows:

$$
\begin{aligned}
\underset{x_i}{\text{minimize}} \quad & \sum_{i=1}^{N} f_0(x_i) \\
\text{subject to} \quad & x_i \in C_i, \\
& A_i x_i \leq y, \\
& y \in Y,
\end{aligned} \tag{3.12}
$$

where y couples the optimization problem in the equality constraint, and x_i is the local variable. Applying the indirect decomposition method, we introduce an auxiliary variable y_i and add it to the problem as follows:

$$
\begin{aligned}
\underset{y, y_i, x_i}{\text{minimize}} \quad & \sum_{i=1}^{N} f_0^i(x_i) \\
\text{subject to} \quad & x_i \in C_i, \\
& A_i x_i \leq y_i, \\
& y_i = y, \\
& y \in Y.
\end{aligned} \tag{3.13}
$$

The decomposable optimization problem (3.13) is created by introducing the auxiliary variable y_i and then relaxing the problem using the equality constraints $y_i = y$. There are some cases where the optimization problem is coupled in the cost function and in the constraints. Therefore, both dual and primal decomposition are combined to decompose the problem into a distributed form. We first use dual decomposition to compute the Lagrange multiplier and then use primal decomposition to solve the primal problem and compute the primal variables [53].

3.3 Distributed optimization problem over a network

Solving an aggregated distributed optimization problem over a communication network is motivated by the flexibility of networked systems. In fact, the communication network has a large impact on the field of decentralized and distributed systems, as it provides the communication capabilities for the sub-systems to cooperate in order to solve a global optimization problem [54]. However, we need to define the communication network and the capacity of the data exchange between sub-problems.

3.3.1 Communication network model

As a result of decomposing the centralized optimization problem into a distributed form, a communication scheme is introduced into the system structure, where the exchange of the global variables between the sub-problems takes place over a communication network. The structure of the communication network consists of communication medium, communication topology, and communication protocol. We consider a communication network consisting of N nodes sharing their state information according to a predefined communication topology. The network is modeled by a directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ represents the set of vertices, and $E \subseteq V \times V$ is set of edges $(v_i, v_j) \in E$ represents the connection between the two vertices i and j [31, 40]. The adjacency matrix A and the incidence matrix B represent the nodes' connection within the network, the adjacency matrix $A = (a_{ij})_{N\times N}$ of the graph G represents the adjacency of the vertices, and the entry a_{ij} shows whether vertex i is adjacent to vertex j and the a_{ij} is defined as follows:

$$a_{ij} = \begin{cases} 1, & if \quad (v_i,v_j)\in E \\ 0. & \text{otherwise} \end{cases} \tag{3.14}$$

The incidence matrix $B = (b_{ik})_{N\times M}$ of graph $G = (V, E)$ with $V = \{v_1, \ldots, v_N\}$ and $E = \{e_1, \ldots, e_M\}$ defines the connection between the vertices; the entry b_{ik} shows whether the vertex i is incident to edge j as follows [55]:

$$b_{ij} = \begin{cases} 1 & if \quad v_i\in e_j \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

The set of nodes $\mathcal{N}_i = \{j \in V : (v_i, v_j) \in E\}$ that have a direct connection with node i is denoted as neighbors of node i. Specifically, we consider the following communication typologies presented in Fig. 3.2, which define the interaction between nodes and the state information flow within the network:

1. **Unidirectional topology:** In the directed graph G, there is a unidirectional link between two vertices v_i and v_j if $(v_i, v_j) \in E$ and $(v_j, v_i) \notin E$ or $(v_j, v_i) \in E$ and $(v_i, v_j) \notin E$. Then vertex v_i is connected to v_j only in one direction and data flows from v_i to v_j in one direction or from v_j to v_i, respectively.
2. **Bi-directional topology:** In the directed graph G, there is a bidirectional link between the two vertices v_i and v_j if they are parallel adjacent, and if $(v_i, v_j) \in E$ and $(v_j, v_i) \in E$. This implies that vertex v_i transmits and receives from vertex v_j, and vertex v_j transmits and receives from vertex v_i.
3. **Broadcasting topology:** In the directed graph $G = (V, E)$, $\forall(v_i, v_j) \in E$ and $\forall(v_j, v_i) \in E$, the vertex $v_i \in V$ broadcasts its information to all $v_j \in V$ at the same time.

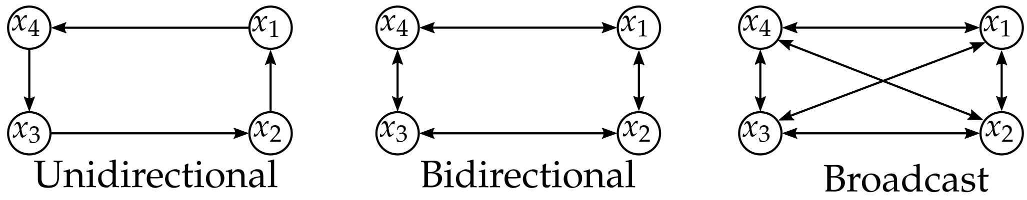

FIGURE 3.2: a) Unidirectional topology, b) Bidirectional topology, c) Broadcasting topology.

3.4 Solving the distributed optimization problem approach

Typically, optimization problems can be solved by a variety of iteration-based methods, such as the interior point method, the subgradient method, the descent method, the cutting-plane method, Newton's method, etc [50]. To solve the distributed optimization problem, we typically follow a three-step approach. Firstly, we decompose the centralized problem into a distributed form using a decomposition method. Secondly, we introduce a communication topology that facilitates the solution method. Finally, we select a method that is well-suited to the distributed nature of the problem. When selecting a method, we take into account implementation issues such as system performance, computation requirements, and the communication constraints of the network. The work of [56, 57] shows that the subgradient method is a simple method in terms of implementation, consumes fewer computation resources, has fewer memory requirements, and can be used for large problems such as those found in distributed systems. Therefore, we will use the subgradient method throughout this work to develop our solution algorithms.

3.4.1 Subgradient method

The subgradient method [57, 58] is a first-order simple algorithm. Its performance is affected by problem-scaling conditions. Even though it is a bit slower than second-order algorithms, it has the advantage of being easily adaptable to different kinds of problems and requires fewer computation resources to converge. To solve the optimization problem (3.5) using the subgradient method, iterated updating of the local variables x_i in the direction of the function f_0^i is performed. Thus, at each subgradient iteration k, the computed solution takes a step in the direction of a negative subgradient of the function $f_0^i(x_i)$ at gradient g_{x_i}. Hence, in each subgradient iteration, the local variable is updated as follows:

$$x_i[k+1] = x_i[k] - \alpha_k g_{x_i}[k], \tag{3.16}$$

where k is the iteration sequence, and $g_{x_i}[k] = \nabla_{x_i} f_0(x_i)[k]$ is the gradient of the local cost function f_0^i at $x_i[k]$. Here, α_k denotes the step size, which is fixed ahead of time.

In general, the cost function $f_0^i(x_i)$ is almost differentiable over its entire domain. Therefore, the selection of the step size usually has a high impact on the convergence of the convex optimization problem solution [56].

3.4.2 Consensus algorithm

The concept of consensus algorithms [59] is that the connected nodes must agree on a common decision, or the exchanged information must converge to the same value. In the context of distributed networked systems, node agreement on one value is highly important for many applications, such as exchange of measurements, time synchronization, data fusion, load balancing, and control systems [60, 61]. We consider a class of distributed networked systems where the connected nodes jointly solve optimization subproblems and exchange a copy of their global variables. In order for all nodes to agree on a common value of the global variables, we need to implement a consensus algorithm to ensure that the solution of the optimization problem converges to the optimal value.

3.4.3 Weighted average consensus algorithm

In general, the weighted average consensus algorithm is used if the state of each node in the network must converge to a specific weighted average value of the overall states. Hence, the impact of the state received from the neighbors $j \in \mathcal{N}_i$ on the state of node i is regulated by a weighting factor associated with each neighbor j [62, 63]. The algorithm updates the consensus state value $\check{x}_i$ of node i using an iterative procedure with respect to the weighting value of each received state $\hat{x}_j$ [64] as follows:

$$x_i^{(c+1)} = x_i^{(c)} + \sum_{j \in \mathcal{N}_i} w_{ij}(x_j - x_i^{(c)}), \tag{3.17}$$

where $x_i^{(c+1)}$ is the updated consensus state, x_j is the state received from neighbor $j \in \mathcal{N}_i$, c is the consensus iteration counter, and the weighting factor w_{ij} is the element of the doubly stochastic matrix $W \in \mathbb{R}^{N \times N}$ specified by the following conditions:

(a) $\mathbf{1}_{\boldsymbol{n}}^T \boldsymbol{W} = \mathbf{1}_{\boldsymbol{n}}^T$

(b) $\boldsymbol{W} \mathbf{1}_{\boldsymbol{n}} = \mathbf{1}_{\boldsymbol{n}}$

(c) $\rho(W - \frac{1_m 1_m^T}{m}) < 1$ must hold true, where $1_m \in \mathbb{R}^m$ refers to the column vector with all elements equal to 1, and ρ is the spectral radius of a matrix. It turns out that such a W guarantees the convergence of a consensus algorithm, *i.e.* $\lim_{c \to \infty} x_i^{(c)} = a 1_m$, for some $a \in \mathbb{R}$ depending on the initial value $x_i^{(0)}$ [65, 66].

The consensus algorithm considers conditions under which the local information states x_i of all nodes converge to the same value after experiencing a sufficiently large number of iterations. This value is typically determined by the initial values $x_i(0)$ and the double stochastic matrix $W \in R^{N \times N}$ associated with graph G [64, 67, 68].

3.4.4 Projection into a convex set

Projection algorithms are mainly used if the optimal solution of the convex optimization problem is a common point that exists in the intersection of two closed convex sets. To find this point, we use a projection algorithm such as alternating projection [69] or Dykstra's projection algorithm [70]. In general, the alternating projection algorithm uses a sequence of projection steps of the point into the most distant set. consider the following optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & x \in C, \end{aligned} \tag{3.18}$$

If the optimal solution of the optimization problem is point $x^* \in C$, and C is a convex set that exists in the intersection of m closed convex sets such as:

$$C = C_1 \cap \cdots \cap C_m, \tag{3.19}$$

then the projection of point x onto C is carried out by a successive projection of x onto $C_i, i = 1, \ldots m$. A sequential orthogonal projection of x strongly converges to a point $x^* \in C_{i,j}$ where $C_{i,j} = C_i \cap C_j$. For example, for two convex sets C_1 and C_2 $\in \Re^n$, projection $\mathcal{P}_1$ on C_1 and projection $\mathcal{P}_2$ on C_2, for the initial value $x_0 \in C_1$, the alternating projection algorithm generates a sequence of points $x_k \in C_1$, and $y_k \in C_2$ [71] by:

$$\begin{aligned} y_k &= \mathcal{P}_2(x_k), \\ x_{k+1} &= \mathcal{P}_1(y_k), \end{aligned}$$

where the projection sequences $\mathcal{P}_1$ and $\mathcal{P}_1$ will be executed iteratively in order to find the optimal point x_i^* in the intersection of both sets C_1 and C_2

3.4.5 Problem solver: projected subgradient with consensus algorithm

A projected subgradient based on consensus is formulated and will be used in this work as the main solution algorithm for the distributed convex optimization problem. The proposed solver of the considered optimal control problem consists of three layers: subgradient update, consensus algorithm, and alternating projection algorithm. Where, we integrate the alternating projection algorithm after each subgradient update in order to find the optimal solution within the intersection of the closed convex sets. The implementation of the projected subgradient layer is defined by:

$$x_i[k+1] = \mathcal{P}_{x_i}\big[x_i[k] - \alpha_k g_{x_i}[k]\big], \tag{3.20}$$

where $\mathcal{P}_{x_i}$ denotes the alternating projection sequence after each update of the subgradient $x_i[k+1]$. Following [67, 72], a projected subgradient method combined with the consensus algorithm is developed, where each node i performs the subgradient update and then exchanges its state information with its neighbors c times. This is followed by a projection of the solution into the convex sets. The complete solution algorithm is formulated as follows:

$$x_i[k+1] = \mathcal{P}_{x_i}\left[\sum [\boldsymbol{W}^c]_{ij}\left(x_j[k] - \alpha_k g_j(x_j[k])\right)\right], \tag{3.21}$$

where $[\boldsymbol{W}^c]_{ij}$ denotes the ij^{th} elements of the consensus matrix W, and c denotes the consensus iteration. The solver algorithm is essentially an iterative scheme with k being the counter of the subgradient iterations. It consists of the following three steps:

1. **Subgradient update step:** updating of the local primal variables x_i using the subgradient step:
$$x_i[k] = x_i[k] - \alpha_k g_{x_i}[k],$$

2. **Consensus step:** running the consensus algorithm for a finite number of iterations, so that each state x_i of all nodes reaches the consensus value:
$$x_i^{(c+1)}[k] = x_i^{(c)}[k] + \sum_{j \in N_i} w_{ij}(x_i^{(c)}[k] - \hat{x}_j^{(c)}[k]),$$

3. **Alternating projection step:** projecting $x_i^c[k]$ onto non-empty closed convex sets $\boldsymbol{C}_i$:
$$x_i[k+1] = Proj_{\boldsymbol{C}_i}(x_i[k]).$$

The proposed solution algorithm will be extended by the communication scheme, that will be performed for exchanging the state x_i with respect to the communication protocol and topology.

3.5 Benchmark: vehicle dynamics optimal control problem

This work is motivated by the new trends in the automotive research area, such as electric vehicles, autonomous driving [73–75], vehicle-to-vehicle communication (V2V), multi-vehicle communication, and vehicle to infrastructure communication (V2X) [76–80]. This work focuses on considering the effect of the wireless communication network on the optimal control problem, and investigating the effect of the communication layer on the control loop of the control problem. In this phase, we consider an optimal control problem of vehicle dynamics as a benchmark for our simulation study. A vehicle dynamics optimal control problem will be formulated and solved over different wireless communication network setups. In this section, we will formulate the optimization problem of the optimal allocation of tire adhesion forces and decompose this centralized optimization problem into a distributed form. Following that, we will solve the distributed optimization problem over a communication network.

3.5.1 Optimal control problem: Real-time allocation of tire adhesion forces

The main concept of integrated chassis control is to utilize redundant single-wheel actuation for steering, braking, and active suspension. This provides perceived reliable options for meeting the increasing demands on safety, and comfort. Recently, distributed on-board feed-forward control has been proposed in [81] as a platform for the formulation and implementation of the optimal control problem.
We consider the optimal allocation of the tire adhesion forces by minimizing the maximal tire adhesion for each tire [45]. We consider an optimization scenario that penalizes the maximal utilization of the tire's adhesion for a given maneuver and provides equal distribution of the adhesion utilization to all the vehicle's tires.

3.5.2 Problem formulation

The objective of the optimal control task consists of achieving the smallest possible utilization of the adhesion potential η_i and keeping it below the physical adhesion limit, which corresponds to the prescribed maneuver of the temporal evolution of the state of the vehicle's motion dynamics $\xi_d(t) = [v_{xd}(t), v_{yd}(t), \omega_{vd}(t)]$, where v_{xd} is the longitudinal speed, v_{yd} is the lateral speed, and ω_{vd} presents the yaw rate. Referring to equation (2.18) of vehicle dynamics:

$$M\dot{\xi} = g(\xi) + A_\delta(\theta)F_\delta + A_\gamma(\theta)F_\gamma,$$

, this equation can be rewritten as follows:

$$Y_d \triangleq M\dot{\xi_d} - g(\xi_d) = A_x F_x + A_y F_y, \tag{3.22}$$

where the explicit dependency on time t is dropped, and the vectors F_x and F_y collect the x- and y-components of the tire friction forces, while $A_x = A_\varphi(0)$ and $A_y = A_\gamma(0)$; see [42]. The linear equation (3.22) with the unknown force variables F_x and F_y is clearly undetermined. The variables Y_d and ξ_d are generated by a trajectory planning algorithm using a single-track model. The resultant reference planar motion Y_d is used to generate different driving maneuvers, which will be used in the formulation of the optimal control problem and for evaluating the performance of the solution algorithms.

$$\text{Longitudinal motion: } Fx_{sum} = ma_x = -f_f \sin(\theta_f) + f_r \sin(\theta_r), \tag{3.23}$$

$$\text{Lateral motion: } Fy_{sum} = ma_y = f_f \cos(\theta_f) + f_r \cos(\theta_r), \tag{3.24}$$

$$\text{Yaw motion: } M_z = I_z \dot{w}_z = l_f f_f - l_r f_r. \tag{3.25}$$

Referring to equations (2.18),(2.19), and (2.20) for the description of the dependency of $F_{i\varphi}$ and $F_{i\gamma}$ on the normal force N_f, the friction coefficients $\mu_{i\varphi}$ and $\mu_{i\gamma}$ are usually introduced according to:

$$\text{longitudinal tire friction force: } F_{i\varphi} = \mu_{i\varphi} \cdot N_f, \tag{3.26}$$

$$\text{lateral tire friction force } : F_{i\gamma} = \mu_{i\gamma} \cdot N_f, \tag{3.27}$$

where the coefficient $\mu_i^T = [\mu_{i\varphi}, \mu_{i\gamma}]$ is a dynamic variable that determines the transmission of the lateral and longitudinal tire forces to the vehicle body for a fixed slip ζ_i. It is physically confined by $\|\mu_i\| \leq \mu_{\max}$, i.e., the Kamm circle in Fig. 3.3, right, where $\mu_{\max}$ depends on the road conditions.

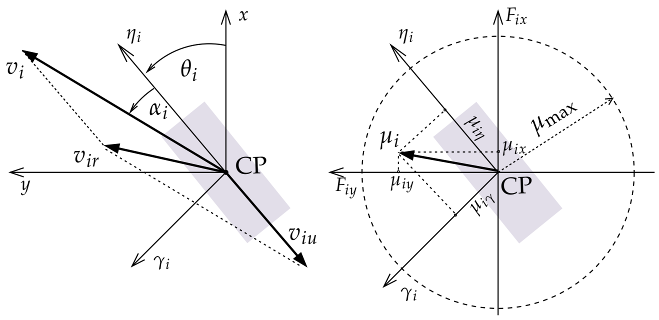

FIGURE 3.3: Definition of tire slip and friction coefficients

Following [45], in this work we consider the maximal adhesion potential of the tires as the cost function. The adhesion utilization of the i^{th} tire, η_i, is defined as the ratio between the magnitude of a tire force F_i and its normal force N_f (3.26) and (3.27):

$$\eta_i := \|\mu_i\| = \frac{\|F_i\|}{N_f}. \quad i = 1, \ldots, N \tag{3.28}$$

With reference to the concept of the Kamm circle in Fig. 3.3, $0 \leq \eta_i \leq \mu_{\max}$ represents a physically feasible condition. The task of optimal control consists of achieving the smallest possible utilization of the adhesion potential η_i, and to keep it below the adhesion limit in order to ensure an optimal safety reserve in every driving situation.

As stated earlier, the variables N_f can be expressed in terms of the tire forces in the xy coordinates by using:

$$\begin{aligned} N_f &= c + P_x F_x + P_y F_y, \\ P_x &= P_\eta(0), \\ P_y &= P_\gamma(0). \end{aligned}$$

However, it turns out that N_f is specified by the desired motion Y_d, as $N_f = GY_d$, where G is fixed by the vehicle geometry, which simplifies the inequality constraints $\sqrt{F_{xi}^2 + F_{yi}^2} - \eta N_f \leq 0$. Therefore, we shall be interested in minimizing the adhesion potential $\eta := \max_{i \in \{1,\ldots,N\}} \eta_i$, which brings us to the following formulation of the optimal control problem:

$$\begin{aligned} &\underset{\eta}{\text{minimize}} \quad J_0 = \eta \\ &\text{subject to:} \\ &\qquad f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta N_f \leq 0, \;\; i = 1, \ldots, N \\ &\qquad f_2 = \eta - \mu_{\max} \leq 0, \\ &\qquad h_1 = A_x F_x + A_y F_y - Y_d = 0. \end{aligned} \tag{3.29}$$

Moreover, it is important to emphasize that (3.29) represents a standard convex optimization problem, which can be reformulated as:

$$\begin{aligned} \underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \quad & J_0 = \sum_{i=1}^{N} \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right), \\ \text{subject to} \quad & f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f \leq 0, \\ & f_2 = \eta_i - \mu_{\max} \leq 0, \\ & h_1 = A_x F_x + A_y F_y - Y_d = 0. \end{aligned} \tag{3.30}$$

where we included the longitudinal force F_{xi} and the lateral force F_{yi} in the cost function, cardinally added a regularization term ($\epsilon \ll 1$) to regulate their effect on the cost function, and imposed the consistency constraints $\eta_i = \eta_j$. As a consequence, all tires must experience the same utilization by setting:

$$f_0(x) = \eta_i^2 + \epsilon^2 (F_{xi}^2 + F_{yi}^2), \tag{3.31}$$

and

$$x = \begin{bmatrix} \eta_1, & F_{x1}, & F_{y1}, & \ldots, & \eta_N, & F_{xN}, & F_{yN} \end{bmatrix}^T, \tag{3.32}$$

which present the primal variables of the optimization problem (3.30).

3.5.3 Decomposition of the problem into a distributed form

In order to decompose the centralized convex optimization problem (3.30) into a distributed form, we extract it into ($N = 4$) sub-problems and apply the primal decomposition method. Notice that there is no coupling between the sub-problems in the cost function J, and that it is separable because it optimizes only the local variables η_i, F_{xi}, F_{yi} of sub-problem i. On the other side, all sub-problems are coupled in the equality constraint:

$$h_1 = A_x F_x + A_y F_y - Y_d = 0,$$

where the vectors $F_x = [F_{x1}, \dots, F_{xN}]$ and $F_y = [F_{y1}, \dots, F_{yN}]$ consist of the longitudinal forces F_{xj} and the lateral forces F_{yj} of all the sub-problems $j \in \mathcal{N}_i$ of the neighbors of node i. The sub-problems are also coupled in the equality constraints $h_2 = \eta_i - \eta_j = 0$. Sub-problem i needs to receive the computed η_j of its neighbors $j \in \mathcal{N}_i$ to satisfy this equality constraint. Then we decompose the centralized problem into a distributed form by adding auxiliary variables consisting of the local copies η_j, F_{xj} and F_{yj} of the neighboring sub-problems' primal variables. The state variable of each sub-problem i consists of a copy of the state variables of the other sub-problems besides its own local variables. The state of the sub-problem $i = 1, \dots, N$ is defined as follows:

$$x_i = [\eta_{1,i}, F_{x1,i}, F_{y1,i}, \dots, \eta_{N,i}, F_{xN,i}, F_{yN,i}]^T, \tag{3.33}$$

The loss function is completely decomposable and equals the sum of the local functions $f_0^i(x_i)$. Each sub-problem considers its state variables x_i. The following set of sub-problems is defined:

$$\begin{aligned}
&\text{subproblem 1: } \min_{x_1 \in C_1} f_1^0(\eta_1, F_{x1}, F_{y1}),\\
&\text{subproblem 2: } \min_{x_2 \in C_2} f_2^0(\eta_2, F_{x2}, F_{y2}),\\
&\qquad\qquad\qquad \vdots\\
&\text{subproblem N: } \min_{x_N \in C_N} f_N^0(\eta_N, F_{xN}, F_{yN}),\\
&\qquad \text{subject to } f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f \le 0,\\
&\qquad\qquad\qquad f_2 = \eta_i - \mu_{\max} \le 0,\\
&\qquad\qquad\qquad h_1 = A_x F_x + A_y F_y - Y_d = 0.\\
&\qquad\qquad\qquad h_2 = \eta_i - \eta_j = 0, j \in \mathcal{N}_i,
\end{aligned} \tag{3.34}$$

where we added the auxiliary vectors $\hat{\eta}_j$, $\hat{F}_x$, and $\hat{F}_y$ consists of the coupling variables η_j, F_{xj}, and F_{yj} which need to satisfy the following equality constraints:

$$\begin{aligned} h_3 &= F_x - \hat{F}_{xj} = 0, \qquad j \in \mathcal{N}_i \\ h_4 &= F_y - \hat{F}_{yj} = 0, \qquad j \in \mathcal{N}_i \\ h_5 &= \eta_j - \hat{\eta}_j = 0, \qquad j \in \mathcal{N}_i, \end{aligned} \tag{3.35}$$

where $\hat{\eta}_j$, $\hat{F}_{xj}$, and $\hat{F}_{yj}$ indicate the copies of the states received from the neighbors $j \in \mathcal{N}_i$. Finally, the distributed version of (3.30) for each sub-problem i takes the following form:

$$\begin{aligned} &\underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \;\; \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right), \\ &\text{subject to:} \\ &\qquad f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_i \leq 0, \\ &\qquad f_2 = \eta_i - \mu_{\max} \leq 0, \\ &\qquad h_1 = A_x F_x + A_y F_y - Y_d = 0, \\ &\qquad h_2 = \eta_i - \eta_j = 0, \qquad j \in \mathcal{N}_i \end{aligned} \tag{3.36}$$

3.5.4 Implementation of the solution algorithm: Primal method

The overall fully decomposed feed-forward control scheme is illustrated in Fig. 3.4. It includes $N = 4$ independent single actuation units (nodes) for the manipulation of the vehicle dynamics. Each such unit consists of a local solver (optimal controller) of an optimization sub-problem (3.36) for the tire adhesion η_i, and each node is equipped with communication capability in order to connect to the communication network, where the nodes exchange their states with their neighbors $j \in \mathcal{N}_i$. The problem solver utilizes a consensus algorithm for the copies of the global variables received from the neighbors, which will converge to the optimal solution of the optimization problem for all nodes.

We implemented a primal projected subgradient with average consensus algorithm in each sub-problem solver, and exchanged the global variables η_i, F_{xi}, and F_{yi} with the neighbors' problem solvers.

3.5.4.1 Subgradient algorithm update

The specific implementation of the primal subgradient method is mainly based on updating the primal variables with respect to the gradient of the cost function j. Therein, the subgradient update for the primal variables η_i, F_{xi}, F_{yi} of the optimization problem

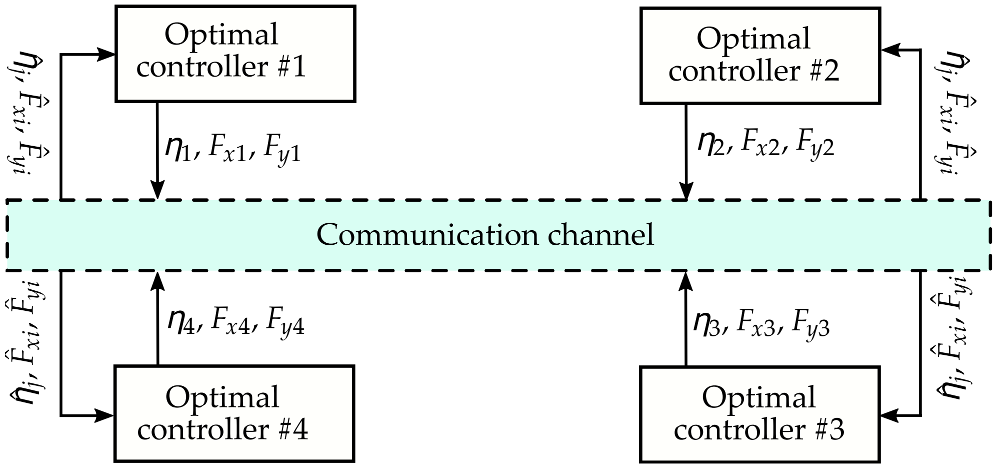

FIGURE 3.4: Fully distributed feed-forward control scheme consisting of N=4 optimal controller solvers of each sub-problem $i = 1, \dots 4$

(3.36) is performed through derivation to the gradient of the cost function

$$g_{x_i} = \frac{\partial j}{\partial x_i} = \frac{\partial}{\partial x_i}(\eta_i^2 + \epsilon^2(F_{xi}^2 + F_{yi}^2))$$

as follows:

$$\begin{aligned} g_{\eta_i} &= \frac{\partial j}{\partial \eta_i} = 2\eta_i, \\ g_{F_{xi}} &= \frac{\partial j}{\partial F_{xi}} = 2\epsilon^2 F_{xi}, \\ g_{F_{yi}} &= \frac{\partial j}{\partial F_{yi}} = 2\epsilon^2 F_{yi}, \end{aligned} \tag{3.37}$$

and the subgradient g_{x_i} update of each sub-problem i in (3.16) is given by

$$g_{x_i}^i[k+1] = \left[0, \dots, 0, 2\eta_i, 2\epsilon^2 F_{xi}, 2\epsilon^2 F_{yi}, 0, \dots, 0\right]^T. \tag{3.38}$$

Note that in each subgradient iteration, only the local variables η_i, F_{xi}, and F_{yi} of sub-problem i are affected. The update equations of the primal local variables are as follows:

$$\begin{aligned} \eta_i[k+1] &= \eta_i[k] - 2\alpha_k \eta_i[k] = \eta_i[k](1 - 2\alpha_k), \\ F_{xi}[k+1] &= F_{xi}[k] - 2\epsilon^2 \alpha_k F_{xi}[k] = F_{xi}[k](1 - 2\epsilon^2 \alpha_k), \\ F_{yi}[k+1] &= F_{yi}[k] - 2\epsilon^2 \alpha_k F_{yi}[k] = F_{yi}[k](1 - 2\epsilon^2 \alpha_k), \end{aligned} \tag{3.39}$$

where the subgradient algorithm updates the local variables η_i, F_{xi} and F_{yi} repetitively, the update is in the direction of subgradient $g_{x_i}[k+1]$, k is the subgradient iteration counter, and α_k is the step size of the subgradient algorithm.

3.5.4.2 Projection

To implement the alternating projection algorithm, we replaced the affine inequalities $f_2 = \eta_i - \mu_{\max} \leq 0$ in (3.36) with the equality $\eta_i - \mu_{\max} + \psi_i = 0$, $i = 1, \dots, N$, where we introduced the slack variables $\psi_i = [\psi_i, \dots, \psi_N] \in \mathbb{R}_+^N$. This leads to the extension of the decision variable x_i from (3.32) to

$$y_i^T = \left[\begin{array}{cccccccc} \eta_1, & F_{x1}, & F_{y1}, & \psi_1 & ,\ldots, & \eta_N, & F_{xN}, & F_{yN}, & \psi_N \end{array}\right], \tag{3.40}$$

in $\mathbb{R}^{4m}$. The constraints in (3.36) are then represented by the intersection of an affine subspace and a convex cone, i.e., $C_i = \mathcal{P}_i \cap \mathcal{K}_i$. The convex cone $\mathcal{K}_i \subset \mathbb{R}^{4m}$ can be defined by the composition

$$\mathcal{K}_i := \mathcal{K}_{1,i} \times \mathcal{K}_{2,i} \times \ldots \times \mathcal{K}_{m,i} \tag{3.41}$$

where $\mathcal{K}_{j,i} \subset \mathbb{R}^4$ represents the convex cones defined by

$$\mathcal{K}_{j,i} = \left\{ z \in \mathbb{R}^4; z^T M z \leq 0, Qz \geq 0 \right\} \tag{3.42}$$

where we introduce

$$z = [z_1, z_2, z_3, z_4] := [y_{4j-3,i}, y_{4j-2,i}, y_{4j-1,i}, y_{4j,i}]^T$$

For the sake of notational simplicity, we drop the subscripts i and j of z representing

$$z = [\eta_i, F_{xi}, F_{yi}, \psi_i]^T$$

, and

$$M = \operatorname{diag}[-1, 1, 1, 0], \quad Q_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that $Qz \geq 0$ guarantees $\eta_i \geq 0$ and $\psi_i \geq 0$. The projection onto $\mathcal{K}_i$ is given by

$$\operatorname{Proj}_{\mathcal{K}_i}(y_i) := \prod_{j=1}^{m} \operatorname{Proj}_{\mathcal{K}_{j,i}}(z). \tag{3.43}$$

Thus, it is sufficient to explore the projection of a point onto the cone $\mathcal{K}_{j,i}$ in $\mathbb{R}^4$. For a given z, the following projection scenarios may arise:

(a) If $z^T M z \leq 0$, and $z_1 \leq 0$, then

$$\operatorname{Proj}_{\mathcal{K}_{j,i}}(z) = \left[0, 0, 0, z_4^+\right]^T_,$$

(b) If $z^T M z \leq 0$ and $z_1 \geq 0$, then

$$\operatorname{Proj}_{\mathcal{K}_{j,i}}(z) = \left[z_1, z_2, z_3, z_4^+\right]^T_,$$

(c) If $z^T M z \leq 0$ and $z_1 \leq 0$, then

$$\text{Proj}_{\mathcal{K}_{j,i}}(z) = \left[\tfrac{1-t}{2} z_1,\, \tfrac{1-t}{2} z_2,\, -\tfrac{1-t}{2t} z_3,\, z_4^+\right]^T,$$

(d) If $z^T M z \leq 0$ and $z_1 \geq 0$, then

$$\text{Proj}_{\mathcal{K}_{j,i}}(z) = \left[\tfrac{1+t}{2} z_1,\, \tfrac{1+t}{2} z_2,\, \tfrac{1+t}{2t} z_3,\, z_4^+\right]^T,$$

where $z_4^+ = \max(0, z_4)$ represents the projection of z_4 onto $\mathbb{R}_+$, and

$$t = \frac{|z_1|}{(z_2^2 + z_3^2)^{1/2}}.$$

These analytical expressions are relevant as they speed up the computations. It is also easy to see that the projection $\mathcal{P}_i$ represents the affine subspace of an equation of the form:

$$\mathcal{P}_i := \{y_i; A_i y_i = b_i\}, \tag{3.44}$$

where A_i and b_i include the affine equality and inequality constraints in (3.30), as well as the non-negativity of the slack variables ψ_i. The projection of y_i onto $\mathcal{P}_i$ is given by the explicit expression

$$\text{Proj}_{\mathcal{P}_i}(y_i) := y_i - A_i^T \left(A_i A_i^T\right)^{-1} (A_i y_i - b_i), \tag{3.45}$$

where A_i is assumed to be a full rank matrix. The projection onto $\mathcal{K}_i$ is performed with a number of projection iterations, which guarantees the optimal solution $\left[\eta_i^*, F_{xi}^*, F_{yi}^*\right]$ as follows:

$$\begin{aligned} \eta_i[k+1] &= Proj_{P_i}(\eta_i[k]), \\ F_{xi}[k+1] &= Proj_{P_i}(F_{xi}[k]), \\ F_{yi}[k+1] &= Proj_{P_i}(F_{yi}[k]). \end{aligned} \tag{3.46}$$

Along with each updating iteration of the subgradient algorithm, an alternating projection will be executed for a number of iterations to find the optimal value in the intersection of the optimal convex set.

3.5.5 Average consensus and communication topology

As a part of the optimization problem solver, an average consensus scheme is implemented in each sub-problem solver. In order to unify the global variable copies, a communication topology is implemented to handle the communication process within the network and provide a communication mechanism for exchanging the state information in the network of sub-problem solvers. Consequently, each sub-problem solver transmits the state updates η_i, F_{xi}, and F_{yi} to its neighbors and every subgradient updates according to a predefined communication topology.

As stated above, we consider a vehicle consisting of $N = 4$ nodes. Each node is assigned an address $\{i : i = 1, 2, 3, 4\}$ as well as a sub-problem solver i, and is provided with communication capacity to handle the communication process. For now, we investigate the performance of the optimal controller behavior with respect to unidirectional and bidirectional communication topology. Both topologies are executed to perform the exchange of the state information between the nodes (sub-problem solvers) and are implemented as follows:

1 - Unidirectional topology: Node i receives the state update $x_j = \{\eta_j, F_{xj}, F_{yj}\}$ from its neighbor (in descending order) for every subgradient update for a number of consensus iterations. The unidirectional topology communication scheme is defined by the adjacency matrix A as follows:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

Node i performs the consensus update of its copies and circulates it to its next neighbor. The communication topology is performed in sequence based on the fixed neighbor's address, and an equal weight value is assigned for each node in the weighting matrix $W = 0.5A$. The number of consensus iterations defines how many times communication is established within each subgradient iteration.

2 - Bidirectional topology: Unlike the unidirectional communication topology, the bidirectional communication topology allows two-way communication between the neighbors, where node i receives the state update $x_j = \{\eta_j, F_{xj}, F_{yj}\}$ from its two neighbors $\{j : j = i - 1, \text{ and } j = i + 1\}$ based on the following adjacency matrix A:

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Node i performs the consensus update of its local copies and transmits its consensus state update to its neighbors. An equal weight is assigned for each node in the weighting matrix $W = \frac{1}{3}A$. The number of consensus iterations defines how many times communication is established within each subgradient update.

The following consensus update is performed according to the selected communication topology:

$$\begin{aligned} \eta_i^{(c+1)} &= \eta_i^{(c)} + \sum_{j\in\mathcal{N}_i} w_{ij}(\hat{\eta}_j - \eta_i^{(c)}), \\ F_{xi}^{(c+1)} &= F_{xi}^{(c)} + \sum_{j\in\mathcal{N}_i} w_{ij}(\hat{F}_{xj} - F_{xi}^{(c)}), \\ F_{yi}^{(c+1)} &= F_{yi}^{(c)} + \sum_{j\in\mathcal{N}_i} w_{ij}(\hat{F}_{yj} - F_{yi}^{(c)}), \end{aligned} \tag{3.47}$$

where, η_j, $\hat{F}_{xj}$, and $\hat{F}_{yj}$ present the values of the primal variables of the sub-problem j received at the solver i, see Fig. 3.4. The complete formulation of the projected subgradient with consensus update is presented in Algorithm 1.

3.6 Evaluation and Discussion

For the evaluation of the proposed distributed optimization algorithm, we use two lane change maneuvers under braking conditions generated by a single-track model. The first maneuver is designed to follow a moderate scenario with a maximal longitudinal deceleration of $a_x \approx -2$ [m/s^2], and maximal lateral acceleration of $a_y \approx 4$ m/s^2, at the initial speed $v = 80$ km/h. In the second maneuver, an extreme lane change scenario is carried out with maximum deceleration $a_x \approx -3$ m/s^2 and maximum lateral acceleration $a_y \approx 8$ m/s^2, at the speed $v = 120$ km/h. In both cases, dry road conditions with $\mu_{\max} = 1$ are assumed to exist.

The step size rule for the subgradient method is chosen to be $\alpha_k = 1/\sqrt{k}$. Our experience shows that the performance of the algorithm is quite sensitive to the initial conditions, which is a generally known fact for subgradient algorithms. In this case, the best results were achieved with approximately zero initial conditions. The alternating projection algorithm, which guarantees convergence to the projection point, issued a linear convergence with the rate of 0.9510.

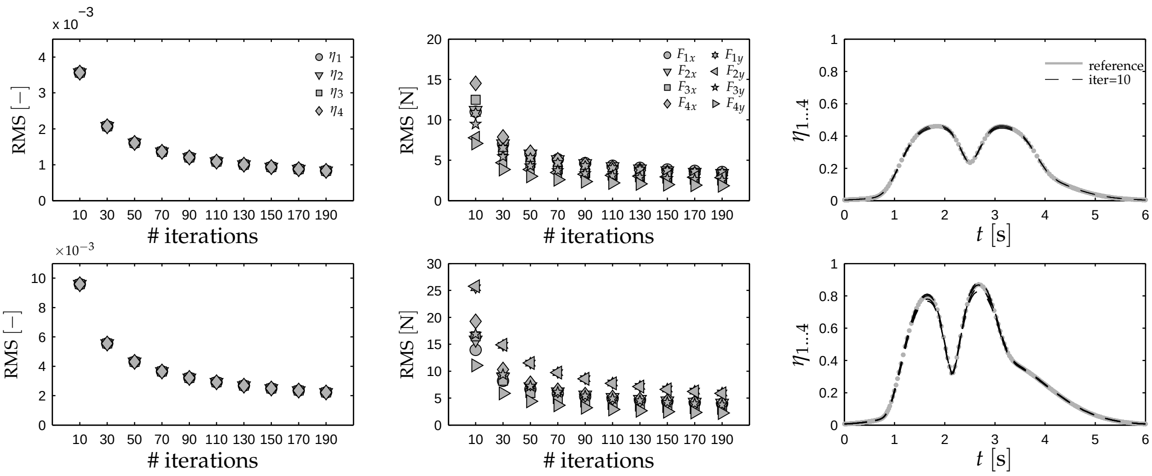

FIGURE 3.5: Error analysis for the moderate and for the extreme maneuver with a uni-directional topology and five consensus and five projection iterations.

The evaluation results corresponding to the moderate maneuver are depicted in the first row of Fig. 3.5, while the second row refers to the extreme maneuver. In both cases, we fixed the distributed optimization scheme to an "equi-weighted" uni-directional topology with five consensus and five projection iterations, and show the root-mean-square errors (RMS) over the maneuver period of the optimal solution of each node i, where $i \in \{1, \ldots, 4\}]$ of the distributed optimization scheme serves as a function of the number of subgradient updates. The RMS errors were computed with reference to the optimal solutions computed by a centralized interior-point algorithm. As expected, the errors in η_i (1st column), and in forces F_{xi} and F_{yi} (2nd column) decrease with an increasing number of iterations, and for a fixed number of iterations, the accuracy is better for the moderate maneuver. Moreover, in both cases the error is negligibly small, even for such a small number of subgradient updates as ten, which indicates fast convergence to a near-optimal solution. In the 3rd column, we depict the temporal evolution of the local adhesion variables η_i in the "worst-case scenario" with only ten subgradient updates and compare them to the optimal centralized value η. Observe that better matching is seen for the moderate maneuver, while slight differences emerged during the extreme maneuver. It is also important to emphasize that none of the tires experienced saturation in the latter case.

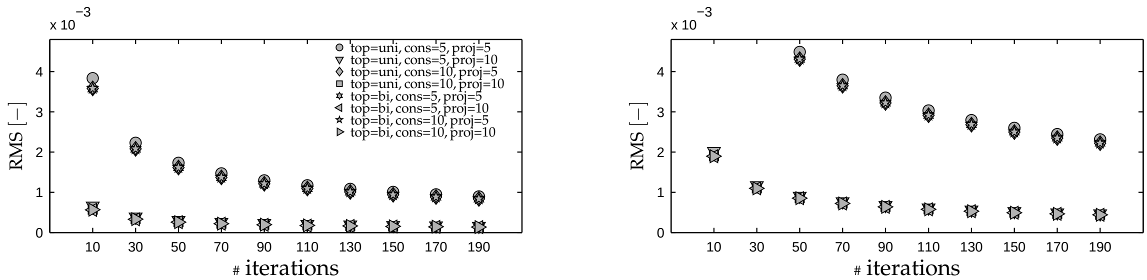

FIGURE 3.6: Convergence analysis under the impact of uni-/bi-directional topology and different numbers of subgradient, consensus, and projection iterations.

In Fig. 3.6, we compare the behavior of the algorithms for different communication topologies (uni- vs. bi-directional) and for different numbers of consensus and projection iteration steps. The plots depict RMS errors in η_i for the first (left figure) and the second maneuver (right figure). Generally, we got intuitive results: More accurate solutions were obtained for the bidirectional topology with a greater number of consensus and projection iteration steps. This trend is appreciably notable in the case of projections: The algorithm appears to exhibit even higher sensitivity than to the subgradient iterations. In our implementation, the alternating projection may be considered as closed within $20 - 30$ iterations for all initial conditions.

The impact of the communication topology on the convergence of the optimal solution depends on the number of consensus iterations in addition to the projection and subgradient iteration effect. Therefore, we fixed the number of projection iterations to 10 iterations, which guarantees optimal convergence of the solution, and tested the performance of the algorithm performance respect to the communication topologies. Fig. 3.7 presents the Q-Linear measure of the convergence error rate of the optimal solution with reference to the communication topology and the number of consensus iterations. We see in Fig. 3.7(left) that the convergence of the optimal solution is slower with the unidirectional topology and with fewer consensus iterations, while in the case of the bidirectional topology shown in Fig. 3.7(right), convergence is improved even with 5

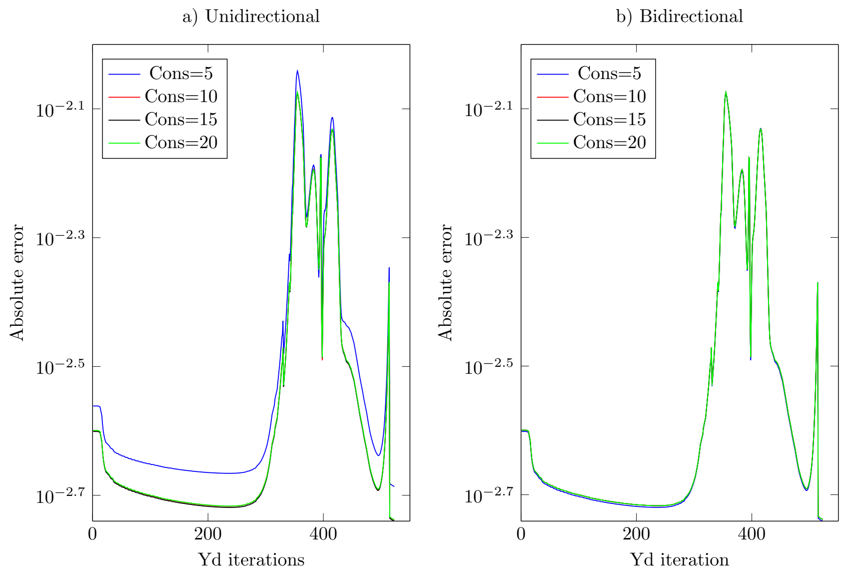

FIGURE 3.7: Absolute error of the convergence error rate of the optimal solution for the complete reference maneuver (Yd iterations) with unidirectional (left) and bidirectional (right) topology.

consensus iterations because data is exchanged in both directions, which improves the convergence rate.

We consider the convergence of the optimal solution in one optimization problem instance. For example, the convergence rate of the extreme maneuver instance (Yd=270) presented in Fig. 3.8 on the left side is the convergence rate of the algorithm with unidirectional communication topology, while the right figure illustrates the convergence rate with bidirectional topology.

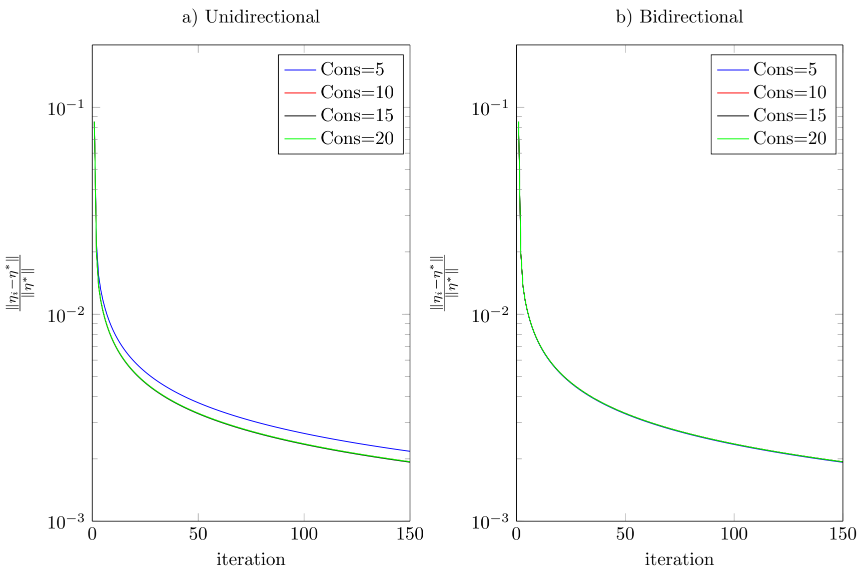

FIGURE 3.8: Convergence rate of the optimal solution of an extreme maneuver instance (Yd=270) with unidirectional (left) and bidirectional (right) topology.

Initialization;
$\eta_i = \eta_0$, $F_{xi} = F_{x0}$, $F_{yi} = F_{y0}$;
set $j \in \mathcal{N}_i$;
while *Subgradient loop* **do**
  Update subgradient state x_i;
    $\eta_i[k+1] \leftarrow \eta_i[k](1 - 2\alpha_k)$;
    $F_{xi}[k+1] \leftarrow F_{xi}[k](1 - 2\epsilon\alpha_k)$;
    $F_{yi}[k+1] \leftarrow F_{yi}[k](1 - 2\epsilon\alpha_k)$;
  Communication topology (Uni Or Bi);
  Average consensus;
  for $c \leq C$ **do**
    Node i: Receives $\hat{\eta}_j[k]$, $\hat{F}_{xj}[k]$, $\hat{F}_{yj}[k]$, $\forall\ a_{ij} = 1$;
      $\eta_i^{(c+1)} \leftarrow \eta_i^{(c)} + \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{\eta}_j - \eta_i^{(c)})$
      $F_{xi}^{(c+1)} \leftarrow F_{xi}^{(c)} + \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{F}_{xj} - F_{xi}^{(c)})$
      $F_{yi}^{(c+1)} \leftarrow F_{yi}^{(c)} + \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{F}_{yj} - F_{yi}^{(c)})$
    Node i: Transmits $\eta_i^{(c)}$, $F_{xi}^{(c)}$, $F_{yi}^{(c)}$, $\forall\ a_{ji} = 1$;
  end
  Projection;
$$\begin{aligned} \eta_i &\leftarrow Proj_{\boldsymbol{C}_i}(\eta_i[k]) \\ F_{xi} &\leftarrow Proj_{\boldsymbol{C}_i}(F_{xi}[k]) \\ F_{yi} &\leftarrow Proj_{\boldsymbol{C}_i}(F_{yi}[k]) \end{aligned}$$
  increment k;
end

Algorithm 1: Distributed projected subgradient with average consensus algorithm.

Chapter 4

Distributed event-triggered TDMA protocol

In chapter 3, we found that the performance of the solver algorithm is affected by the number of communication iterations between subgradient updates, as the nodes exchange their state periodically without considering the limited communication resources. In this chapter, we investigate the use of an event-triggered policy in order to link the performance of the application layer with the resource management of the communication layer and provide a mechanism for optimal use of the available communication resources by reducing the transmission requests through internal assessment of the changes in the node state. Section 4.1 provides an introduction of the event-triggered concept related to event-based communication systems. In section 4.2, we will describe in detail the formulation of the event-triggered scheme within the solution algorithm derived in chapter 3 and provide a simulation result. In section 4.3, we will implement the event-triggered algorithm for a distributed problem communicated over a TDMA-based wireless network and provide a simulation result and discussion. Section 4.4 presents the conclusion of the chapter.

4.1 Event-trigger-based communication

Standard control system theory is based on periodic and sampled time events, where the controller action and system state are sampled at fixed intervals. However, the increasing number of modern devices that share computation and communication resources, along with the development of network technology and large-scale distributed systems, has led to new challenges. For example, wireless networked systems require effective management of communication resources to conserve energy and make efficient use of the available frequency spectrum. To address these challenges, aperiodic control and communication concepts, such as event-triggered and self-triggered approaches, have been introduced in distributed system theory. In these approaches, the system only reacts when its internal state deviates or when the detection of state errors reaches a predefined threshold. By using these concepts, distributed control systems can reduce communication overhead and achieve energy savings, while still maintaining the desired system performance. For example, in [82], the state error threshold is defined based on the system's performance criterion. Similarly, in the context of multi-agent systems, the work [83] proposed a distributed event-triggered control strategy, where the control value of each subsystem converges to the average value with respect to the average difference of its neighbors' control signal. Furthermore, [84] presented an event-triggered scheme for a linear system, whose objective is to reduce the computation effort and provide better utilization of communication resources.

In the context of optimal control over a wireless network, the work [85] developed a framework for event-based communication over a distributed wireless sensor network. It introduced an event-triggered algorithm to reduce data transmission between wireless nodes while solving a distributed optimization problem. This was followed by the work [86], which proposed an event-triggered algorithm added to an augmented Lagrangian method to solve a distributed network utility maximization optimization problem. In [87], the same event-triggered algorithm is used to solve a distributed optimization problem regarding computation of the optimal power distribution in a micro-grids network. Additionally, Model Predictive Control (MPC) combined with an event-triggered condition was introduced in a wireless networked control loop as part of a resource-aware policy in [88], and as resource-aware communication scheduling for wireless protocols that allows only one node to transmit at one time instant in [89].

4.1.1 Event-triggering condition

The basic concept of an event-triggered condition is based on the base internal state error, which depends on computing the difference between the most recently updated state and the previous state, and is formulated as a standard event-triggered condition as follows:

$$\Delta x_i[k] = \|x_i[k] - x_i[k-1]\|\,, \tag{4.1}$$

where Δx_i is the internal state error, $x_i[k]$ is the updated state, and $x_i[k-1]$ is the previous state of sub-system i. For example, in a distributed networked system, an event-based communication scheme traces the state deviation of sub-system i and computes the difference between the actual state and the last transmitted state. Accordingly, the sub-system transmits only the state that violates the triggering threshold to its neighbors, the triggering condition (4.1) can be rewritten as follows:

$$\Delta x_i[k] = \|x_i[k] - \hat{x}_i[\ell_i]\|\,, \tag{4.2}$$

where $x_i[k]$ is the updated state, $\hat{x}_i[\ell_i]$ is the last transmitted state of node i, and ℓ_i is the state transmission counter.

4.1.2 Equality-constraints-based event-triggering threshold

Referring to the distributed optimization problem (3.5), and motivated by the idea that the equality constraints h_q^i gradually decay to zero with each increment of the subgradient iteration (3.16), the solution of the optimization problem must converge to the optimal value in each step. Therefore, the selection of the event-triggered threshold must provide an efficient way to reduce the use of communication resources, regulate the message flow within the network, and also guarantee the performance of the overall system. Knowing that, the trade-off between the communication effort and the convergence of the distributed optimization problem is dominated by the amount of data and the quality of the data exchanged between the nodes [64]. Here we define the event-triggering threshold Δh_q^i as a function of the convergence of the equality constraints $h_q^i(x_i)$ to zero.

Practically, a node transmits its state to its neighbors with respect to the convergence of its equality constraints to zero, which is defined as follows:

$$\Delta h_q = \beta_1 \left\| h_q^i(x_i) \right\| + \beta_2, \tag{4.3}$$

where β_1 and β_2 are the tuning parameters of the event-triggering threshold that regulate the accepted state error level. The final form of the event-triggered condition can be written as:

$$\| x_i[k] - \hat{x}_i[\ell_i] \| \leq \beta_1 \left\| h_q^i(x_i) \right\| + \beta_2, \tag{4.4}$$

This regulates the node state's transmission with respect to the Δh_q^i, and stops requesting communication resources if its state error is less than the acceptable error.

4.2 Formulation of the TDMA-based event-triggered algorithm

The goal of the event-triggered policy is to define a way to maintain the system performance when data transmission between subsystems is reduced. When reducing the data exchange, a distributed system has to keep up its performance and keep the communication resources at an accepted level [90, 91]. An event-triggered scheme is therefore added to the TDMA-based communication protocol, where it regulates the nodes' communication requests and continuously provides the TDMA scheduler with a list of nodes that acquire time slots. Then, the TDMA scheduler rebuilds the TDMA frame by appointing a time slot for the nodes that urgently need to transmit their state according to a predefined criterion. Here, we formulate the following definition of the event-triggered scheme and its relation to the TDMA scheduler as follows:

Definition 4.2.1 (Event-triggered - TDMA-based network)**.** For a distributed wireless networked system, all nodes are connected through a TDMA-based protocol. For a node i, with the updated state $x_i[k+1]$ and the last transmitted state $\hat{x}_i[\ell]$, if the event-triggered condition $\|x_i[k+1] - \hat{x}_i[\ell_i]\|$ exceeded the threshold $\beta_1 \|h_z[k]\| + \beta_2$, then node i will be assigned a time slot T_{si} over the frequency F_i and will transmit its state $x_i[k+1]$ to its neighbors and update its last transmitted state $\hat{x}_i[\ell+1] = x_i[k+1]$.

The proposed extension to the solution algorithm 1 consists of introducing the event-triggered scheme in the communication scheduler of the TDMA protocol, aiming to reduce the communication effort between the nodes within the wireless network, while at the same time keeping the system performance at an acceptable level by allowing only nodes that satisfy the event-triggered condition to transmit their states. We extend the optimal control feed-forward vehicle dynamics approach by adding an event-triggered scheme to the communication layer in order to reduce the number of transmission requests by each node.

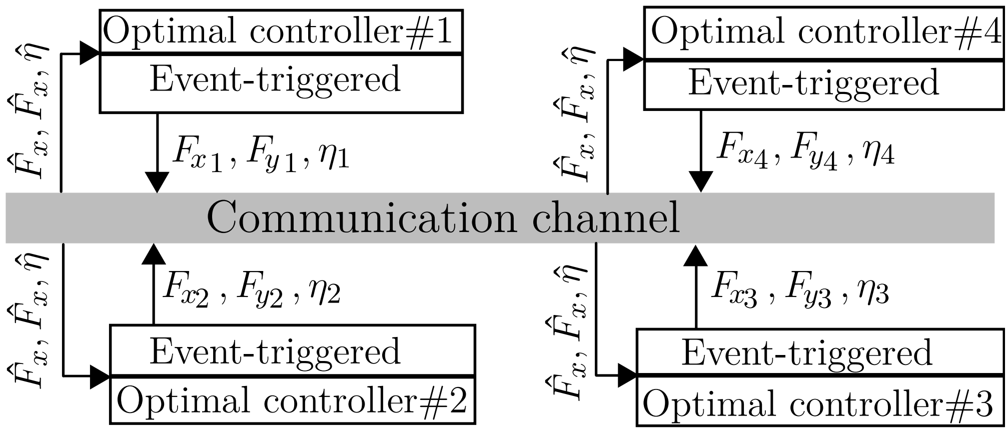

FIGURE 4.1: Distributed event-triggered optimization model.

Figure 4.1 presents the updated system structure where the event-triggered scheme is added to the solution algorithm (optimal controller) in order to trace the internal state error of the node. The event-triggered layer computes the difference between the actual updated state (local variables) F_{xi}, F_{yi}, η_i and the latest transmitted state $\hat{F}_{xi}$, $\hat{F}_{yi}$, $\hat{\eta}_i$ compared to the predefined event-triggered threshold.

The extended distributed event-triggered-based TDMA algorithm consists of three layers including projected subgradient with consensus update, event-triggered scheme, and communication layer-based TDMA protocol. The updated algorithm formulated in (Algorithm 2) is functioning in the following order:

1. **Layer 1: projected subgradient and consensus update**

   Node i updates its local variables η_i, F_{xi}, and F_{yi} by executing the following projected consensus subgradient steps:

(a) **Subgradient update**

The first layer presents the standard projected subgradient solver, which updates the state of node i by computing the subgradient of the primal variables η_i, F_{xi} and F_{yi} in subgradient iteration $k+1$.

$$\begin{aligned} \eta_i[k+1] &= \eta_i[k](1-2\alpha), \\ F_{xi}[k+1] &= F_{xi}[k](1-2\epsilon\alpha), \\ F_{yi}[k+1] &= F_{yi}[k](1-2\epsilon\alpha). \end{aligned} \tag{4.5}$$

The state variable of node i is defined using the updated primal variables $x_i = [\eta_i, F_{xi}, F_{yi}]$.

(b) **Update of the equality constraints**

Node i updates the equality constraint h_1 as a state error threshold of the event-triggered conditions. The threshold is updated using the actual state of node i and the most recently received state $\hat{F}_{xj}$ and $\hat{F}_{jy}$ from the neighbors $j \in \mathcal{N}_i$ as follows:

$$h_1[k] = A_x\hat{F}_{xj}[k] + A_y\hat{F}_{jy}[k] - Y_d. \tag{4.6}$$

In every subgradient iteration k, nodes trace their internal event-triggered threshold and use it as a reference for their transmission condition. We see that the threshold is dynamically changed with respect to the states received from the neighbors and the internal state update of each node.

(c) **Average consensus:**

Node i computes the average consensus of the copies of the received state variables $\hat{x}^i_j = \left[\hat{\eta}_j[k], \hat{F}_{xj}[k], \hat{F}_{jy}[k]\right]$ from the neighbors $j \in \mathcal{N}_i$ as follows:

$$\begin{aligned} \eta_i^{(c+1)}[k] &= \eta_i^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{\eta}_j[k] - \eta_i^{(c)}[k]\right) \\ F_{xi}^{(c+1)}[k] &= F_{xi}^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{F}_{xj}[k] - F_{xi}^{(c)}[k]\right) \\ F_{yi}^{(c+1)}[k] &= F_{yi}^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{F}_{jy}[k] - F_{yi}^{(c)}[k]\right). \end{aligned} \tag{4.7}$$

The number of consensus iterations greatly affects the convergence of the optimal solution. The maximum number C of consensus iterations that makes all nodes converge to the optimal value requires the same number of communication iterations.

(d) **Projection:**

Following the same projection procedure defined in algorithm 1, the projection of $\eta_i[k]$, $F_{xi}[k]$,$\eta_i[k]$ and $\bar{\eta}_i[k]$ onto the intersection of a cone and a hyperplane using an alternating projection method is carried out as follows:

$$\begin{aligned} \eta_i &= Proj_{\boldsymbol{\chi}^i}(\bar{\eta}_i[k]), \\ F_{xi} &= Proj_{\boldsymbol{\chi}^i}(\bar{F}_{xi}[k]), \\ F_{yi} &= Proj_{\boldsymbol{\chi}^i}(\bar{F}_{yi}[k]). \end{aligned} \tag{4.8}$$

2. **Layer 2: event-triggered scheme:**

   The second layer includes the event-triggered scheme, which tests for internal state errors and decides whether the node needs to transmit its state or not. Mainly, in every subgradient update iteration, node i checks its internal state error using the event-triggered conditions:

$$\begin{aligned} \left\|F_{xi}[k] - \hat{F}_{xi}[\ell_i]\right\| &\geq \beta_1 \|h_1[k]\| + \beta_2, \\ \left\|F_{yi}[k] - \hat{F}_{yi}[\ell_i]\right\| &\geq \beta_1 \|h_1[k]\| + \beta_2, \\ \|\eta_i[k] - \hat{\eta}_i[\ell_i]\| &\geq \beta_1 \|h_1[k]\| + \beta_2. \end{aligned} \tag{4.9}$$

   where $h_1[k]$ are the equality constraints, k is the subgradient update counter, and ℓ_i is the counter of the transmitted state of each individual node i. If the difference between the updated state variable and its last transmitted value is greater than the state error threshold, the node requests a time slot in the next TDMA frame by activating the Request to Send (RTS) signal.

3. **Layer 3: TDMA protocol:**

   A TDMA scheduler is imposed on the communication layer to reconstruct the TDMA frame and assigns the time slots for the nodes to exchange their states. If the event-triggered conditions of node i are satisfied and it activates the RTS signal, the TDMA scheduler reconstructs the TDMA frame by assigning a time slot Ts_i for all nodes $i \in \mathcal{N}$ within the next frame. Node i then transmits its updated variables F_{xi}, F_{yi}, and η_i to its neighbors $j \in \mathcal{N}_i$. It then increments the transmission counter ℓ_i and sets $\hat{F}_{xi}[\ell_i] = F_{yi}[k]$, $\hat{F}_{yi}[\ell_i] = F_{yi}[k]$, and $\hat{\eta}_i[\ell_i] = \eta_i[k]$. The reconstructed TDMA frame provides the transmission sequence of the nodes according to their request.

4.2.1 Simulation: Event-triggered algorithm

In this section, we report on a simulation study we conducted of the performance of the event-triggered solution algorithm without considering the effect of the wireless channel. For this purpose, we used an update of the basic simulation setup defined in the previous chapter. Recall that this setup consisted of a lane change maneuver under braking conditions to evaluate the vehicle dynamics performance. We defined the parameters of the algorithm with a maximum longitudinal deceleration of $a_x \approx 3$ m/s^2 and a maximum lateral acceleration of $a_y \approx 8$ m/s^2 at the speed $v = 120$ km/h under dry road conditions with $\mu_{max} = 1$. The subgradient method was used with step size $\alpha = 0.1/\sqrt{k}$, where k was the subgradient counter iteration, and 20 alternating projection iterations were used in order to eliminate the effect of the projection scheme on the solution. We applied the same initial conditions for the subgradient method as in section 3.6. The event-triggered parameters β_1 and β_2 were chosen to guarantee system performance and were set to $\beta_1 = 0.3$ and $\beta_2 = 0.0002$ for tuning the transmitting rate of the node and broadcasting its state update η_i, F_{xi} and F_{yi}.

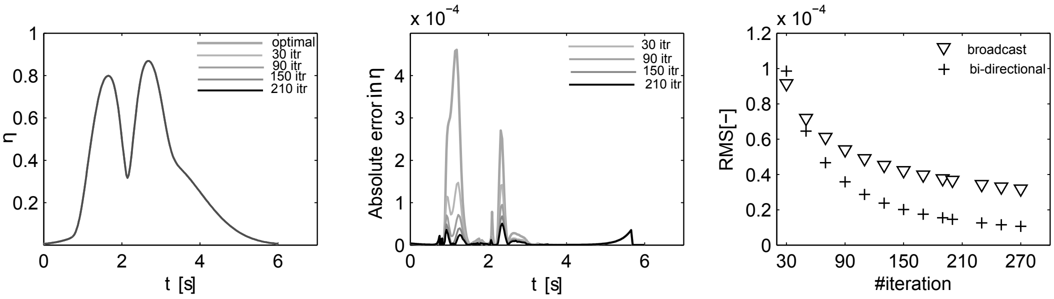

FIGURE 4.2: a) Adhesion potential utilization η_i, b) Absolute error, c) RMS error analysis. For 30, 90, 150, and 210 subgradient iterations.

Fig 4.2 presents the performance of the algorithm over different numbers of subgradient iterations, fig 4.2-a depicts the local adhesion η_i with 30, 90, 150, and 270 subgradient iterations compared to the optimal value η^*. It can be seen that all η_is match quite closely with η^*. Additionally, we see that all four wheels experienced the same adhesion utilization. We note that the event-triggered scheme was active and each node i broadcasted its state variable to all neighbors $j \in \mathcal{N}_i$ based on its internal triggering condition. The middle figure 4.2-b depicts the absolute errors of η_i with 30, 90, 150, and 270 subgradient iterations computed based on the difference to the optimal η^*. We observe that the maximum absolute error reaches the maximum value (5×10^{-5}) at the difficult turn-points of the double lane-change testing maneuver. In the right figure 4.2-c, we present the root mean square error (RMS error) of η_i, always with respect to η^* as a function of different numbers of subgradient iterations. Here we compare the RMS error of the proposed algorithm with the total order broadcast topology and the distributed algorithm using the 'bi-directional' topology with 20 consensus steps and 20 alternating projection iterations as discussed in the previous chapter 3 and in [92]. Based on the RMS error, it can be seen that the performance of the distributed event-triggered algorithm using the broadcast topology is better than that of the distributed algorithm using the bi-directional topology. This implies that the convergence of the proposed algorithm is faster than previous work presented in section 3.6 [92]. Generally, we observed that the RMS error decreases when the number of subgradient iterations increases.

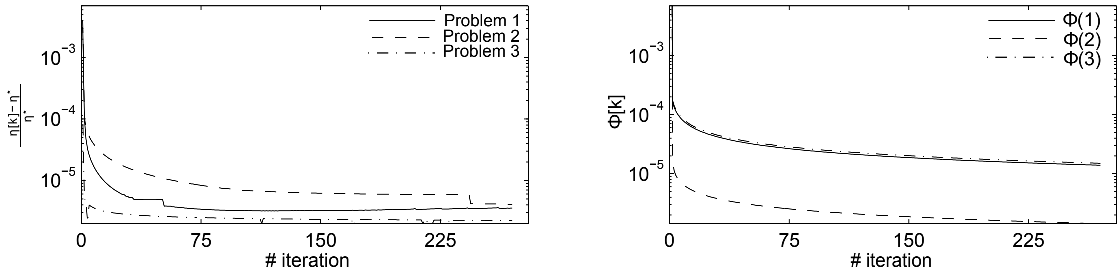

FIGURE 4.3: a) Q-linear measure of η_i convergence rate, b) Decay of the equality constraints Φ. Problems 1, 2, and 3 refer to different maneuver instances.

Fig 4.3 depicts the logarithmic convergence of the algorithm over 270 subgradient update iterations. Fig 4.3-a compares the relative error measure of the convergence rate of the local adhesion η_i referring to three maneuver instances with different levels of difficulty level: an extreme maneuver instance (Problem 2: Yd=270), a moderate maneuver instance (Problem 1: Yd=180), and a simple maneuver instance (Problem 3:Yd=80). Our finding is that in the case of the extreme maneuver instance, the local adhesion η_i convergences slower than in the other two maneuver instances. The right side of Fig 4.3-b shows the logarithmic decay of the equality constraints for the three maneuver instances (Problem 1), (Problem 2), and (Problem 3). For (Problem 2), we see that the equality constraints decay to nearly zero after the 30^{th} subgradient iteration, which is an indication of the convergence speed of the proposed algorithm.

4.3 Wireless-network-based TDMA protocol and broadcasting topology

In this section, we consider a wireless network with TDMA protocol and use the broadcasting topology for the communication sequence. The proposed structure consists of a network served by $N = 4$ nodes forming one vehicle cluster. Therefore, the channel time is divided into a number of fixed frames. Each frame consists of $T_s = 4$ time slots, and each node i is assigned a fixed time slot T_{si} repetitively, as depicted in Fig. 4.4.

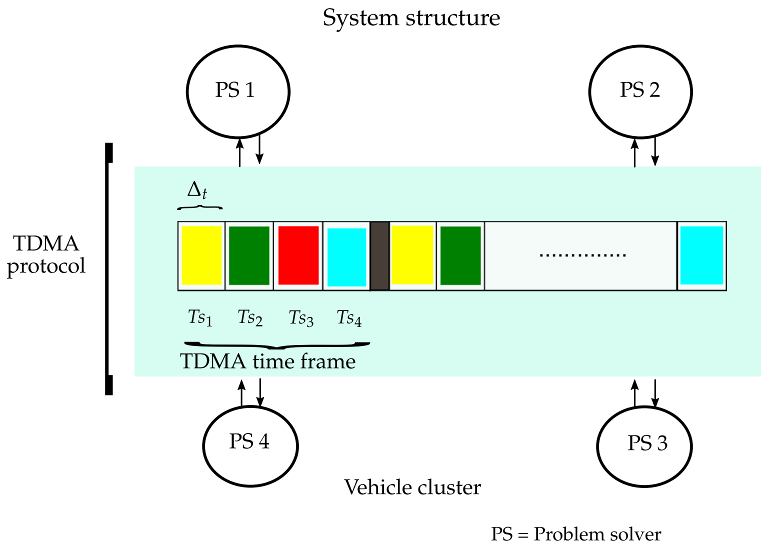

FIGURE 4.4: Event-triggered TDMA protocol structure. For the underlying application, PS1-4 represents, e.g., the ECUs mounted on the individual vehicle wheels that take over steering of the vehicle throughout a predefined maneuver.

The TDMA protocol scheduling mechanism is used to share wireless channel time equally by assigning the full frequency bandwidth to a node for a fixed time slot T_{s_i}. In more detail: a TDMA time slot is assigned to node i so that it can broadcast its state variable x_i to its neighbors $j \in \mathcal{N}_i$. Node j also updates its local copies of the variables received from node i.

4.3.1 Simulation: event-triggered solver over TDMA protocol

This section presents the simulation results of the proposed event-triggered solver considering the effect of the wireless network-based TDMA protocol and broadcasting topology. We will analyze how the convergence of the distributed optimization problem and the computed local adhesion η_i are affected by the TDMA-based wireless network. Moreover, we will test and discuss the benefits of using the event-triggered scheme in the algorithm on the basis of communication reduction and maintaining an acceptable error computation. Basically, the evaluation of the algorithm is based on the efficiency of the event-triggered scheme related to communication reduction, the relative absolute error measure of the convergence of η_i with reference to the optimal value η^* computed by means of the centralized optimization problem solver, and the decay rate of the equality constraints.

In order to test the TDMA protocol in a wireless network setup, we consider an Additive White Gaussian Noise (AWGN) wireless channel with a perfect channel state being $SNR \geq 70dB$, a good channel state being $SNR \leq 70dB$, and a bad channel state being $SNR \leq 20dB$. Each node broadcasts its state variable over the wireless channel with different SNR levels. Note that the SNR level is fixed during each broadcasting process.

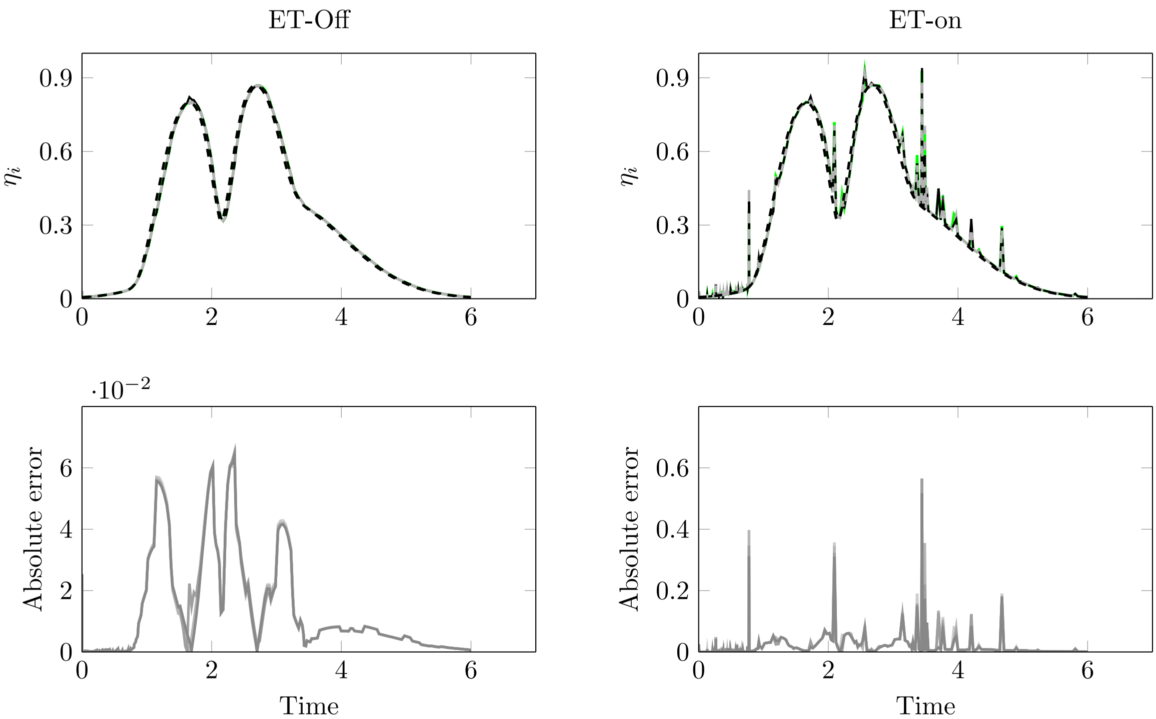

FIGURE 4.5: Adhesion η_i for complete maneuver,
a) $SNR \leq 70dB$, event-triggered = Off,
b) $SNR \leq 70dB$, event-triggered = On.

Fig 4.5 presents the computed adhesion η_i for a full lane change maneuver under braking conditions for the four nodes connected over a normal-condition wireless channel operating on $SNR \leq 70dB$. The top left of Fig 4.5-a shows the η_i computed when the event-triggered scheme was switched off, and the bottom left of Fig 4.5-a presents the absolute error. The top right of Fig 4.5-b shows the η_i computed when the event-triggered scheme was activated, and the bottom right of Fig 4.5-b presents the absolute error. It can be seen that the absolute error increased due to the effect of the decrease of the data exchanged and due to the effect of the channel state changes.

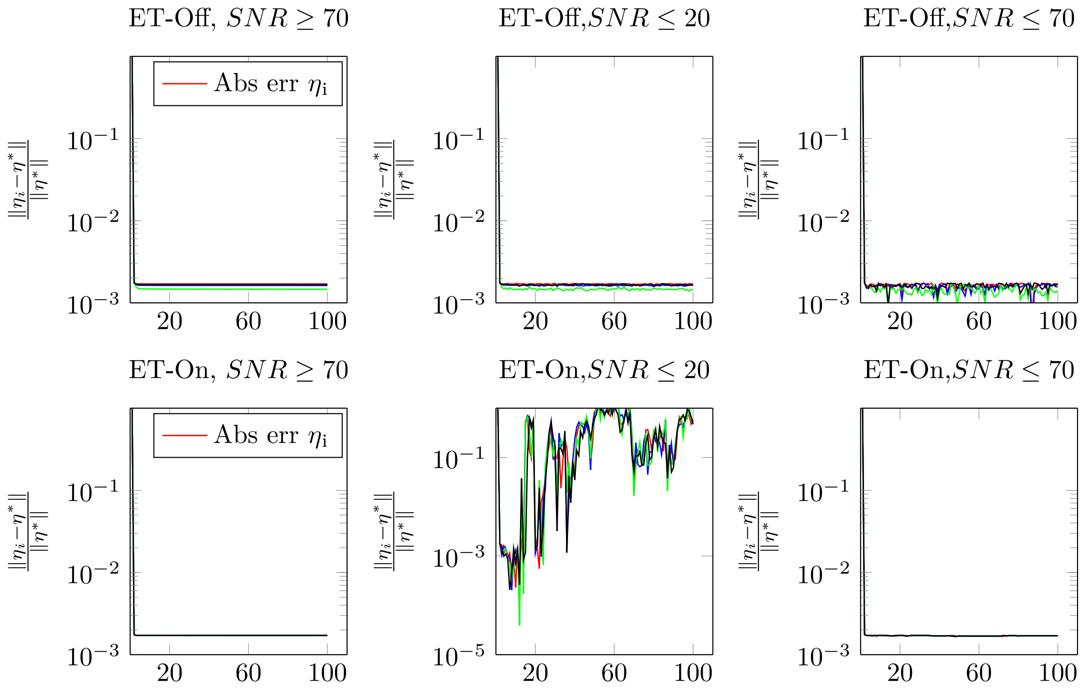

FIGURE 4.6: Relative error of the η_i convergence measure for the extreme maneuver (Problem 1-Yd=270), a) Perfect wireless channel $SNR \geq 70$ and ET-On/Off, b) Poor wireless channel $SNR \leq 20dB$ and ET-On/Off, c) Moderate wireless channel $SNR \leq 70dB$ and ET-On/Off.

Fig 4.6 presents the relative error measure of the convergence rate of the computed local adhesion η_i over the extreme maneuver instance (Problem 1 $Yd = 270$), which was tested in the case where the event-triggered scheme was active and when it was inactive. The state of the wireless channel is considered perfect with $SNR \geq 70dB$, poor with $SNR \leq 20dB$, and normal with $SNR \leq 70dB$. The top of Fig 4.6-a shows the relative error measure convergence rate of η_i when the wireless nodes communicated over a perfect wireless channel and the event-triggered scheme was active. The bottom of Fig 4.6-a shows the relative error measure convergence rate of η_i when the event-triggered scheme was inactive. It can be seen that the algorithm had almost the same performance in both cases with only a slight difference, which indicates that the proposed algorithm has good performance. The top of Fig 4.6-b presents the relative error measure convergence rate of η_i when the interconnected nodes broadcasted their state η_i, F_{xi}, F_{yi} over poor wireless channel conditions with $SNR \leq 20dB$ and the event-triggered scheme was inactive. The bottom of Fig 4.6-b presents the relative error measure convergence rate of η_i for the same $SNR \leq 20dB$ when the event-triggered scheme was switched on. Bad performance of the algorithm can be seen because the combination of the poor wireless channel state and the reduction in the number of the broadcasted nodes' states introduced error between the transmitted states at the receiver. Furthermore, the convergence rate of the η_i is acceptable when the event-triggered scheme was switched off. Fig 4.6-c presents the relative error measure convergence rate of η_i for the normal wireless network state with $SNR \leq 70dB$ in both cases, with an active event-triggered scheme and with an inactive event-triggered scheme. We observe that the relative error measure of the convergence rate of the computed η_i is almost identical, which proves that the algorithm has good performance when the wireless nodes are connected to such a channel.

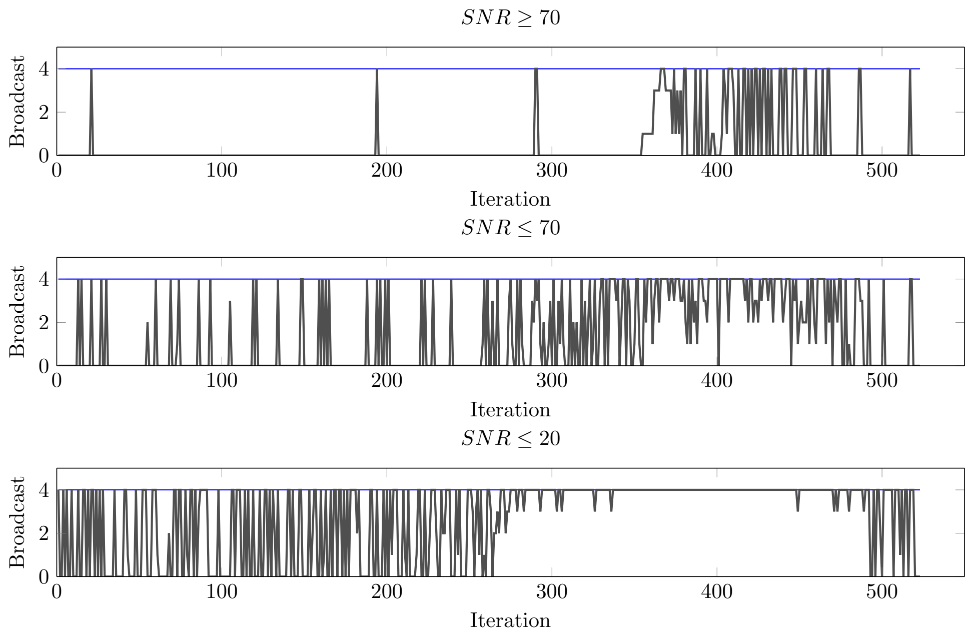

Figure 4.7: The effect of the event-triggered scheme on communication reduction,
a) Wireless channel with $SNR - Off$,
b) Wireless channel with $SNR \leq 70dB$,
c) Wireless channel with $SNR \leq 20dB$.

Fig 4.7 presents the broadcasting activities of the four interconnected nodes and how the event-triggered scheme reduces the communication requests in different wireless channel states. Basically, each node broadcasts its state variables in every iteration of the subgradient update. The total number of active nodes in the network is equal to $N = 4$ nodes, which is represented by the horizontal line at 4. We see in Fig 4.7-a that in the case of the perfect channel state, the network load is decreased by reducing the number of active nodes by 11%. Fig 4.7-b shows that the wireless network load is reduced by decreasing the number of active nodes by 33%. In the case of the bad channel state, Fig 4.7-c, the number of active nodes is decreased by 68%.

The proposed TDMA distributed event-triggered optimization algorithm leads to reduced communication between the nodes. The maximum number of broadcasts is set to be equal to the number of subgradient iterations. As stated above, the main purpose of the proposed algorithm is to minimize the broadcast activity.

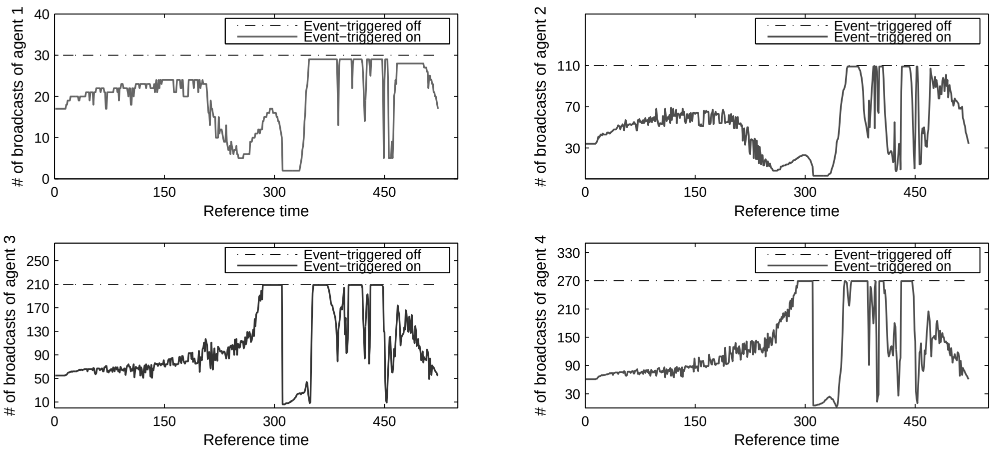

Figure 4.8: Communication reduction analysis.

Fig 4.8 depicts the nodes' communication activity over a given maneuver. Each figure refers to the absolute number of broadcasts of the respective node. The dashed horizontal line represents the maximum number of broadcasts as a reference, whereas the plots underneath indicate the actual number of broadcasts. It is the consequence of the event-triggered impact that the number of broadcasts at each node decreases dramatically. Observe, however, that at the difficult turn-points in the maneuver no communication reduction is achieved. In general, we conclude that the proposed algorithm has reduced the number of broadcasts by 40% compared to the total number of broadcasts. It can also be seen that in the simple maneuver (problem 3), the number of broadcasts was reduced by 60%. We can conclude that the distributed event-triggered algorithm is efficient in terms of speed of convergence, communication reduction, and performance.

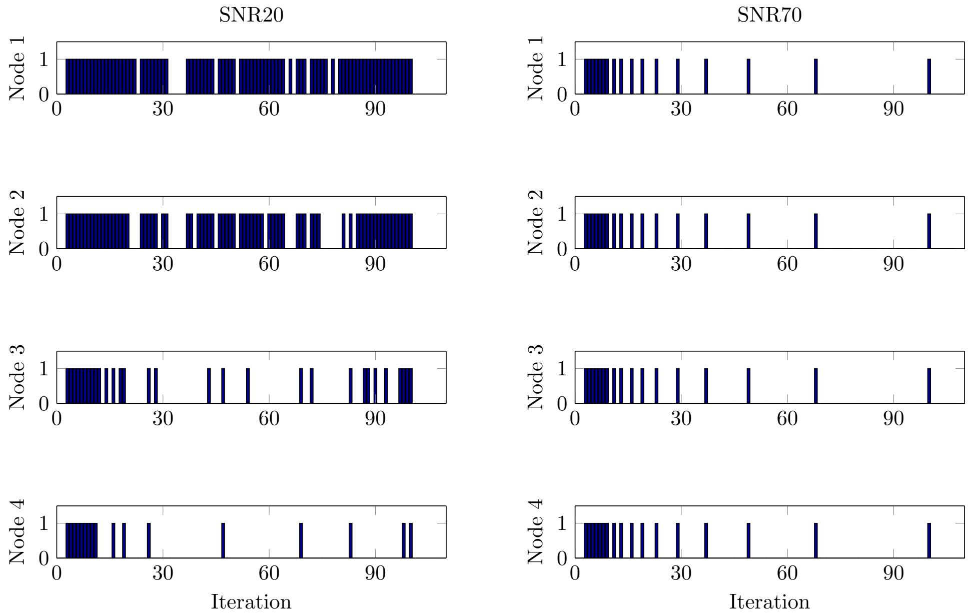

FIGURE 4.9: TDMA scheme communication reduction,
a) Communication activities with channel state $SNR \leq 20dB$,
b) Communication activities with channel state $SNR \leq 70dB$.

Fig 4.9 presents the communication activities of the interconnected nodes $i = 1$,$i = 2$, $i = 3$, and $i = 4$ of the maneuver instance (Problem 1, Yd=290). Fig 4.9-a shows the communication between the four nodes for the same problem with a bad wireless channel state $SNR \leq 20dB$ while Fig 4.9-b shows the communication between the four nodes for the same problem but with a good wireless channel state $SNR \leq 70dB$. We observe that the node activities dramatically decreased when the channel state was better. Also, some of the nodes were more active than the others in the case of the bad channel state.

4.4 Chapter conclusion

The introduced event-triggered layer was implemented in order to reduce communication between the control nodes, maintain the performance of the system, and guarantee convergence of the algorithm to some near-optimal solution. The simulation results showed that a communication reduction of up to 40% was achieved. We believe that this venture represents an important step towards applying our optimization algorithms in real time.

Initialization;
set $\mathcal{N}_i = \{j, j \in M\}$, $i = \{1, ..., N\}$;
set $k = 1$; $\ell_i = 1$;
set $S =$ Subgradient number;
broadcast η_i, F_{xi}, F_{yi}, $\forall\ j \in \mathcal{N}_i$;
while $k <= S$ **do**

Node;

Update subgradient state x_i;

$$\eta_i[k+1] \leftarrow \eta_i[k](1-2\alpha),$$
$$F_{xi}[k+1] \leftarrow F_{xi}[k](1-2\epsilon\alpha),$$
$$F_{yi}[k+1] \leftarrow F_{yi}[k](1-2\epsilon\alpha),$$

Event-triggered condition;

$$\begin{aligned}
\left\| F_{xi}[k+1] - \hat{F}_{xi}[\ell_i] \right\| &\geq \beta_1 \left\| h_1[k] \right\| + \beta_2 \\
\left\| F_{yi}[k+1] - \hat{F}_{yi}[\ell_i] \right\| &\geq \beta_1 \left\| h_1[k] \right\| + \beta_2 \\
\left\| \eta_i[k+1] - \hat{\eta}_i[\ell_i] \right\| &\geq \beta_1 \left\| h_1[k] \right\| + \beta_2
\end{aligned}$$

Wireless channel TDMA communication protocol;

$$Tx_i \leftarrow T_{si};$$

Broadcast

η_i, F_{xi}, F_{yi}, $\forall\ j \in \mathcal{N}_i$;

set:

$$\hat{F}_{xi}[\ell_i] \leftarrow F_{xi}[k]$$
$$\hat{F}_{yi}[\ell_i] \leftarrow F_{yi}[k]$$
$$\hat{\eta}_i[\ell_i] \leftarrow \eta_i[k]$$

Receive

Node $j \in \mathcal{N}_i$ If no time slot assigned then

Receive $\hat{\eta}_j[k]$, $\hat{F}_{xj}[k]$, $\hat{F}_{jy}[k]$, $\forall\ j \in \mathcal{N}_i$;

Average consensus;

$$\eta_i^{(c+1)}[k] \leftarrow \eta_i^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{\eta}_j[k] - \eta_i^{(c)}[k]\right)$$
$$F_{xi}^{(c+1)}[k] \leftarrow F_{xi}^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{F}_{xj}[k] - F_{xi}^{(c)}[k]\right)$$
$$F_{yi}^{(c+1)}[k] \leftarrow F_{yi}^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{F}_{jy}[k] - F_{yi}^{(c)}[k]\right)$$

Projection;

$$\begin{aligned}
\eta_i &\leftarrow Proj_{\boldsymbol{\chi}^i}(\eta_i[k]) \\
F_{xi} &\leftarrow Proj_{\boldsymbol{\chi}^i}(F_{xi}[k]) \\
F_{yi} &\leftarrow Proj_{\boldsymbol{\chi}^i}(F_{yi}[k])
\end{aligned}$$

increment k;

end

Algorithm 2: TDMA event-triggered distributed projected subgradient with average consensus algorithm

Chapter 5

Sensitivity-based event-triggered TDMA protocol

In section 3.5.4, we derived a primal subgradient algorithm to solve the optimal control problem (3.30). In general, the implementation of the primal subgradient method does not consider the equality and inequality constraints in updating the primal variables. Therefore, in this chapter, we will apply the dual decomposition method on the optimization problem and decompose it to a distributed form. Following that, we will implement a dual subgradient algorithm to solve the decomposed problems. Mainly, we will upgrade the event-based communication algorithm developed in the previous chapter by adding a sensitivity analysis layer, which is used to measure the effect of the transmitted node state on its neighbors' solution of their associated sub-problems. Where, a sensitivity-based event-triggered scheme will be added to the TDMA scheduler to handle the assignment of the time slots with respect to the nodes' effect on the neighbors' sub-problem solution.

This chapter is organized as follows. In section 5.1, we will introduce the dual subgradient method, while section 5.2 presents the implementation of the dual subgradient method to solve the vehicle dynamics problem, in both centralized and distributed form. Section 5.3 introduces the dual event-based communication scheme. In section 5.4, the sensitivity analysis of the underlying optimization problem will be discussed. In section 5.5, we will introduce the adaptive event-based TDMA protocol based on sensitivity analysis. We will end the chapter with a simulation and discussion in section 5.6, followed by the chapter conclusion in section 5.7.

5.1 Dual subgradient method

In order to solve the optimization problem (3.5) using the dual subgradient method, we need to formulate the dual optimization problem, which involves the derivation of the Lagrangian function $L_i(x_i, \lambda_p^i, \nu_q^i)$ of the optimization problem. In general, the Lagrangian L takes the constraints into account by combining the cost function f_i^0 with a weighted sum of the equality h_q^i and inequality f_p^i constraints [47]. We recall the equation (3.6) which formulates the Lagrangian function as follows:

$$L_i(x_i, \lambda_p^i, \nu_q^i) = f_0^i(x_i) + \sum_{p=1}^{P} \lambda_p^i f_p^i(x_i) + \sum_{q=1}^{Q} \nu_q^i h_q^i(x_i). \tag{5.1}$$

where $\lambda_p^i = \left\{\lambda_1^i, \lambda_2^i, ..., \lambda_P^i\right\}$ and $\nu_q^i = \left\{\nu_1^i, \nu_2^i, ..., \nu_Q^i\right\}$ are the Lagrange multipliers associated with inequality and equality constraints of the sub-problem i, respectively. The corresponding dual objective function g_i is defined as follows:

$$g_i(\lambda_p^i, \nu_q^i) = \inf_{x_i} L_i(x_i, \lambda_p^i, \nu_q^i) = \inf_{x_i} \sum_{i}^{N} f_0^i(x_i) + \sum_{p}^{P} \lambda_p^i f_p^i(x_i) + \sum_{q}^{Q} \nu_q^i h_q^i(x_i). \tag{5.2}$$

The solution of the dual function $g(\lambda_p^i, \nu_q^i)$ is defined as the minimum value of the Lagrangian L over the primal variable x_i. The corresponding dual optimization problem is written as follows:

$$\begin{aligned} \text{maximize} \quad & g_i(\lambda_p^i, \nu_q^i) \\ \text{subject to} \quad & \lambda_p^i \geq 0. \end{aligned} \tag{5.3}$$

The dual subgradient method [93] is used to solve the dual problem (5.3), where the dual variables λ_p^i and ν_q^i are updated in an iterative manner as follows:

$$\begin{aligned} \lambda_p^i[k+1] &= (\lambda_p^i[k] - \alpha f_p^i[k])_+, \\ \nu_q^i[k+1] &= \nu_q^i[k] - \alpha h_q^i[k], \end{aligned} \tag{5.4}$$

where k is the subgradient iteration counter and α is the step size. To satisfy the inequality constraint $\lambda_p^i \geq 0$, a projection of λ_p to the zero value using $(\lambda_p^i)_+ = \max(0, \lambda_p^i)$ is applied if $\lambda_p^i \leq 0$, and the subgradients $f_p^i[k]$, and $h_q^i[k]$ are updated as follows:

$$\begin{aligned} f_p^i[k] &= -f_p^i(x_i^*[k]), \\ h_q^i[k] &= -h_q^i(x_i^*[k]), \end{aligned} \tag{5.5}$$

where the right-hand side argument $x_i^*[k] = x_i^*(\lambda_p^i[k], \nu_q^i[k])$ refers to the current optimal value of the primal variables at time instant k, i.e.:

$$x_i^* = \arg\inf_{x_i} L(x_i, \lambda_p^i[k], \nu_q^i[k]). \tag{5.6}$$

Here we see that the updated dual variables $\lambda_p^i[k]$, and $h_q^i[k]$ are used to update the optimal primal variables x_i^*.

5.2 Dual subgradient algorithm for solving the vehicle dynamics optimal control problem

Recall the vehicle dynamics optimal control problem (3.30), whose objective is to achieve the smallest possible utilization of the adhesion potential η_i while respecting the adhesion limit, which was defined as follows:

$$\begin{aligned} \underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \quad & J_0 = \sum_{i=1}^{N} \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right), \\ \text{subject to} \quad & f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f \leq 0, \\ & f_2 = \eta_i - \mu_{\max} \leq 0, \\ & h_1 = A_x F_x + A_y F_y - Y_d = 0, \end{aligned} \tag{5.7}$$

where, η_i, F_{xi} and F_{yi} are considered as decision variables, and $\epsilon \ll 1$ is a regularization term of the optimization problem. The A_x and A_y matrices are defined by the vehicle's geometric parameters, and Y_d is the reference trajectory. $\hat{F}_{xj} = (F_{x1}, \ldots, F_{x4})^T$ and $\hat{F}_{yj} = (F_{y1}, \ldots, F_{y4})^T$ are the longitudinal and the lateral forces vectors, respectively. N_f is the normal force and $\mu_{\max}$ is the maximum friction coefficient parameter.

5.2.1 Formulation of the centralized dual problem

Basically, the formulation of the centralized dual optimization problem is based on deriving the Lagrangian L of the primal optimization problem (5.7), which reads as follows:

$$\begin{aligned} L(\eta, F_x, F_y, \lambda, \sigma, \nu) = \sum_{i=1}^{N} \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right) + \sum_{i=1}^{N} \lambda(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f) + \\ \sum_{i=1}^{N} \sigma(\eta_i - \mu_{\max}) + \nu^T (A_x F_x + A_y F_y - Y_d), \end{aligned} \tag{5.8}$$

with $\eta = (\eta_1, \ldots, \eta_4)$, $F_x = (F_{x1}, \ldots, F_{x4})^T$, and $F_y = (F_{y1}, \ldots, F_{y4})^T$ being primal variables, and $\lambda = (\lambda_1, \ldots, \lambda_4), \sigma = (\sigma_1, \ldots, \sigma_4)$, and $\nu = (\nu_1, \nu_2, \nu_3)$ being the dual variables associated with the inequality $f_1$and f_2 and equality constraints h_1, respectively. The corresponding dual function is given by:

$$\begin{aligned} g_i(\lambda_i, \sigma, \nu) = \inf_{\eta_i, F_{xi}, F_{yi}} \big(\eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right) + \lambda(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f) + \\ \sigma \eta_i + \nu^T (A_x F_x + A_y F_y)\big) - \nu_i^T Y_d - \sigma \mu_{\max}, \end{aligned} \tag{5.9}$$

Invoking the dual optimization problem:

$$\begin{aligned} \text{maximize} \quad & g(\lambda_i, \sigma, \nu) \\ \text{subject to} \quad & \lambda \geq 0, \\ & \sigma \geq 0. \end{aligned} \tag{5.10}$$

We use the subgradient method to solve the dual optimization problem (5.10) and perform the subgradient update of the dual variables λ, σ and ν as follows:

$$\begin{aligned}
\lambda_i[k+1] &= (\lambda[k] - \alpha f_1[k])_+, \\
\sigma[k+1] &= (\sigma[k] - \alpha f_2[k])_+, \\
\nu^T[k+1] &= \nu^T[k] - \alpha h_1[k].
\end{aligned} \tag{5.11}$$

The subgradient of $f_1[k], f_2[k]$, and $h_1[k]$ are defined as:

$$\begin{aligned}
f_1[k] &= -f_1(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \\
f_2[k] &= -f_2(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \\
h_1[k] &= -h_1(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]),
\end{aligned} \tag{5.12}$$

where, as it turns out, the expressions on the right represent the analytical solutions to $(\eta_i^*, F_{xi}^*, F_{yi}^*) = \arg\inf_{\eta_i, F_{xi}, F_{yi}} L(\eta_i, F_{xi}, F_{yi}, \lambda[k], \sigma[k], \nu[k])$, which is computed as follows:

$$\begin{aligned}
\eta_i^*(\lambda, \sigma) &= \frac{1}{2}\left(\lambda N_f - \sigma\right), \\
F_{xi}^*(\lambda, \nu) &= \frac{\kappa_{xi}}{2\epsilon^2}\left(\frac{\lambda}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right), \\
F_{yi}^*(\lambda, \nu) &= \frac{\kappa_{yi}}{2\epsilon^2}\left(\frac{\lambda}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right),
\end{aligned} \tag{5.13}$$

with

$$\kappa_{xi} = \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}, \qquad \kappa_{yi} = \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}. \tag{5.14}$$

We solve the dual optimization problem (5.10) by implementing the dual subgradient algorithm. The complete centralized dual subgradient solution algorithm is illustrated in Algorithm (4).

The performance of the centralized dual subgradient algorithm is shown in the following figures, where we plot the output of the algorithm's primal variable η_i and the dual parameters ν_i, which represent the convergence of the equality constraints h_1. The left side of Fig. 5.1 shows the optimal value of η_i, $i = 1, \dots, 4$ and the dual variables ν for a difficult maneuver instance (Yd=300).

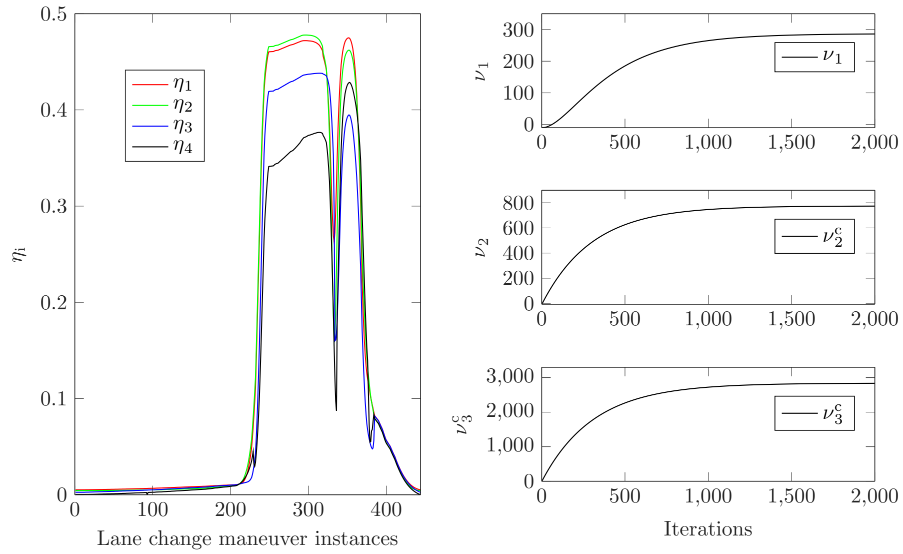

FIGURE 5.1: The performance of the centralized dual algorithm includes η_i and ν for maneuver instance Yd=300.

5.2.2 Formulation of the distributed dual problem

We see that the optimization problem (5.7) is separable in the cost function f_0 and in the inequality constraints f_1 and f_2 because they depend only on the local primal variables $\eta_i, F_{xi}, F_{yi}, \lambda_i, \sigma_i$, and that the equality constraints $h_1 = A_x\hat{F}_{xj} + A_y\hat{F}_{yj} - Y_d = 0$ couple the N sub-problems because they depend on the longitudinal forces F_{xj} and the lateral forces F_{yj} of the neighbors' sub-problems $j \in \mathcal{N}_i$. This implies that the dual variables $\nu = (\nu_1, \nu_2, \nu_3)$ also couple the dual problem and need to be exchanged between the sub-problems in order to achieve an optimal solution of the overall optimization problem at each sub-problem. Therefore, we break the optimization problem (5.7) down into N sub-problems by first extracting the problem into N sub-problems and keeping a copy of each global variable ν, $\hat{F}^i_{xj}$, and $\hat{F}^i_{yj}$ at each sub-problem i, and then defining the distributed optimization sub-problems i as follows:

$$\begin{aligned} \underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \quad & f_0^i = \eta_i^2 + \epsilon^2\left(F_{xi}^2 + F_{yi}^2\right), \\ \text{subject to} \quad & f_1^i = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i \leq 0, \\ & f_2^i = \eta_i - \mu_{\max} \leq 0, \\ & h_1^i = A_x\hat{F}^i_{xj} + A_y\hat{F}^i_{yj} - Y_d = 0. \end{aligned} \tag{5.15}$$

Observe that $\hat{F}^i_{xj}$, $\hat{F}^i_{yj}$ refer to the local copies at node i. Here we apply dual decomposition by introducing the Lagrangian L_i of sub-problem i as:

$$\begin{aligned} L_i(\eta_i, F_{xi}, F_{yi}, \lambda_i, \sigma_i, \nu_i) = {} & \eta_i^2 + \epsilon^2\left(F_{xi}^2 + F_{yi}^2\right) + \lambda_i(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i) + \\ & \sigma_i(\eta_i - \mu_{\max}) + \nu_i(A_x\hat{F}^i_{xj} + A_y\hat{F}^i_{yj} - Y_d), \quad j \in \mathcal{N}_i, \end{aligned} \tag{5.16}$$

where we introduce a local copy ν_i of the global dual variable ν to each sub-problem i. Herein, the dual variables λ_i and σ_i are considered as local dual variables because their updating depends only on the local primal variables of sub-problem i. The corresponding dual function is given by:

$$\begin{aligned} g_i(\lambda_i, \sigma_i, \nu_i) = \inf_{\eta_i, F_{xi}, F_{yi}} \big(\eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right) + \lambda_i(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i) + \\ \sigma_i \eta_i + \nu_i (A_x \hat{F}_{xj} + A_y \hat{F}_{yj})\big) - \nu_i Y_d - \sigma_i \mu_{\max}, \end{aligned} \tag{5.17}$$

Invoking the distributed dual optimization sub-problem:

$$\begin{aligned} \text{maximize} \quad & g_i(\lambda_i, \sigma_i, \nu_i) \\ \text{subject to} \quad & \lambda_i \geq 0, \\ & \sigma_i \geq 0. \end{aligned} \tag{5.18}$$

Applying the subgradient method to solve the dual optimization sub-problem (5.18) and using it to update the local dual variables λ_i, σ_i, and the copy of the global variable ν_i, we get:

$$\begin{aligned} \lambda_i[k+1] &= (\lambda_i[k] - \alpha f_1^i[k])_+, \\ \sigma_i[k+1] &= (\sigma_i[k] - \alpha f_2^i[k])_+, \\ \nu_i[k+1] &= \nu_i[k] - \alpha h_1^i[k], \ . \end{aligned} \tag{5.19}$$

The subgradients with respect to the inequality constraints $f_1[k], f_2[k]$, and the equality constraint $h_1^i[k]$ are defined as:

$$\begin{aligned} f_1^i[k] &= -f_1^i(\eta^*[k], F_x^*[k], F_y^*[k]), \\ f_2^i[k] &= -f_2^i(\eta^*[k], F_x^*[k], F_y^*[k]), \\ h_1^i[k] &= -h_1^i(\eta^*[k], F_x^*[k], F_y^*[k]), \end{aligned} \tag{5.20}$$

where solving the dual problem analytically provides the following analytical solution expressions for updating the optimal primal variables η_i^*, F_{xi}^* and F_{yi}^*:

$$\begin{aligned} \eta_i^*(\lambda_i, \sigma_i) &= \frac{1}{2}\left(\lambda_i N_f^i - \sigma_i\right), \\ F_{xi}^*(\lambda_i, \nu_i) &= \frac{\kappa_{xi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right), \\ F_{yi}^*(\lambda_i, \nu_i) &= \frac{\kappa_{yi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right), \end{aligned} \tag{5.21}$$

with

$$\kappa_{xi} = \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}, \qquad \kappa_{yi} = \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}. \tag{5.22}$$

As a result of the distributed form of the problem setup, and the introduction of the local copies of the global variables ν at each sub-problem i, we need to involve an exchanging mechanism for the local copies $\nu_i = (\nu_i^1, \nu_i^2, \nu_i^3)$ between the associated sub-problems to guarantee that the ν_i copies are identical. To this end, we invoke an average consensus algorithm to compute the average value of the received $\hat{\nu}_j$ and the local dual variables ν_i. The average consensus algorithm [64, 65] is executed after each subgradient iteration by performing the following operation:

$$\nu_i^{(c+1)}[k] = \nu_i^{(c)}[k] + \frac{1}{N} \sum_{j \in \mathcal{N}_i} w_{ij} \left(\hat{\nu}_j[k] - \nu_i^{(c)}[k] \right), \qquad \forall i \in \mathcal{N}. \tag{5.23}$$

Here we implement a distributed dual subgradient solution algorithm and invoke a communication protocol to exchange the information between the solvers. The complete distributed dual subgradient solution algorithm is illustrated in the Algorithm (5).

In order to ensure the solver performance, we evaluate the performance of the distributed dual subgradient algorithm, where we compare its performance with the centralized dual subgradient algorithm. The top right of Fig. 5.2 shows the distributed optimal primal variables η_i, $\{i = 1, \ldots, 4\}$, and the bottom-right figure presents the absolute error compared with η_i computed by the centralized algorithm. Notice that the absolute error is noticeably increased when the vehicle performs a lane change maneuver.

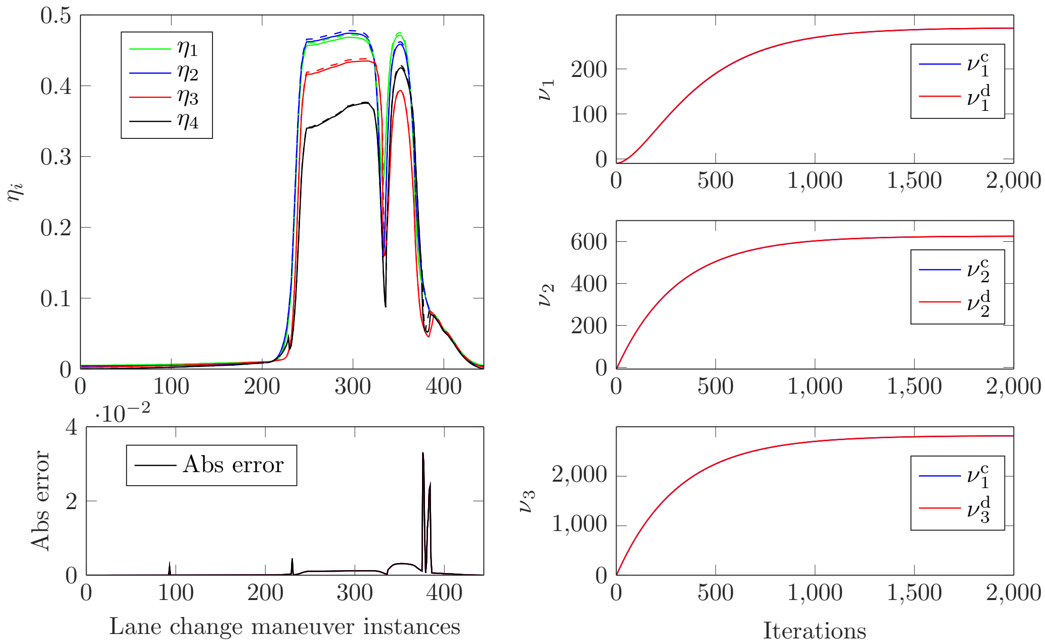

FIGURE 5.2: The performance of the distributed dual subgradient solver. a) distributed primal η_i, b) absolute error compared to the centralized algorithm, c) comparison of both distributed and centralized η.

The left panel of the figure presents the local copy of the distributed dual variable ν_i^d

compared with the equivalent centralized dual variable ν_i^c. We see that there is quite a difference between the distributed and the centralized variables, which is due to the effect of the communication and consensus algorithm.

5.3 Event-based communication

In the context of a distributed wireless networked control system, the efficient use of the communication resources is highly important to guarantee the system's performance. However, typical time-driven control policies are independent of the system's state evolution and communication protocols, which may result in unnecessarily high utilization of communication resources [94]. Therefore, introducing an event-based mechanism in the distributed system structure results in a purposeful reduction of the communication resources in accordance with the dynamic evolution of the local subsystem state. Underlying this approach is always the hope that a certain loss of system performance is justifiable by an essential reduction of the communication effort, where in a large-scale wireless distributed system, an event-based mechanism triggers the communication process if the local state error deviates beyond a predefined threshold [86].

5.3.1 Event-based communication for the dual algorithm solver

Recall that the distributed decomposition of the optimization problem (5.15) introduces local copies of the global dual variables ν at every node $i = 1, \ldots, N$. Additionally, in the solution algorithm, a communication scheme is introduced for exchanging the copies ν_i between the nodes. Then the average consensus (5.23) is applied to guarantee that all the copies are identical. In order to reduce the consumption of communication resources, an event-based communication protocol is defined with respect to the evolution of the node's local state, where the event-triggered condition traces the difference between the most recently updated variable and the last transmitted variable. Intuitively, the data (e.g., the ν_i variables) is exchanged only if it possesses a sufficiently high level of novelty compared to the last information transmission. In this sense, exchanging the dual variables is controlled by the event-triggered condition, which computes the difference between the most recently updated ν_i and the last transmitted $\hat{\nu}_i$. The event-triggered condition reads:

$$\|\nu_i[k+1] - \hat{\nu}_i[k]\| \geq \beta_0 \left\|h^i[k]\right\| + \beta_1, \quad i \in \mathcal{N} \tag{5.24}$$

i.e., the data (ν_i, F_{xi} and F_{yi} variables) is broadcasted from node i if the latter condition holds true; otherwise no transmission takes place. The overall algorithm now involves an iteration of three sub-sequential steps: subgradient updates, event-triggered data exchange, and consensus, which is formulated in the algorithm 3. Note that this algorithm will serve as the reference to the scheme introduced in the next section.

The following figures demonstrate the performance of the distributed dual event-based algorithm compared to the centralized dual solver. We consider the same lane change maneuver setup, and the parameters $\beta_0 = 0.003$ and $\beta_1 = 0.00023$ for the event-triggered condition.

Fig. 5.3 shows the computed η_i of the four wheels, and the absolute error $\eta_i - \eta^*$,

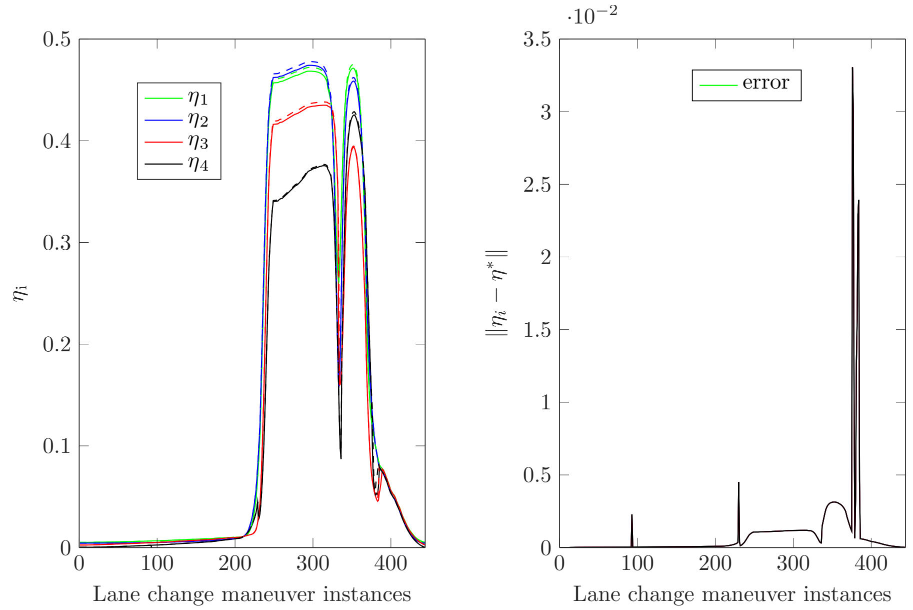

FIGURE 5.3: Performance of the dual algorithm ν.

In Fig. 5.4, the evolution of the local dual variables ν_i^d for a difficult maneuver instance (Yd=300) is illustrated in comparison with the global variables ν_i^c within a fixed subgradient iteration = 150 (top), and for a subgradient iteration = 2000 (bottom) to check the performance of the algorithm until the Lagrange multiplier ν_i reaches its final value.

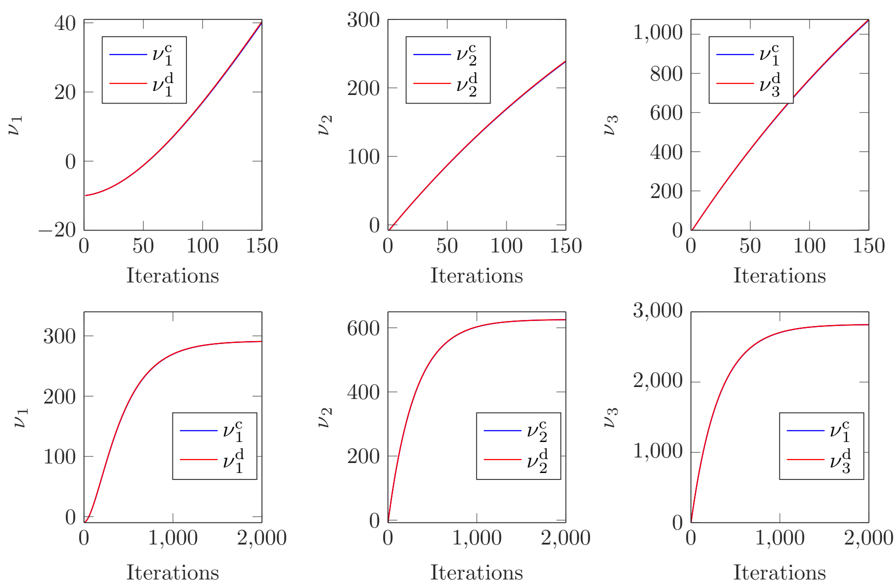

FIGURE 5.4: Performance of the dual algorithm ν.

5.4 Sensitivity analysis for event-based communication

In general, conventional event-based communication considers local changes of a node and based on that decides whether to transmit its state or not. We notice that the local decision considers only the effect on the node's local state without looking at the global effect of the node's state transmission, although nodes do exchange their local information, provided each node is globally informed about the state of its neighbors. Henceforth, we will consider the information available at each node and perform a sensitivity analysis of the effect of the updated node's state on its neighbors' state, and utilize this as part of the event-based communication scheme.

5.4.1 Sensitivity analysis

Sensitivity analysis is mainly used to quantify variations of the optimal problem solution with respect to changes of some parameters [52], and depends on the computation of the derivatives of the output with respect to the independent input variables. Basically, sensitivity analysis depends on the Lagrangian of the optimization problem and use of Lagrange multipliers to identify the sensitivity information related to constraint deviations with respect to a perturbation parameter [95]. To demonstrate the sensitivity analysis of the optimization problem with respect to a perturbation parameter, we consider the following adaptation of the standard optimization problem (3.1):

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_0(x_i, \varepsilon_j) \\ \text{subject to} \quad & f_p^i(x_i, \varepsilon_j) \leq 0, \; p = 1, \dots, P_i, j \in \mathcal{N}_i \\ & h_q^i(x_i, \varepsilon_j) = 0, \; q = 1, \dots, Q_i, j \in \mathcal{N}_i, \end{aligned} \tag{5.25}$$

where x_i is the local state and ε_j is the perturbation variable of sub-problem i received from its neighbors' sub-problems $j \in \mathcal{N}_i$. To emphasize, the local cost function f_0 is updated based on its own state x_i and is perturbed by the ε_j received from its neighbors $\mathcal{N}_i$. Here, we consider the effect of parameter ε_j on the optimal solution of sub-problem i by computing the sensitivity of the optimal solution with respect to the changes in ε_j. Following the work [95, 96], we formulate the sensitivity analysis, first deriving the Lagrangian of the optimization problem (5.25) as follows:

$$L_i(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i) = f_0(x_i, \varepsilon_j) + \lambda_p^i f_p^i(x_i, \varepsilon_j) + \nu_q^i h_q^i(x_i, \varepsilon_j), \tag{5.26}$$

Recall that λ_p^i and ν_q^i are the Lagrange multipliers associated with the equality and inequality constraints, respectively. Here, the state sensitivity defines the dependency of the state x_i on the received neighbor ε_j. In other words, the sensitivity is computed based on the Jacobian matrices $\nabla_{x_i} L_i$ and $\nabla_{\varepsilon_j} L_i$ of the Lagrangian L_i, which is defined as:

$$\dot{S}(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i) = \nabla_{x_i} L_i(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i) S(.) + \nabla_{\varepsilon_j} L_i(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i), \tag{5.27}$$

where $\nabla_{x_i} L_i$ and $\nabla_{\varepsilon_j} L_i$ are the Jacobian matrices of (5.26) with respect to the local variables x_i and the received variables ε_j computed as follows:

$$\begin{aligned} \nabla_{x_i} L_i &= \frac{\partial L_i(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i)}{\partial x_i}, \\ \nabla_{\varepsilon_j} L_i &= \frac{\partial L_i(x_i, \varepsilon_j, \lambda_p^i, \nu_q^i)}{\partial \varepsilon_j}, \end{aligned} \tag{5.28}$$

and

$$S(x_i, \varepsilon_j) = \frac{\partial x_i}{\partial \varepsilon_j}, \tag{5.29}$$

where x_i and ε_j are vectors consisting of the element of the state of node i and the perturbation variables received from node j, respectively. The sensitivity function S provides the first-order estimates of the effect of the received parameter ε_j variations on the state x_i.

Since we solve the dual optimization problem using the subgradient method and update the dual Lagrange variables continuously, refer to (5.4) for more details. Next, we rewrite the subgradient update of the dual variables λ_p^i, ν_q^i with respect to the perturbation variable ε_j as follows:

$$\begin{aligned} \lambda_p^i[k+1] &= \left(\lambda_p^i[k] - \alpha \cdot f_p^i(xi[k], \varepsilon_j[k])\right)_+, \\ \nu_q^i[k+1] &= \left(\nu_q^i[k] - \alpha \cdot h_q^i(xi[k], \varepsilon_j[k])\right). \end{aligned} \tag{5.30}$$

Here, we see the dependency between updating the dual variables λ_p^i and ν_q^i and the gradient of the constraints associated with each of them, which can be seen in the following:

$$\begin{aligned} f_p^i[k] &= -f_p^i(x_i^*[k], \varepsilon_j), \\ h_q^i[k] &= -h_1^i(x_i^*[k], \varepsilon_j). \end{aligned} \tag{5.31}$$

Moreover, the effect of the perturbation variables can also be seen in the consensus algorithm introduced into the subgradient solver, where the exchanged global Lagrange multiplier implicitly affects the local optimal solution of the sub-problem i by affecting the local dual variables used to update the primal solution. In this context, the consensus-based perturbation can be seen as follows:

$$\nu_q^i[k+1] = \frac{1}{N}\left(\nu_q^i[k] - \alpha \cdot h_i(x_i^*, \varepsilon_j) + \nu_q^j[k]\right),$$

where both the consensus variables ν_q^j and the perturbation parameters ε_j have a direct effect on the local dual variable ν_i. Given that the optimal solution x_i^* depends explicitly on the dual variables λ_p^i and ν_q^i, we can formally write the above difference equation as:

$$\bar{\nu}_q^i[k+1] = \chi^i(\nu_q^i[k], \nu_q^j[k], \varepsilon_j[k]), \tag{5.32}$$

where χ^i is inferred by the latter equation after substituting the symbolic expressions for the optimal solutions x_i^*. The effect of the received information ε_j and ν_q^j in the evolution of ν_i^q is computed according to the sensitivity analysis [96], and we define the sensitivity-based difference equation by:

$$S^i_{\varepsilon_j}[k+1] = A^i \, S^i_{\varepsilon_j}[k] + B^i_{\varepsilon_j}, \tag{5.33}$$
$$S^i_{\nu_j}[k+1] = A^i \, S^i_{\nu_j}[k] + B^i_{\nu_j}, \tag{5.34}$$

where

$$S^i_{\varepsilon_j} = \frac{\partial \bar{\nu}^i_q}{\partial \varepsilon_j}. \tag{5.35}$$
$$S^i_{\nu_j} = \frac{\partial \bar{\nu}^i_q}{\partial \nu_j}. \tag{5.36}$$

stands for the sensitivity matrices of ν_i w.r.t. ε_j and ν_j, respectively, and

$$A^i = \frac{\partial \chi^i}{\partial \nu_i}, \quad B^i_f = \frac{\partial \chi^i}{\partial \varepsilon_j}, \quad \text{and} \quad B^i_\nu = \frac{\partial \chi^i}{\partial \nu_j}, \tag{5.37}$$

where A^i, B^i_f, and B^i_ν are the Jacobean matrices of the subgradient update (5.32) with respect to ν_i, ε_j, and ν_j, respectively. Given that, the effect of the variables received from the neighbors j is encapsulated in the sensitivity matrices $S^i_{\varepsilon_j}$ and $S^i_{\nu_j}$.

5.5 Adaptive event-based TDMA protocol based on sensitivity analysis

In order to overcome the extensive consumption of communication resources of the distributed solution solver, we now devise an algorithm for the design of an adaptive TDMA event-based communication protocol with respect to the sensitivity analysis of the data exchanged among the connected nodes.

5.5.1 System structure: Adaptive event-based TDMA scheduler

We consider the following setup: a wireless networked system consisting of $\mathcal{N} = \{1, \ldots, N\}$ nodes connected through a TDMA protocol [97]. Each node i is equipped with a local solver of the optimization problem and consists of a dual subgradient algorithm and an event-triggered scheme that moderates the event-based communication policy. The event-based communication policy links both the problem solver (PS) and the TDMA scheduler, and regulates the nodes' transmission requests with respect to the sensitivity criterion. The event-triggered scheme provides the TDMA scheduler with nodes that fulfill only the event-triggered condition according to which the scheduler divides the channel time. More specifically, we consider the system depicted in Fig. 5.5, which consists of a wireless network operating based on the TDMA protocol and consisting of $N = 4$ nodes forming one vehicle cluster.

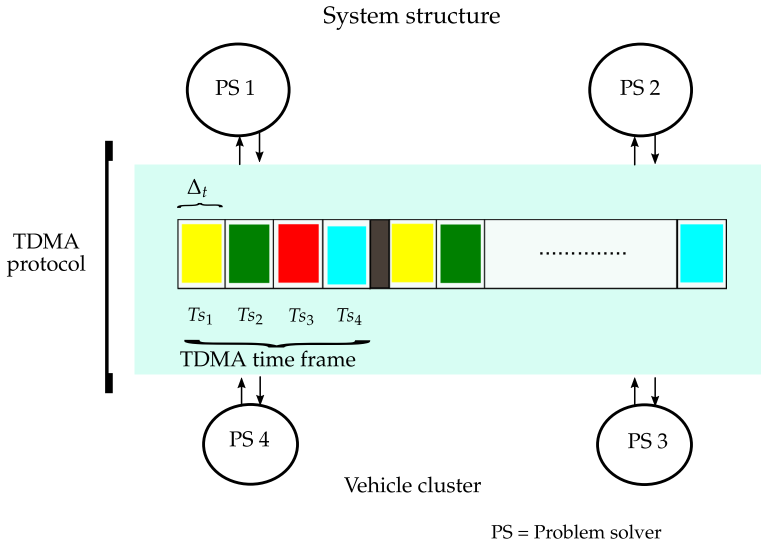

FIGURE 5.5: Event-triggered TDMA protocol structure. For the underlying application, PS1-4 represents, e.g., the ECUs mounted on the individual vehicle wheels that take over the steering of the vehicle throughout a predefined maneuver.

If the event-triggered condition (5.24) is fulfilled, the node activates the transmission request RTS_i signal, thus acquiring a time slot in the next frame. Then the TDAM scheduler adapts the number of time slots within the TDMA frame based on the number of transmission requests, and also regulates the channel utilization time assigned within each slot. Therefore, the assignment of time slots is done according to the node transmission request that is activated by the event-based communication scheme.

5.5.2 Application of the sensitivity analysis to the optimal vehicle dynamics problem

Here, we consider the information exchange of the force vectors F_{xj}, F_{yj} and the ν copies between the nodes as "perturbing parameters" of the iterative optimization difference equation inferred by the subgradient algorithm. We compute the sensitivity of the local variables ν_i with respect to the variables ν_j, F_{xj} and F_{yj} received from neighbor j. According to the explanation in the previous section, these variables explicitly affect the evolution of ν_i. The force components F_{xj} and F_{yj} will affect mainly the subgradient vector h_i, while ν_j has an impact via the average consensus (5.24).

Updating the equality constraint $h_1^i[k](F_{xi}, F_{yi}, \hat{F}_{xj}, \hat{F}_{yj})$ can be seen as a function of the analytical solution of F_{xi}^*, F_{yi}^*, which depends on the dual multiplier ν_i, and ,$\hat{\nu}_j$. Referring to (5.23), the perturbation parameters $\hat{\nu}_j$ directly affect the update of the local ν_i and have an indirect impact on the primal variables F_{xi}, and F_{yi}. Recall that the subgradient update step followed by an average consensus for the Lagrange variables associated with the equality constraints can be formally given by:

$$\bar{\nu}_i[k+1] = \frac{1}{N_i}\left(\nu_i[k] - \alpha \cdot h_1^i(F_{xi}^*[k], F_{yi}^*[k], \hat{f}_j[k]) + \hat{\nu}_j^i[k]\right),$$

where $\bar{\nu}_i[k+1]$ is the ν_i after consensus. For convenience reasons, we assume that N_i is the number of neighbors of node i in $j \in \mathcal{N}_i$. Also, let $\hat{f}_j$ formally be a placeholder for the received external force variables F_{xj} and F_{yj}, and, similarly, $\hat{\nu}_j^i$ the copy of the global dual variable ν at the jth node received at node i. Given that both variables, F_{xi}^* and F_{yi}^*, depend explicitly on ν_i, as defined by (5.21) and (5.22), we can formally write the above difference equation as:

$$\bar{\nu}_i[k+1] = \chi^i(\nu_i[k], \hat{\nu}_j[k], \hat{f}_j[k]), \tag{5.38}$$

where χ^i is inferred by the latter equation after substituting the symbolic expressions for the optimal solutions F_{xi}^* and F_{yi}^* from (5.22). Now, we are particularly interested in the effect of the received information $\bar{f}_j$ and $\hat{\nu}_j$ in the evolution of ν_i. According to the sensitivity analysis, we can associate to this system a system of the difference equation given by:

$$S_f^i[k+1] = A^i\, S_f^i[k] + B_f^i, \tag{5.39}$$

$$S_\nu^i[k+1] = A^i\, S_\nu^i[k] + B_\nu^i, \tag{5.40}$$

where

$$S_f^i = \left.\frac{\partial \bar{\nu}^i}{\partial \hat{f}_j}\right|_{\hat{f}^j = \hat{f}_j^i} \tag{5.41}$$

$$S_\nu^i = \left.\frac{\partial \bar{\nu}^i}{\partial \hat{\nu}_j}\right|_{\hat{\nu}_j = \hat{\nu}_j^i} \tag{5.42}$$

stands for the sensitivity matrices of ν_i w.r.t. $\hat{f}_j$ and $\hat{\nu}_j$, respectively, and

$$A^i = \frac{\partial \chi^i}{\partial \nu_i}, \quad B_f^i = \frac{\partial \chi^i}{\partial \hat{f}_j}, \quad \text{and} \quad B_\nu^i = \frac{\partial \chi^i}{\partial \hat{\nu}^j}. \tag{5.43}$$

Notice that $\hat{f}_j^i$ is to be understood as the last local copy received at the ith node for the local force vectors $\hat{f}_j$. Similarly, $\hat{\nu}_j^i$ represents the most recent copy received at the ith node for the local copy of the global variable ν at the jth node. Moreover, due to the linearity of function χ in terms of its variables, the matrices A^i and B^i are constant. As already stated above, we consider the received dual variables ν_j at node i in the context of the sensitivity analysis as a perturbing parameter on the computation of the local dual variables. As long as no information is received, the most recent copy $\hat{\nu}_j^i$ is utilized in the update optimization equations. Analogously, we introduce the notation $\hat{F}_{xj}^i$ and $\hat{F}_{yj}^i$.

5.5.3 Effect of the neighbors

The sensitivity functions S_f^i and S_ν^i provide the first-order estimates of the effect of the received variations of f_j and ν_j on the local copy of the dual variable ν_i as expressed by

$$\nu_i[k](f_j, \nu_j) = \nu_i[k](\hat{f}_j, \hat{\nu}_j^i) + S_f^i\,(f_j[k] - \hat{f}_j^i[k]) + S_\nu^i\,(\nu_j[k] - \hat{\nu}_j^i[k]). \tag{5.44}$$

The latter equation quantifies the effect of the reception of ν_j, as well as F_{xj} and F_{yj} into ν_i updates at the ith node. In the next step, we now want to apply these estimations to design an event-triggered communication protocol. To compute the approximated dual variables $\tilde{\nu}_j^i[k+1]$ of the neighbors $j \in \mathcal{N}_i$ at node i, we continuously compute the sensitivity functions S_f^i and S_ν^i w.r.t. the nodes in the neighborhood $\mathcal{N}_i$. However, while its computation can only take place at the ith node, it is important to clarify that the sensitivity functions are actually needed at node j, as they can only be used there to estimate the effect of the transmission of the local variables ν^j, F_{xj}, F_{yj} (cf. (5.44)). Hence, in addition to transmitting ν_j, F_{xj}, F_{yj}, we also need to invoke the exchange of the sensitivity matrix functions S_f and S_ν. Clearly, this is a price that has to be paid by our sensitivity-based event-triggered communication protocol.

5.5.4 Approximation of the neighbors' state

To finalize this section, we emphasize that the estimated impact of the variables F_{xj}, F_{yj} and ν^j at time k on the variable ν^i at node i as computed at node j is done by means of this expression:

$$\begin{aligned}\tilde{\Delta}\nu_i^j[k+1] = \\ &\hat{S}_{i,\nu}^j(\nu_j[k+1] - \hat{\nu}_j^i[k]) \\ &+\hat{S}_{i,F_x}^j(F_{xj}[k+1] - \hat{F}_{xj}^i[k]) \\ &+\hat{S}_{i,F_y}^j(F_{yj}[k+1] - \hat{F}_{yj}^i[k]).\end{aligned} \tag{5.45}$$

For example, here $\hat{S}_{i,F_x}^j$ stands for the most recent copy of the sensitivity matrix $S_{F_x}^i$ available at node j. The other hat-designated sensitivity matrices are to be read in a similar manner. Then the approximated dual variables $\tilde{\nu}_i^j[k+1]$ are computed as follows:

$$\tilde{\nu}_i^j[k+1] = \hat{\nu}_i^j[k] + \tilde{\Delta}\nu_i^j[k+1], \tag{5.46}$$

where $\hat{\nu}_i^j[k]$ is the last transmitted state of node j known by node i. Each node computes the approximated dual variables of its neighbors and executes the event-triggered condition.

5.5.5 Event-triggered condition

The event-triggered condition of node j is based on how the transition of the dual variables ν_j will affect the convergence of the distributed optimization problem at node i, i.e., the evolution of ν_i. In particular, the transmission of the dual variables will affect the neighbors' solution. Therefore, we invoke the concept of an approximated dual variable $\tilde{\nu}_i^j$, which represents an estimation of the variable ν_i as seen from node j. We can now introduce the sensitivity-based event-triggered condition as follows:

$$\left\| \tilde{\Delta}\nu_i^j[k+1] \right\| \geq \beta_0 \left\| h_j \right\| + \beta_1, \tag{5.47}$$

where $0 < \beta_0 \leq 1$ and $0 < \beta_1 < 0.01$ are the triggering parameters that tune the acceptability of the error level. In other words, if this condition is violated, then a transmission request for ν_j, F_{xj}, F_{yj} as well as the corresponding sensitivity matrices $S^j_{\nu^i}$, $S^j_{F_x}$, $S^j_{F_y}$, $\forall i \in \mathcal{N}$ is initiated. Generally, it is important to keep in mind that in the presentation of our sensitivity-based TDMA algorithm, we do consider node j as the sender and node i as the receiver of the information at hand. For the sake of simplicity, we here suggest a broadcast rather than peer-to-peer communication.

5.6 Simulation and discussion

The performance of the proposed algorithms is compared with the result of the centralized optimization problem, where the computed solutions of the adhesion potential η_i and the Lagrange multipliers ν_i are compared with the optimal solution. With the term "centralized optimization" we denote the scenario where a special ECU is dedicated to the accommodation of the evolution of the global ν variables. That is, we dispense with its local copies ν_i. First, we will show the simulation result of the distributed optimization problem, followed by the simulation result of the adaptive TDMA sensitivity-based algorithm. A lane change maneuver under braking conditions is used as a reference trajectory for the vehicle dynamics optimization problem, and the parameters of the maneuver scenario are defined with a maximum longitudinal deceleration of $a_x \approx 3$ m/s^2 and maximum lateral acceleration of $a_y \approx 8$ m/s^2 at a speed of $v = 80$ km/h under dry road conditions with $\mu_{max} = 1$.

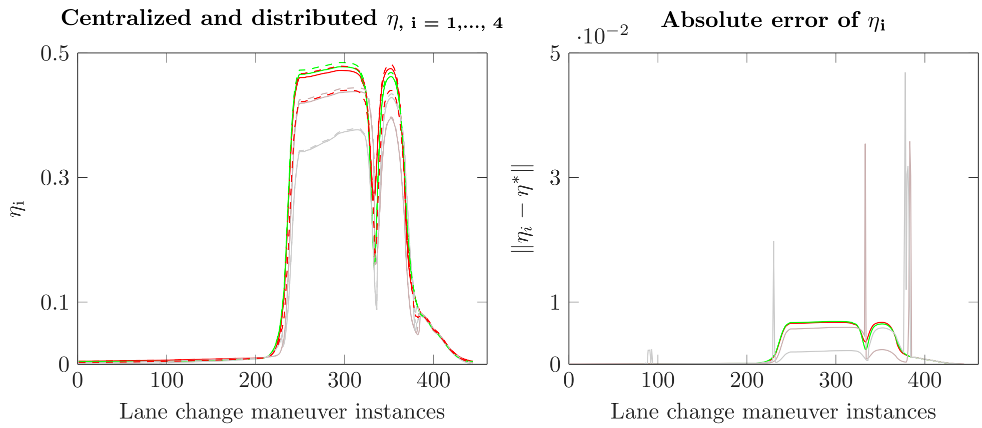

FIGURE 5.6: Performance of the distributed algorithm compared with the centralized one (left), and the absolute error of the distributed algorithm η_i (right).

Fig. 5.6 compares the distributed problem solution with the optimal solution of the centralized problem, and it shows the absolute error of the distributed η_i compared with the centralized solution. The simulation showed good performance of the distributed algorithm even in the critical driving scenario. The absolute error decreased dramatically in the simple maneuver instance and remained acceptable in the difficult one.

Fig. 5.7 presents the convergence of the local dual multipliers ν_i^c with respect to the global dual multiplier ν computed by a centralized solver algorithm. The simulation showed complete convergence of both variables even in the critical maneuver instance.

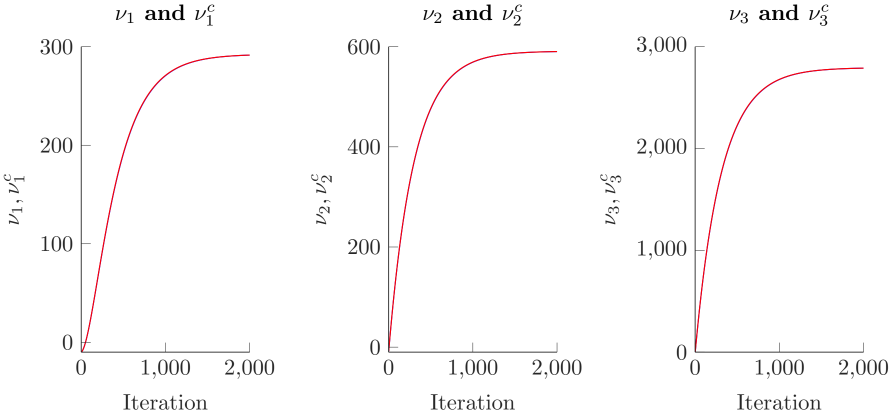

FIGURE 5.7: Behavior of the distributed dual variables ν_i^c (red plots) compared with the global variables ν (blue plots) of the centralized optimization for a critical maneuver instance (250th in Fig. 5.6).

In order to evaluate the proposed sensitivity-based event-triggered adaptive TDMA algorithm, the algorithm was implemented in four nodes communicating over a wireless network based on the TDMA protocol. We point out that the simulation study considered the convergence of the distributed optimization problem with respect to the adaptive TDMA communication protocol. The TDMA-based wireless network was simulated as an additive white Gaussian noise channel, and the SNR was randomly distributed over frequency and time with a maximum value of up to 130 dB. The efficiency of the proposed algorithm was evaluated in terms of the communication reduction in relation to the convergence of the distributed optimization problem and the absolute error rate due to the reduction of the nodes' communication caused by the event-triggered scheme. The application layer consisted of one vehicle cluster consisting of $N = 4$ nodes, which was internally equipped with an event-triggered subgradient solver of the vehicle dynamics optimization problem. The event-triggered scheme parameters were set to $\beta_0 = 0.03$ and $\beta_1 = 0.0001$, which were chosen to guarantee the optimal performance of the distributed algorithm. The dual subgradient method was updated with a fixed step size $\alpha = 0.04$ and was run for 150 subgradient update iteration steps.

The convergence rate of the distributed optimization problem was computed according to the relative error $= \frac{\|\eta_i - \eta^*\|}{\|\eta^*\|}$. The relative error traces the performance of the proposed algorithm by computing the convergence rate of η_i with respect to η_i^* computed by the centralized algorithm. Following the evaluation criterion from our previous work, we started with a full maneuver simulation in order to evaluate the algorithm's performance with different driving scenarios including moderate and extreme maneuvers.

Fig. 5.8 presents the computed adhesion potential η_i for $\{i = 1, ..., 4\}$ nodes, and also shows the absolute error $\|\eta_i - \eta^*\|$ computed with reference to the optimal η^*. We see that the algorithm's performance greatly improved, as it provided a nearly accurate η_i

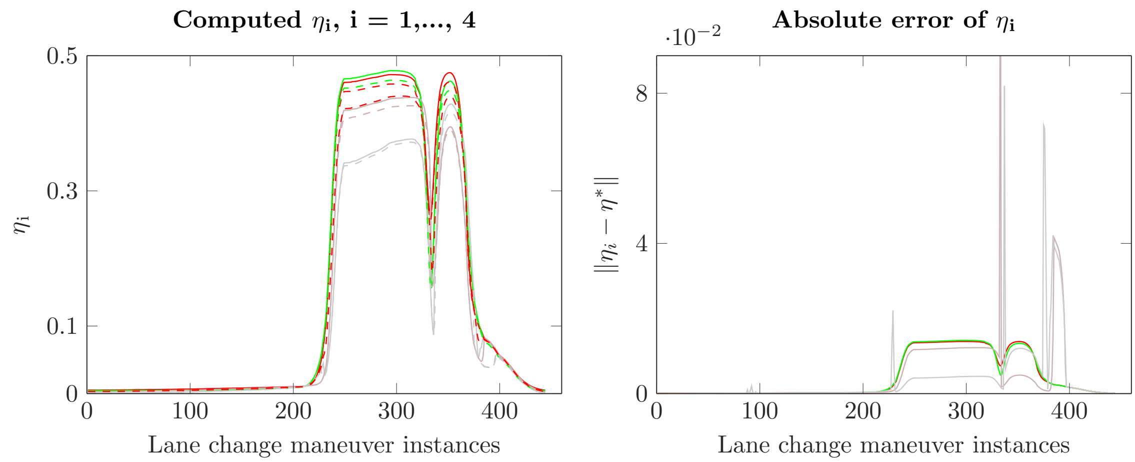

FIGURE 5.8: Algorithm performance over the complete maneuver: computed adhesion potential η_i for all nodes (left) and absolute error (right). The relative error increases in the case of difficult maneuvers.

value with respect to the optimal adhesion potential η^*. Also, a noticeable decrease in the absolute error value can be seen in the case of the simple and moderate maneuver instances. Therefore, the sensitivity-based event-triggered scheme maintains the system's performance with respect to the computation of η_i, also in instances of extreme driving maneuvers. Fig. 5.9 compares the communication activities within the network of the adaptive TDMA and the fixed TDMA protocol of the complete lane change maneuver and shows a noticeable reduction in communication activities up to 70% in the simple and moderate maneuver instances. A noticeable communication reduction can also be seen in the case of the adaptive TDMA protocol. The event-triggered scheme regulates the transmission requests and provides efficient use of communication resources.

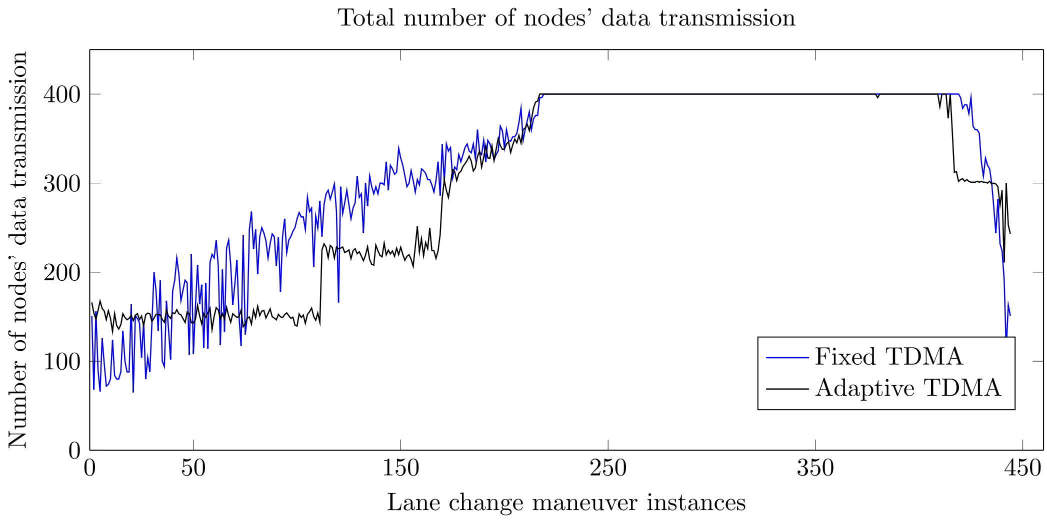

FIGURE 5.9: The net communication activities of all nodes.

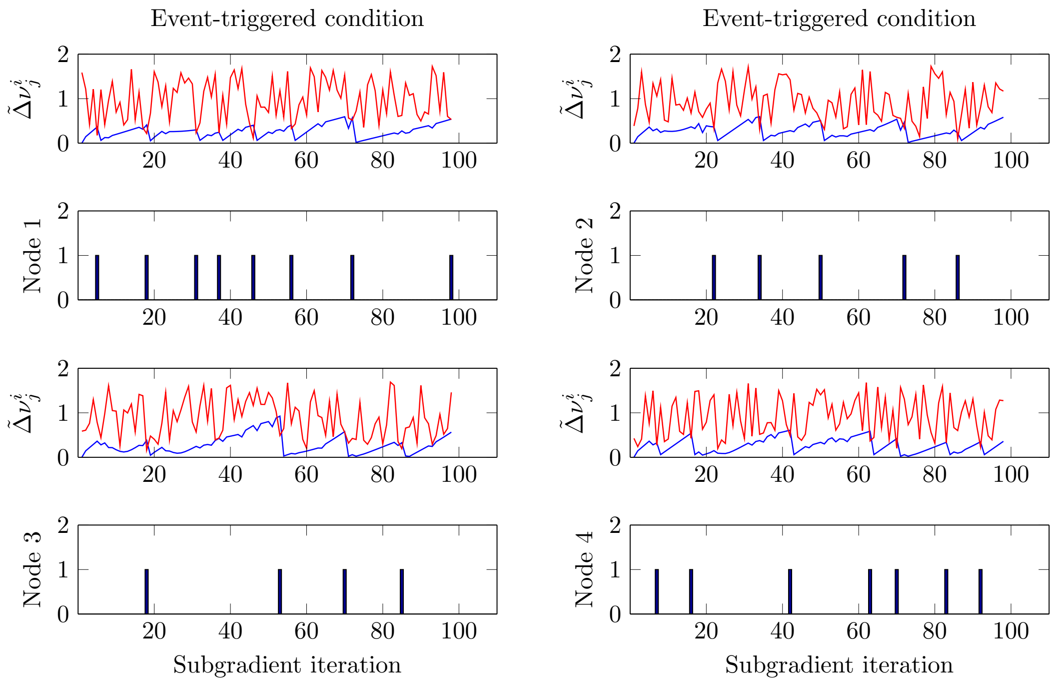

FIGURE 5.10: Performance of the event-triggered condition $\left\|\tilde{\Delta}\nu_i^j[k+1]\right\|$ (blue plots) w.r.t the triggering threshold $\beta_0\left\|h^j\right\|+\beta_1$ (red plots), cf. (5.47), for a maneuver sample with rather low communication load.

Fig. 5.10 presents the performance of the event-triggered condition $\left\|\tilde{\Delta}\nu_i^j[k+1]\right\|$ with respect to the threshold $\beta_0\left\|h^j\right\|+\beta_1$ and shows the communication activities for each node. We see that the node broadcasts its state update if the triggering condition reaches the acceptable error level regulated by the triggering threshold. On the other hand, the event-triggered threshold depends on the equality constraint $h^j=0$, which is sensitive to changes in the other nodes' state variables. It can be seen that the proposed sensitivity event-triggered algorithm leads to a communication reduction between the nodes. Notice that, in the extreme maneuver instance, the nodes require a higher communication rate in order to converge to the optimal value. Therefore, the event-triggered condition will try to reduce communication requests.

Fig. 5.11 shows the communication activities of all nodes in the case of simple, moderate, and difficult maneuver instances. We observe that communication is reduced dramatically in simple and moderate maneuver instances. In the extreme maneuver instance, communication is not reduced because the solution algorithm requires more data exchange in order to converge to the optimal value.

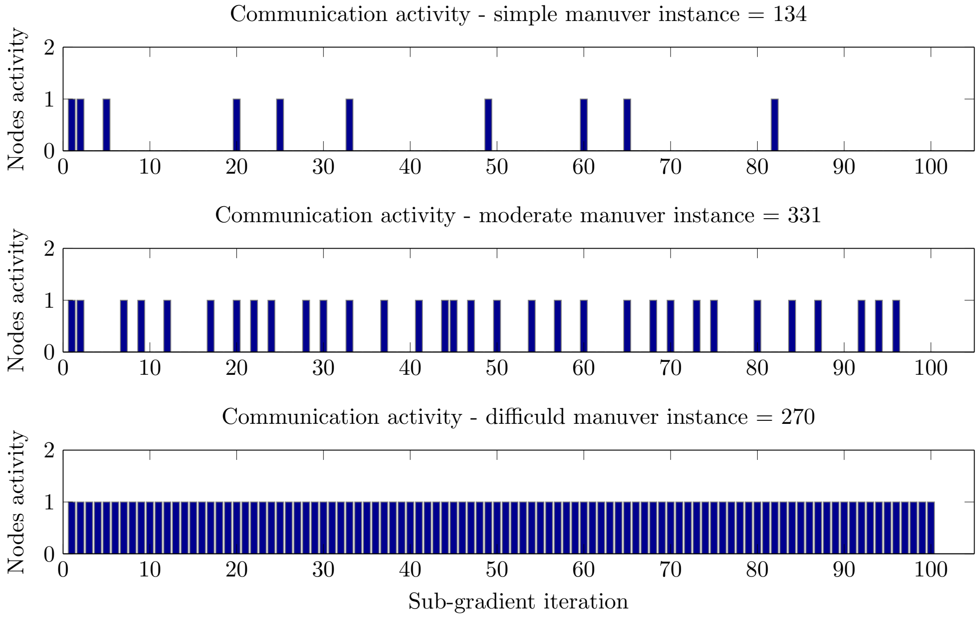

FIGURE 5.11: Communication activities: Node communication is reduced in the simple maneuver instance, somewhat reduced in the moderate maneuver instance, and not reduced in the difficult maneuver case.

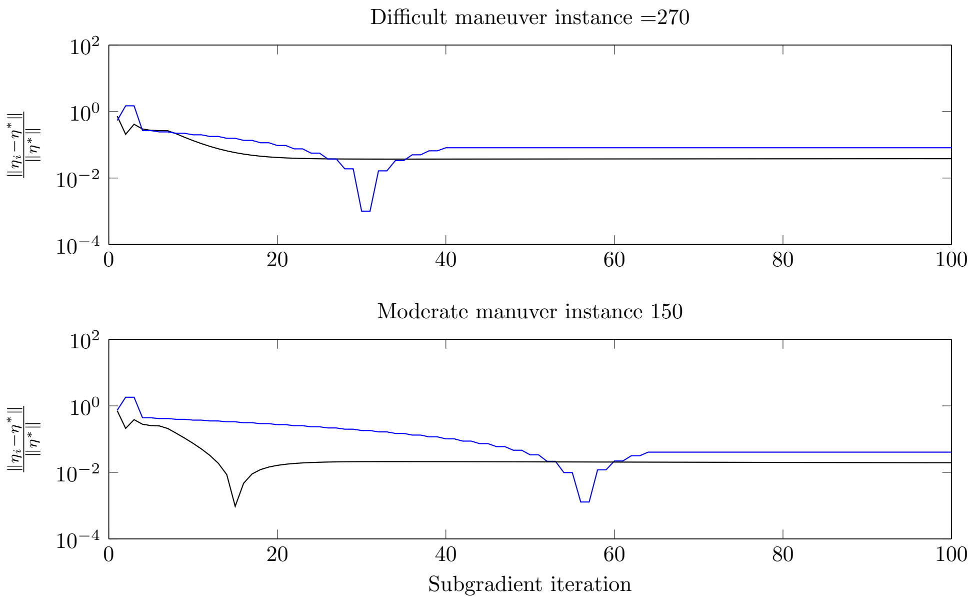

FIGURE 5.12: Convergence rate in the extreme and moderate maneuver instances for event-triggered algorithms based on local increments (blue plots) and on estimated increments about the nodes in the environment (black plots).

Finally, Fig. 5.12 presents the convergence rate of the extreme and moderate maneuver instances. We see that the distributed problem converges slower in the case of the extreme maneuver instance and faster in the moderate case, with reduced error in the computed η_i.

5.7 Chapter conclusion

In this chapter, we presented a sensitivity-based event-triggered adaptive TDMA protocol with application to the optimal control problem of vehicle dynamics. In particular, the sensitivity of the distributed optimization problem was used to formulate an event-triggering condition with respect to the convergence of the overall optimization problem. The tracked sensitivity functions were used to approximate the effect of the transmission of the state update of a node on its neighbors' states, which represents the basis of our event-triggered condition for a transmission request. The simulation results demonstrate that the proposed protocol achieved acceptable communication reduction and also maintained the system's performance even during critical driving maneuvers, which are typically associated with high communication demands. Additionally, we considered event-triggered algorithms based on local state increments. The analysis was completed informally through extensive simulation in various scenarios. Finally, we extended the algorithm for optimal tire force allocation in a setting reflecting the minimization of tire adhesion, and used the dual subgradient algorithm as a solution algorithm.

To conclude, we introduced a sensitivity-based event-triggered TDMA protocol in connection with the guidance and stabilization problem of vehicle dynamics. We noticed that the TDMA protocol limits the system's capacity in terms of the number of nodes that can be served. By increasing the number of nodes in the network, the inherited time delay is increased dramatically by the factor of the number of connected nodes, which implies a reduction of the network capacity in terms of preserving system performance. Moreover, we also conclude that the usage of the event-triggered scheme is promising even though the number of nodes is strictly limited by the TDMA protocol in case of time-sensitive applications.

Algorithm 3: Event-based communication for dual algorithm

Exchange: $F_{xi}, F_{yi}, \nu_i, \forall i \in \mathcal{N}$;
while *Subgradient loop* **do**
  Update: $\eta_i, F_{xi}, F_{yi}, \lambda_i, \sigma_i, \nu_i$;
    If $\big(\|\nu_i[k+1] - \hat{\nu}_i[k]\| \geq \beta_0 \left\|h^i[k]\right\| + \beta_1 \big)$
      RTS$_i \leftarrow$ Active
    end
    If Active(RTS$_i$)
      Transmit: ν_i, F_{xi}, F_{yi} ;
    end
  Node *i* receives $\hat{\nu}_j, \hat{F}_{jx}, \hat{F}_{jy}, \forall j \in \mathcal{N}_i$;
  Consensus: $\nu_i^{(c+1)}[k] = \nu_i^{(c)}[k] + \frac{1}{N} \sum_{j \in \mathcal{N}_i} w_{ij} \left(\hat{\nu}_j[k] - \nu_i^{(c)}[k] \right)$;
end

Algorithm 4: Centralized dual subgradient algorithm.

while *Subgradient loop* **do**
  Update κ_{xi},κ_{yi};
    $\kappa_{xi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}$;
    $\kappa_{yi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}$;
  Update primal variables;
    $\eta_i \leftarrow \frac{1}{2} (\lambda N_f - \sigma)$;
    $F_{xi} \leftarrow \frac{\kappa_{xi}}{2\epsilon^2} \left(\frac{\lambda}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1 \right)$;
    $F_{yi} \leftarrow \frac{\kappa_{yi}}{2\epsilon^2} \left(\frac{\lambda}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1 \right)$;
  Update dual variables;
    $f1[k] \leftarrow \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f$;
    $f2[k] \leftarrow \eta_i - \mu_{\max}$;
    $h1[k] \leftarrow A_x \hat{F}_{xj} + A_y \hat{F}_{yj} - Y_d$;
    $\lambda[k+1] \leftarrow (\lambda[k] - \alpha f_1[k])_+$
    $\sigma[k+1] \leftarrow (\sigma[k] - \alpha f_2[k])_+$
    $\nu^T[k+1] \leftarrow \nu^T[k] - \alpha h_1[k]$
  Projection λ,σ;
  $\lambda[k+1] \leftarrow max(\lambda[k+1], 0)$;
  $\sigma[k+1] \leftarrow max(\sigma[k+1], 0)$;
  increment k;
end

Algorithm 5: Distributed dual subgradient algorithm.

Exchange $\nu_i, F_{xi}, F_{yi}, \forall i \in \mathcal{N}$;
while *Subgradient loop* **do**
  Update κ_{xi},κ_{yi};
    $\kappa_{xi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}$;
    $\kappa_{yi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}$;
  Update primal variables;
    $\eta_i \leftarrow \frac{1}{2}\left(\lambda N_f - \sigma_i\right)$;
    $F_{xi} \leftarrow \frac{\kappa_{xi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2+\kappa_{yi}^2}} - 1\right)$;
    $F_{yi} \leftarrow \frac{\kappa_{yi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2+\kappa_{yi}^2}} - 1\right)$;
  Exchange ν_i;
  Consensus algorithm;
    $\nu_i^{(c+1)} \leftarrow \nu_i^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{\nu}_j[k] - \nu_i^{(c)}[k]\right)$;
  Update dual variables;
    $f_1^i[k] \leftarrow \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i$;
    $f_2^i[k] \leftarrow \eta_i - \mu_{\max}$;
    $h_1^i[k] \leftarrow A_x \hat{F}_{xj} + A_y \hat{F}_{yj} - Y_d$;
    $\lambda_i[k+1] \leftarrow (\lambda_i[k] - \alpha f_1^i[k])_+$
    $\sigma_i[k+1] \leftarrow (\sigma_i[k] - \alpha f_2^i[k])_+$
    $\nu_i^T[k+1] \leftarrow \nu_i^T[k] - \alpha h_1^i[k]$
  Projection λ_i,σ_i;
  $\lambda_i[k+1] \leftarrow max(\lambda_i[k+1], 0)$;
  $\sigma_i[k+1] \leftarrow max(\sigma_i[k+1], 0)$;
  increment k;
end

Algorithm 6: Sensitivity-based adaptive TDMA

Exchange: $\nu^i, F_{xi}, F_{yi}, S^i_{\nu}, S^i_{F_x}, S^i_{F_y}, \forall i \in \mathcal{N}$;
while *Subgradient loop* **do**
  Update κ_{xi},κ_{yi};
    $\kappa_{xi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}$;
    $\kappa_{yi} \leftarrow \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}$;
  Update primal variables;
    $\eta_i \leftarrow \frac{1}{2}(\lambda N_f - \sigma_i)$;
    $F_{xi} \leftarrow \frac{\kappa_{xi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2+\kappa_{yi}^2}} - 1\right)$;
    $F_{yi} \leftarrow \frac{\kappa_{yi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2+\kappa_{yi}^2}} - 1\right)$;
  Compute sensitivity matrices:
    $S^i_{F_x}[k+1] \leftarrow A^i\, S^i_{F_x}[k] + B^i_{F_x}$;
    $S^i_{F_y}[k+1] \leftarrow A^i\, S^i_{F_y}[k] + B^i_{F_y}$;
    $S^i_{\nu}[k+1] \leftarrow A^i\, S^i_{\nu}[k] + B^i_{\nu}$;
  Compute approximated change:
    $\tilde{\Delta}\nu_i^j[k+1]$ at a $j \in \mathcal{N}_i$;
    If $\left(\left\|\tilde{\Delta}\nu_i^j[k+1]\right\| \geq \beta_0 \left\|h^j\right\| + \beta_1\right)$
      $RTS_j \leftarrow$ Active;
    end
    Adaptive TDMA protocol
    If (Active(RTS_j))
      Rebuild the TDMA frame and assign Node$_j \leftarrow Ts_j$
      Transmit ν^j, F_{xj}, F_{yj}, S^j_{ν}, $S^j_{F_x}$, $S^j_{F_y}$;
    end
  Node i receives: $\nu^j, F_{xj}, F_{yj}, S^j_{\nu}, S^j_{F_x}, S^j_{F_y}, \forall j \in \mathcal{N}_i$;
  Consensus: $\nu_i^{(c+1)} \leftarrow \nu_i^{(c)}[k] + \frac{1}{N}\sum_{j\in\mathcal{N}_i} w_{ij}\left(\hat{\nu}_j[k] - \nu_i^{(c)}[k]\right)$;
  Update dual variables;
    $f_1^i[k] \leftarrow \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i$;
    $f_2^i[k] \leftarrow \eta_i - \mu_{\max}$;
    $h_1^i[k] \leftarrow A_x \hat{F}_{xj} + A_y \hat{F}_{yj} - Y_d$;
    $\lambda_i[k+1] \leftarrow (\lambda_i[k] - \alpha f_1^i[k])_+$
    $\sigma_i[k+1] \leftarrow (\sigma_i[k] - \alpha f_2^i[k])_+$
    $\nu_i^T[k+1] \leftarrow \nu_i^T[k] - \alpha h_1^i[k]$
  Projection λ_i,σ_i;
  $\lambda_i[k+1] \leftarrow max(\lambda_i[k+1], 0)$;
  $\sigma_i[k+1] \leftarrow max(\sigma_i[k+1], 0)$;
  increment k;
end

Chapter 6

Distributed event-triggered adaptive OFDMA protocol

In a distributed wireless network control system, the exchange of states over a wireless channel requires extensive consumption of communication resources [98]. At the same time, the wireless channel suffers from limited resources in terms of bandwidth, data rate, and channel capacity. Therefore, with an increasing number of wireless nodes requiring service from the channel, resource allocation techniques are needed to implement proper distribution of the channel resources among the connected nodes. This channel resource allocation has to guarantee that all connected nodes receive at least a minimum of the required service, and must ensure that their requests are handled in accordance with the quality-of-service (QoS) requirements [25].

In general, event-based communication mechanisms in network control applications introduce a purposeful reduction of the communication in accordance with the dynamic evolution of the local subsystem states. We refer to the work [99], which considers sensitivity-based event-triggered communication policies in the context of model-predictive-control (MPC) and used sensitivity analysis of the optimization problem of the local MPC controller. We adapted this idea by designing an event-triggered policy for the resource allocation optimal control problem in the Orthogonal Frequency-Division Multiple Access (OFDMA) protocol and linking it with the local optimal control problem of our application benchmark. Here, we extend these ideas in two directions. The first one refers to the adaptive OFDMA resource allocation protocol and is based on linking event-triggered conditions to the sensitivity analysis of the underlying optimization problem. The second extension is to apply this idea to an individual vehicle's control dynamics within a multi-vehicle cluster.

This chapter is organized as follows. Section 6.1 introduces the general concept of the OFDMA-based wireless network. Section 6.2 presents the event-triggered sensitivity-based OFDMA protocol. Section 6.3 presents the system structure of a multi-vehicle cluster communicating over an OFDMA-based wireless network. Section 6.4 presents the simulation results and the discussion, and the chapter ends with the conclusion in section 6.5.

6.1 Orthogonal Frequency Division Multiple Access (OFDMA) protocol

6.1.1 Technical issues and operation of the OFDMA protocol

The OFDMA protocol was considered an essential technique in the past decade and is the main type of medium access control used in the fourth generation of wireless communication (4G) [100], as well as for 5G implemented in Long-Term Evolution (LTE) technology. The basic principle of OFDMA is to divide the available bandwidth into a number of low data rate sub-frequencies (called subcarriers) in an orthogonal structure. In fact, the orthogonality technique yields overlapping spectra of the individual subcarriers and can be achieved through proper selection of the carrier spacing between the sub-frequency bands [2], as shown here:

$$X_z(t) = \begin{cases} \sum_{i=1}^{k_b} X(l)\sqrt{\frac{E_b}{T_b}}e^{[j2\pi f_d(2l-k_b-1)t]}, & if \quad 0 \leq t \leq T_p \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

where $X(l)$ is the modulation sample, $2f_d$ is the separation frequency between two subcarriers needed to reduce frequency interference, and $W_b = \frac{K_b}{T_b}$ is the bit rate per second [40]. Fig. 6.1 shows the orthogonality principle of the frequency division multiplexing (OFDM), where the frequency band is divided into a large number of closely spaced sub-frequencies orthogonal to each other. Accordingly, each set of subcarriers is used to transmit in parallel over one time slot. For example, in the LTE standard, a set of 180

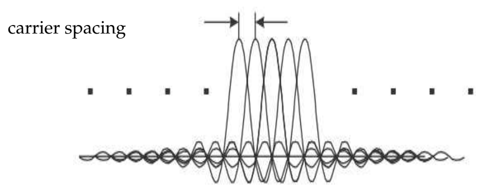

FIGURE 6.1: OFDM figure.

KHz frequency bands presents a collection of 12 subcarriers grouped into one resource block (RB) [101], where the RB contains the sub-frequency band over a specific time slot.

Moreover, this provides efficient use of the available spectrum, as the bandwidth is divided into a number of closely spaced narrow sub-frequencies. It operates on the basis of the Frequency Division Duplex (FDD) multiplexing technique, which uses two separate sub-frequency bands, one for transmitting and the other one for receiving. The two sub-frequency bands are separated by a guard band to reduce frequency overlapping and interference. In general, FDD provides high efficient use of the available spectrum, and robustness against multi-path fading channels, resistance to multi-user interference, and simplified equalization.

6.1.2 OFDMA network infrastructure

The main infrastructure of an OFDMA-based wireless network consists of Base Station (BS), also called Access Point (AP), which exercises the communication control between the interconnected wireless nodes. Below, we list the main components and functionalities of an OFDMA-based network.

1. **Access Point (AP)**

   The primary tasks of the AP are resource scheduling, subcarrier assignment, and communication handling. Technically, communication and data exchange between the interconnected nodes and the AP is performed in two phases, the uplink phase and the downlink phase. In the uplink phase, the AP uploads the data from the nodes that are assigned subcarriers, and in the downlink phase, it delivers the transmitted data to the destination nodes and broadcasts the resource allocation table for the next communication frame. The AP is equipped with an OFDM-based transceiver, which handles the uplink and downlink processes, and with a scheduler, which performs the resource allocation and subcarrier assignment operations. The AP assigns the subcarriers to the nodes in the form of (RB) and selects the appropriate modulation method to modulate the transmitted data over the subcarrier. For instance, the modulation method is selected based on the node's distance to the AP and with respect to the measured Signal to Noise Ratio (SNR) of the node over all subcarriers. Accordingly, for nodes that are closer to the access point with better SNR, the AP selects the 64-Quadrature Amplitude Modulation (64QAM) with 6 bit/Hz for data transmission, whereas for more distant nodes with less (SNR), 16-Quadrature Amplitude Modulation (16QAM) with 4 bit/Hz or Quadrature Phase Shift Keying (QPSK) modulation with a 2 bit/Hz data rate is used, respectively.

2. **OFDMA transceiver**

   The AP is equipped with a transceiver, which is depicted in Fig .6.2 and consists of an OFDM-based transmitter and a receiver that handle the data exchange between nodes.

   The transmitter modulates the transmitted data separately over each sub-frequency band and selects the proper modulation method based on the node's SNR measurement, where the data stream of node i is divided into blocks of length N

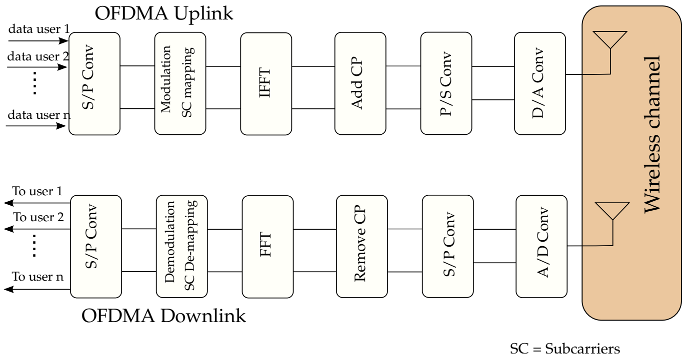

FIGURE 6.2: OFDMA transceiver - uplink transmitter and downlink receiver.

data symbols. Then the N data symbols are modulated into their assigned subcarriers in the modulation and subcarrier mapping block, whose output is $s(n) = [s_1(n), ..., s_i(n), ..., s_N(n)]$ modulated symbols. Following that, the modulated symbols enter the Inverse Fast Fourier Transform (IFFT) block, where the (IFFT) block transforms the N data stream from the frequency domain into the time domain: At the end, the set of $s(n)$ symbols is manipulated by the IFFT block and produces the time domain block vector $\{x(n) = x_1(n), ..., x_i(n), ..., x_n(N)\}$ by applying the IFFT transformation as follows: [1, 101]:

$$IFFT\{X[n]\} = x[n] = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} X[i] e^{-j\frac{2\pi n_i}{N}}, \tag{6.2}$$

where the resultant $x(n)$ presents the orthogonality of the subcarriers produced by the IFFT block, which goes through the Cyclic Prefix (CP) block to insert a frequency guard, which equals 5.2 μs in the time domain and 15 KHz in the frequency domain, between two successive subcarriers in order to eliminate the Inter Symbol Interference (ISI) problem [101]. In contrast, the OFDMA receiver works in the reverse order than the transmitter: It starts by converting the received serial data stream into a N symbols parallel data stream; then the produced data stream goes into the CP remover block to remove the cyclic prefix guard interval. In order to recover the data sequence $X[i]$, the data stream enters the Fast Fourier Transform (FFT) block to recapitulate the frequency domain data out of the discrete time sequence $x[n]$ by applying the Fast Fourier Transform (FFT) [1] as follows:

$$FFT\{x[n]\} = X[i] = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} x[N] \exp^{-j\frac{2\pi n_i}{N}}, \tag{6.3}$$

where the output frequency domain data $X[i]$ goes into the demodulation block, where the signal streams are demodulated, and the data is extracted from the attached frequency; see Fig. 6.2.

3. **OFDMA scheduler and subcarrier assignment**

The main functionality of the OFDMA scheduler is to define which subcarrier is to be assigned to each user, and which power level needs to be allocated to the node over that subcarrier to transmit its data. In order to guarantee maximum use of the available spectrum, the scheduler is equipped with a resource allocation algorithm that solves the optimization problem for the optimal allocation of the available communication resources with respect to a predefined criterion, such as minimizing the power level, maximizing the data rate of the channel, or maximizing the data rate for each subcarrier with different sets of constraints. Generally, the OFDMA scheduler deals with a communication request from the connected nodes and manages the assignment of the available subcarriers to the nodes in specific time slots by solving a resource allocation optimization problem.

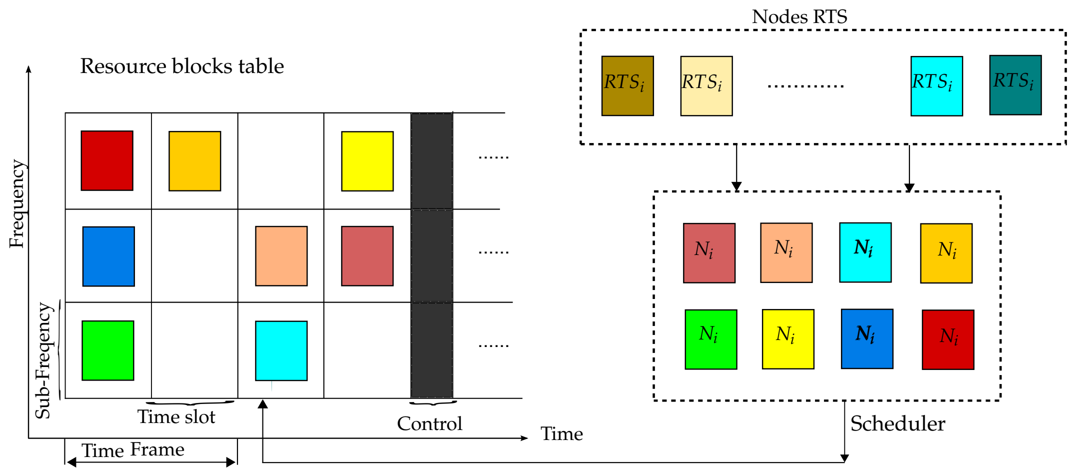

FIGURE 6.3: AP scheduler and resource block representation.

The resource allocation optimization problem is generally resolved with respect to the nodes' SNR over the available subcarriers reported to the AP. As a result, the scheduler assigns the resource block (see Fig.6.3) to the selected node by filling the resource allocation table with the subcarriers and the time slot assigned to that node.

6.1.3 Resource allocation

A set of general resource allocation algorithms used in the OFDMA scheduler is listed in [100, 102]. These include: maximum sum rate algorithm, which maximizes the sum rate of all users; maximum fairness algorithm, which maximizes the minimum user rate; proportional rate constraint algorithm, which maximizes the sum throughput of all users; and proportional fairness scheduling, which is related to the latency tolerance to reach maximum throughput over the entire channel time. In general, the main concern of resource allocation algorithms is to maximize the data rate and minimize power consumption. The work [103] proposed an enhanced rate adaptive resource allocation scheme for OFDMA, where an optimization problem was solved to maximize the sum of the adaptive rates with constraints on the user's transmit power. A priority-based criterion and proportional fairness cost function was then applied to this solution. The same approach is discussed in [104], where an algorithm for proportional rate maximization is introduced that is formulated as a nonlinear mixed integer programming problem and constraints are defined on the power consumption of each user. A stochastic approximation of the dual problem was used here to perform an adaptive approximation in the case of availability and unavailability of channel distribution information. This was followed by the work [105], which proposes an optimal subcarrier and power allocation strategy for the downlink communication of the OFDMA protocol. Its cost function minimizes the power consumption of the user with constraints on the data rate and allocates an optimal power for each user to transmit a fixed amount of data within a particular time slot. On the other side, the algorithm proposed in [106] aims to maximize the overall rate while achieving proportional fairness among users under total power constraints.

6.2 Event-triggered sensitivity-based OFDMA protocol

In this section, our aim is to establish a connection between the OFDMA scheduler and the application layer, where we map the resource allocation algorithm to the performance of the distributed optimal control problem and improve the performance of the event-based communication scheme by considering the effect of the neighbors' dynamics on the local event-triggered condition. Previously, we used the local state difference as an event-triggered condition, and the scheduler blindly assigned the available communication resources to the nodes, without any direct connection between the computation of the optimal solution and the resource allocation mechanism. Therefore, we will now use a sensitivity analysis with respect to the effect of the state exchange on its neighbors' state to determine how the transmission of the node's state will improve the computation of the overall optimal solution by the neighbors.

6.2.1 Sensitivity analysis

In the context of event-based communication, the effect of the data exchange between distributed sub-problems will be determined by means of sensitivity analysis and will be used to compute the weight parameters w_i of the node within the resource allocation optimization problem. In general, this approach is motivated by the idea of introducing the sensitivity of the optimal solution of the neighbors' sub-problems with respect to the nodes' state transmission. The approach utilizes the optimality conditions of the distributed optimal control problem. For generality reasons, we consider a networked system presented as a distributed convex optimization problem, where the cost function f_i^0 and the inequality constraints $f_p^i \leq 0$ are convex and the equality constraints $h_q^i = 0$ are affine:

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i^0(x_i, \hat{x}_j^i) \\ \text{subject to} \quad & f_p^i(x_i, \hat{x}_j^i) \leq 0, \; p = 1, \ldots, P_i, j \in \mathcal{N}_i \\ & h_q^i(x_i, \hat{x}_j^i) = 0, \; q = 1, \ldots, Q_i, j \in \mathcal{N}_i, \end{aligned} \tag{6.4}$$

where x_i is the local state of sub-problem f_i^0. The local cost function f_i^0 is updated based on its own state x_i and on the received state $\hat{x}_j^i$ (corresponding to x_j) from its neighbors $\mathcal{N}_i$, which includes the set of sub-problems that exchange their state with the ith optimizing sub-process. The Lagrangian $L_i(x_i, \hat{x}_j^i, \lambda_p^i, \nu_q^i)$ corresponding to the i^{th} sub-problem (6.4) is then given by:

$$L_i(x_i, \hat{x}_j^i, \lambda_p^i, \nu_q^i) = f_i^0 + \sum_{p=1}^{P_i} \lambda_p^i f_p^i + \sum_{q=1}^{Q_i} \nu_q^i h_q^i \tag{6.5}$$

where $\lambda_p^i \geq 0$ and ν_q^i are the dual multipliers. Referring to the first-order sensitivity theorem, let $y_i^* = [x_i^*, \lambda_p^{i*}, \nu_q^{i*}]^T$ represent the minimizer to the above Lagrangian L_i and consider $\varepsilon_j := x_j - \hat{x}_j^i$ as a perturbation of the received state of sub-problem j at sub-problem i. Here x_j is to stand for an update of $\hat{x}_j^i$ in (6.5).

More specifically, we consider the basic sensitivity theorem [107], which is based on first-order Karush–Kuhn–Tucker (KKT) conditions [47, 50, 52]. It is defined for the convex optimization problem (6.4) and its Lagrangian (6.5) with the assumption that the cost function f_i^0 is differentiable. The Karush–Kuhn–Tucker (KKT) optimality conditions state that, if the primal problem is convex for the points x^*, λ^*, and ν^*, it is sufficient for the point x^* to be primal optimal, and for the points λ^*, and ν^* to be dual optimal if they satisfy the following KKT conditions:

$$\begin{aligned} & \nabla L_i(x_i, \hat{x}_j^i, \lambda_p^i, \nu_q^i) = 0, \\ & \lambda_p^i f_i(x^*, \hat{x}_j^i) \leq 0, \qquad p = 0, \ldots, P_i, \\ & h_i(x^*, \hat{x}_j^i) = 0, \qquad q = 0, \ldots, Q_i, \end{aligned} \tag{6.6}$$

Herein, the KKT second-order optimality conditions are defined as follows:

$$\begin{aligned} & f_p^i(x_i^*, \hat{x}_j^i) \leq 0, \qquad p = 0, \ldots, P_i, \\ & h_q^i(x_i^*, \hat{x}_j^i) = 0, \qquad q = 0, \ldots, Q_i, \\ & \lambda_p^* > 0, \qquad p = 0, \ldots, P_i, \\ & \lambda_p^i f_i^0(x_i^*, \hat{x}_j^i) = 0, \qquad i = 0, \ldots, \mathcal{N}, j \in \mathcal{N}_i \\ & \nabla f_i^0(x_i^*, \hat{x}_j^i) + \sum_{p=1}^{P_i} \lambda_p^* \nabla f_p^i(x^*, \hat{x}_j^i) + \sum_{q=1}^{Q_i} \nu_q^* \nabla h_q^i(x^*, \hat{x}_j^i) = 0, \end{aligned} \tag{6.7}$$

The points x_i^*, λ^* and ν^* are the primal and the dual optimal parameters with zero duality gap, under the assumption that the system (6.6) is once continuously differentiable in all arguments in order to derive the Jacobian matrix with respect to the parameters $x_i, \hat{x}_j^i, \lambda_p^i$, and ν_q^i. Basically, the KKT conditions are used to formulate the basic sensitivity analysis theorem [107], which we will use to compute the sensitivity of the data exchange between the sub-problems and then to define the event-triggered condition and the node weight parameter for the resource allocation problem.

6.2.2 Application of the basic sensitivity analysis theorem

The basic sensitivity analysis theorem [107] states that if the cost function f_0^i is twice continuously differentiable, f_p^i and h_q^i are continuously differentiable, then the gradients ∇f_p^i and ∇h_q^i are linearly independent, and the second-order sufficient optimality condition holds at $y_i^* = [x_i^*, \lambda_p^{i*}, \nu_q^{i*}]^T$, yielding a once continuously differentiable vector function $y_i(\varepsilon_j) := [x_i(\varepsilon_j), \lambda_p^i(\varepsilon_j), \nu_q^i(\varepsilon_j)]^T$ for small ε_j in the neighborhood of the received variables $\hat{x}_j^i$. Technically, we approximate the change of the neighbors' $j \in \mathcal{N}_i$ state based on the change of the state of node i by computing the sensitivity matrix S_i^j and solving the following sensitivity equation [107]:

$$S_i^j = M_i^{-1} N_i^j, \tag{6.8}$$

where M_i is the Jacobian matrix of (6.5) with respect to x_i, λ_p^i, and ν_q^i of node i, and N_i^j is the negative Jacobian matrix with respect to x_j^i received at node i. The sensitivity matrices M_i and N_i^j are computed as follows:

$$M_i = \begin{bmatrix} \nabla_{x_i}^2 L_i & -\nabla_{x_i}^T f_1^i & \dots & -\nabla_{x_i}^T f_{P_i}^i & \nabla_{x_i}^T h_1^i & \dots & \nabla_{x_i}^T h_{Q_i}^i \\ \lambda_1^i \nabla_{x_i} f_1^i & f_1^i & 0 & 0 & 0 & 0 & 0 \\ \vdots & 0 & \ddots & 0 & 0 & 0 & 0 \\ \lambda_{P_i}^i \nabla_{x_i} f_{P_i}^i & 0 & 0 & f_{P_i}^i & 0 & 0 & 0 \\ \nabla_{x_i} h_1^i & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \nabla_{x_i} h_{Q_i}^i & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$N_i^j = \left[-\nabla_{\hat{x}_j^i x_i}^{2,T} L_i, -\lambda_1^i \nabla_{\hat{x}_j^i}^T f_1^i, \dots, -\lambda_{P_i}^i \nabla_{\hat{x}_j^i}^T f_{P_i}^i, -\nabla_{\hat{x}_j^i}^T h_1^i, \dots, \nabla_{\hat{x}_j^i}^T h_{Q_i}^i \right]^T. \tag{6.9}$$

Herein, each node computes the sensitivity matrix S_i^j with respect to the state received from its neighbors and transmits it to each neighbor. The sensitivity information S_i^j provides the effect of the perturbation variables $\hat{x}_j^i$ received for the neighbors $j \in \mathcal{N}_i$

on the local state variables x_i, which will be used to approximate the changes of the neighbors' states with respect to the local updated state.

6.2.3 State approximation $\tilde{x}_j^i$

Mainly, we approximate the neighbors' $j \in \mathcal{N}_i$ state by utilizing the sensitivity matrix computed at node i, and use it to evaluate how big its effect is in order to assign communication resources to node i according to that. In general, the approximated state $\tilde{x}_j^i$ of the neighbor j is computed at node i using the last received state $\hat{x}_j^i$ and the received sensitivity $\hat{S}_j^i$ (note that the roles of i and j are switched now, cf. Eq. (6.8)). The approximation of the state $\tilde{x}_j^i$ at node i [99] is computed as follows:

$$\tilde{x}_j^i[k+1] = \hat{x}_j^i[k] + \hat{S}_j^i(x_i[k+1] - \hat{x}_i^j[k]), \tag{6.10}$$

where $\hat{x}_j^i[k]$ is the last transmitted state of node j received at node i, whereas $\hat{x}_i^j[k]$ is the last transmitted state of node i received by node j. Also note that $\hat{S}_j^i$ is the most recent copy of the sensitivity matrix S_j^i received at node i from node j. Again, S_j^i reflects the effects of the "perturbing" state x_i on the state x_j. The matrix S_j^i needs to be computed at node j and has to be shared with node i.

6.2.4 Event-triggering condition

The formulation of the event-triggering condition of node i is based on how the transmission of the state x_i will affect the optimization problem convergence. Therefore, the computed approximated state $\tilde{x}_j^i$ approximates the solution of node j, which is used to calculate the effect of the updated state of node i on the convergence of the overall optimization problem. We now introduce the approximated event-triggered condition of the changes of the overall cost function with respect to the approximated value of the state $\tilde{x}_j^i$ of all neighbors. To this end, we compute the cost function increment $\Delta\tilde{J}_0^i$ of (6.4) at node i as follows:

$$\Delta\tilde{J}_0^i \cong \nabla_{x_i} f_i^0(x_i, \hat{x}_j^i)\Delta x_i + \sum_{j \in \mathcal{N}_i} \nabla_{x_j} f_j^0(\tilde{x}_j^i, \hat{x}_i^j)\Delta\tilde{x}_j^i, \tag{6.11}$$

where $\nabla_{x_i} f_i^0$ and $\nabla_{x_j} f_j^0$ are the gradients of the cost function assigned to nodes i and node j, respectively, while $\Delta x_i = x_i[k+1] - x^i[k]$ and the estimated state increment $\Delta\tilde{x}_j^i$ is computed as the difference of the approximated value of the state $\tilde{x}_j^i$ from equation (6.11) at time $k+1$ and the last transmitted state $\hat{x}_j^i$ known by node i:

$$\Delta\tilde{x}_j^i = \tilde{x}_j^i[k+1] - \hat{x}_j^i[k]. \tag{6.12}$$

The estimated increment $\Delta\tilde{J}_0^i$ of the cost function J_0^i is used to define the triggering criterion of node i, as well as for the computation of the node weight w_i (see below),

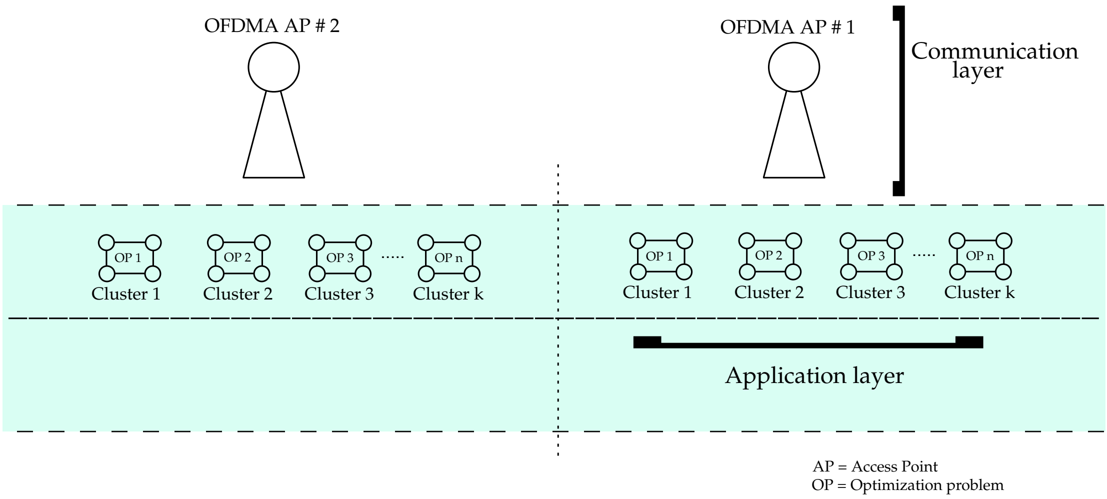

FIGURE 6.4: System structure consisting of two layers: communication layer and application layer. Note that here, a cluster refers to the nodes hosted within a single vehicle.

which is utilized for resource allocation. More specifically, node i activates the Request To Send RTS_i signal, thereby acquiring subcarriers from the AP to transmit its state x_i to the neighbors j if $\Delta\tilde{J}_0^i$ fulfills the following condition:

$$\left\|\Delta\tilde{J}_0^i\right\| \geq \beta_0 \left\|h_q^i(x_i, \tilde{x}_j^i)\right\| + \beta_1, \tag{6.13}$$

where $0 < \beta_0 \leq 1$ and $0 < \beta_1 < 0.01$ are the triggering parameters that tune the acceptability of the state error level.

6.3 OFDMA-based wireless networked vehicle clusters

We consider the distributed wireless system depicted in Fig. 6.4, which consists of the AP and a set of V multi-vehicles. Each vehicle is presented as a cluster with $n = 4$ wireless nodes connected to the AP. The system structure consists of two layers, the communication layer at the AP level and the application layer at the wireless nodes level. The first layer consists of an event-based resource allocation algorithm for subcarrier assignment, where the OFDMA scheduler builds the resource allocation table (RB) by solving the resource allocation optimization problem that maximizes the node rate over the available subcarriers. The application layer consists of $n\,V$ wireless nodes located within the AP coverage area. As a result of the limited number of subcarriers, the AP needs to assign the subcarriers to the nodes that have a higher effect on the convergence of the distributed optimization problem. We introduce the sensitivity-based event-triggered scheme into the problem solver, where each node i approximates the effect of its state update on its neighbor $j \in \mathcal{N}_i$ state by computing and exchanging the state sensitivity with respect to the changes on its neighbors' state.

We consider the multi-vehicle wireless distributed system depicted in Fig. 6.5. The detailed system block diagram consists of an AP and a set of vehicle clusters. The AP collects the SNR_i measurements from nodes $i \in \mathcal{N}$ over all subcarriers $s \in \mathcal{K}$, and RTS_i requests. Then the AP computes the weight w_i of node i based on its state sensitivity S_i^j on the approximated state $\tilde{x}_j$ of its neighbor j. The AP solves the resource allocation optimization problem and optimally assigns the subcarrier s with maximum data rate to node i. In the meanwhile, the AP periodically performs the uplink phase and the downlink operation during each time frame according to the system specifications as follows:

1. Uplink phase: The AP collects the transmitted data from each node, including RTS, SNR measurements, and the updated state x_i. At the same time, the AP solves the resource allocation problem and rebuilds the RB table.

2. Downlink phase The access point broadcasts the state x_i to the neighbors $j \in N_i$ within each cluster and broadcasts the resource allocation information for the next time frame for all connected nodes.

As a result, the AP needs to assign the subcarriers to the nodes that have a higher effect on the optimal solution of the overall optimization problem at the application layer and manage the assignment of the communication resources with respect to the performance of the application layer.

The detailed multi-vehicle wireless distributed system is depicted in Fig. 6.5. The block diagram consists of an AP and a set of vehicle clusters. The AP collects the SNR_i measurements from nodes $i \in \mathcal{N}$ over all subcarriers $s \in \mathcal{K}$, and RTS_i requests. Then the AP computes the weight w_i of node i based on its state sensitivity S_i^j on the approximated state $\tilde{x}_j$ of its neighbor j. The AP solves a standard resource allocation optimization problem to optimally assign a subcarrier with maximum data rate to node i. Generally, we assume that each subcarrier is assigned to only one node.

6.3.1 Application layer: optimal control problem of vehicle dynamics and subgradient solver

We consider the distributed event-triggered optimization scheme presented in Fig. 6.5. There, we consider the vehicle dynamics optimal control problem (5.7) and its distributed form (5.15), which consists of achieving the smallest possible utilization of the adhesion potential η_i and keeping it below the physical adhesion limit. We added the equality constraint h_2, which guarantees that all tires experience the same adhesion potential and which is formulated as follows:

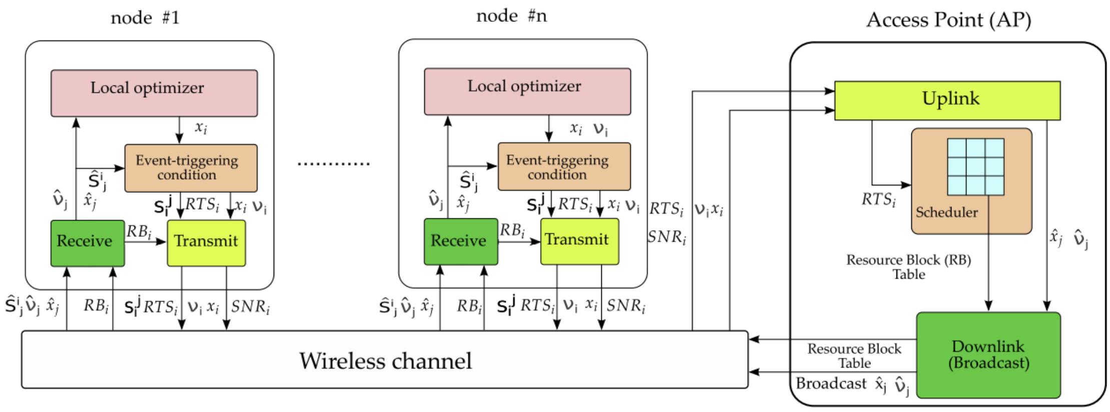

FIGURE 6.5: OFDMA-based event-triggered distributed wireless network system. Note that the resource allocation at the Access Point (AP) is based upon the state evolution at the local optimizer controllers. Following the sensitivity-based policy, more resources are allocated to nodes whose states are associated with higher sensitivity of the objective function w.r.t. the information exchange.

$$
\begin{aligned}
\underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \quad & J_0 = \sum_{i=1}^{N} \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right), \\
\text{subject to} \quad & f_1 = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f \le 0, \\
& f_2 = \eta_i - \mu_{\max} \le 0, \\
& h_1 = A_x \hat{F}_{xj} + A_y \hat{F}_{yj} - Y_d = 0. \\
& h_2 = \eta_i - \hat{\eta}_j = 0, j \in \mathcal{N}_i,
\end{aligned}
\tag{6.14}
$$

The distributed sub-problem i was defined as:

$$
\begin{aligned}
\underset{\eta_i, F_{xi}, F_{yi}}{\text{minimize}} \quad & f_0^i = \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right), \\
\text{subject to} \quad & f_1^i = \sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_f^i \le 0, \\
& f_2^i = \eta_i - \mu_{\max} \le 0, \\
& h_1^i = A_x \hat{F}_{xj}^i + A_y \hat{F}_{yj}^i - Y_d = 0. \\
& h_2^i = \eta_i - \hat{\eta}_j^i = 0, j \in \mathcal{N}_i,
\end{aligned}
\tag{6.15}
$$

where η_i is the adhesion potential, F_{xi} and F_{yi} are the longitudinal and lateral forces, respectively, $\epsilon \ll 1$ is a regularization term added to the cost function, A_x and A_y are matrices defined by the vehicle's geometric parameters, Y_d is the reference trajectory, $\hat{F}_{xj}$, $\hat{F}_{yj}$ are the received longitudinal and lateral forces vectors, N_i is the normal force, and $\mu_{\max}$ is the maximum friction coefficient parameter. The optimization problem (6.14) is distributed into $i = 1, \ldots, N$ sub-problems, and the Lagrangian L_i of sub-problem i is:

$$L_i(\eta_i, F_{xi}, F_{yi}, \lambda_i, \sigma_i, \nu_i, \theta_i) = \eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right) + \lambda_i(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_i) + \sigma_i(\eta_i - \mu_{\max}) + \nu_i^T (A_x \hat{F}_{xj} + A_y \hat{F}_{yj} - Y_d) + \theta_i(\eta_i - \hat{\eta}_j), \forall j \in \mathcal{N}_i, \quad (6.16)$$

where η_i, F_{xi}, and F_{yi} are the primal variables, and $\lambda_i, \sigma_i, \nu_i$, and θ_i are the dual multipliers. The corresponding dual function $g_i(\lambda_i, \sigma_i, \nu_i, \theta_i) = \text{Inf}_{\eta_i, F_{xi}, F_{yi}} L_i(\eta_i, F_{xi}, F_{yi}, \lambda_i, \sigma_i, \nu_i, \theta_i)$ of the distributed sub-problem i is written as:

$$g_i(\lambda_i, \sigma_i, \nu_i, \theta_i) = \inf\big(\eta_i^2 + \epsilon^2 \left(F_{xi}^2 + F_{yi}^2\right) + \lambda_i(\sqrt{F_{xi}^2 + F_{yi}^2} - \eta_i N_i) + \sigma_i(\eta_i) + \nu_i^T (A_x \hat{F}_{xj} + A_y \hat{F}_{yj}) + \theta_i \eta_i - \nu_i^T Y_d - \sigma_i \mu_{max} - \theta_i \hat{\eta}_j, \forall j \in \mathcal{N}_i. \quad (6.17)$$

We use the subgradient method to solve the dual problem (6.17) and perform the subgradient update of the dual variables $\lambda_i, \sigma_i, \nu_i$ and θ_i associated with each sub-problem i as follows:

$$\begin{aligned} \lambda_i[k+1] &= (\lambda_i[k] - \alpha f_1^i[k])_+, \\ \sigma_i[k+1] &= (\sigma_i[k] - \alpha f_2^i[k])_+, \\ \nu_i[k+1] &= \nu_i[k] - \alpha h_1^i[k], \\ \theta_i[k+1] &= \theta_i[k] - \alpha h_2^i[k], \end{aligned} \quad (6.18)$$

where $i = 1, \ldots, n$, $x_+ := \max(0, x)$, and α is the step size [93]. The subgradients of $f_1^i[k], f_2^i[k], h_1^i[k]$, and $h_2^i[k]$ are defined as:

$$\begin{aligned} f_1[k] &= -f_1^i(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \\ f_2[k] &= -f_2^i(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \\ h_1[k] &= -h_1^i(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \\ h_2[k] &= -h_2^i(\eta_i^*[k], F_{xi}^*[k], F_{yi}^*[k]), \end{aligned} \quad (6.19)$$

where

$$\begin{aligned} \eta_i^*[k] &= \eta_i(\lambda_i[k], \sigma_i[k], \nu_i[k], \theta_i[k]), \\ F_{xi}^*[k] &= F_{xi}(\lambda_i[k], \sigma_i[k], \nu_i[k], \theta_i[k]), \\ F_{yi}^*[k] &= F_{yi}(\lambda_i[k], \sigma_i[k], \nu_i[k], \theta_i[k]), \end{aligned} \quad (6.20)$$

are the right-side expressions representing analytical solutions to

$$(\eta_i^*, F_{xi}^*, F_{yi}^*) = \arg\ \text{inf}_{\eta_i, F_{xi}, F_{yi}} L_i(\eta_i, F_{xi}, F_{yi}, \lambda_i[k], \sigma_i[k], \nu_i[k], \theta_i[k])$$

, which is computed as follows:

$$\begin{aligned}
\eta_i^*(\lambda_i, \sigma_i) &= \frac{1}{2}\left(\lambda_i N_f - \sigma_i - \theta_i\right), \\
F_{xi}^*(\lambda, \nu) &= \frac{\kappa_{xi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right), \\
F_{yi}^*(\lambda, \nu) &= \frac{\kappa_{yi}}{2\epsilon^2}\left(\frac{\lambda_i}{\sqrt{\kappa_{xi}^2 + \kappa_{yi}^2}} - 1\right),
\end{aligned} \tag{6.21}$$

with

$$\kappa_{xi} = \sum_{\ell=1}^{3} \nu_\ell A x_{\ell i}, \qquad \kappa_{yi} = \sum_{\ell=1}^{3} \nu_\ell A y_{\ell i}. \tag{6.22}$$

6.3.2 Communication layer: node-weight-based resource allocation problem.

According to our findings, the performance of the application layer is highly correlated with the communication layer activities, and depends on the collaboration between the execution of the application at the application layer and the assignment of the communication resources at the communication layer; refer to Fig. 6.4 and Fig. 6.5. Therefore, we propose mapping the sensitivity of the optimization problem at the application layer to the resource allocation problem, which can be implemented by computing the weight of node w_i with respect to the node's sensitivity and utilizing it in the cost function of the resource allocation problem. By doing this, we couple the effect of the data exchange of each sub-problem at the application layer with the assignment mechanism of the communication resources at the OFDMA scheduler. In particular, the weight w_i of node i is used to reflect the effect of the change of state x_i of node i on the convergence of the overall optimization problem, wherein the sensitivity effect of node i on its neighbors' sub-problems is encapsulated in the approximated state $\tilde{x}_j^i$ of each neighbor $j \in \mathcal{N}$. Consequently, the value of the weight w_i of node i should be proportional to the incremental participation of node i (i.e., of the Δx^i) in the entire cost increment ΔJ_0^i (6.12). It is computed as the ratio of the changes of the sub-problem of node i with respect to the overall change caused by the state x_i as follows:

$$w_i = \frac{\|\nabla_{x_i} f_i^0 \, \Delta x_i\|}{\|\Delta \tilde{J}_0^i\|}, \tag{6.23}$$

where the term $\|\nabla_{x_i} f_i^0 \Delta x_i\|$ presents the approximated change of the cost function of sub-problem i, and $\Delta \tilde{J}_0^i$ refers to the change of the overall cost function with respect to the approximated state $\tilde{x}_j^i$.

6.3.3 Resource allocation optimization problem

We consider a standard resource allocation optimization problem such as discussed in [108, 109], where the cost function maximizes the node rate based on the measured Signal to Noise Ratio $SNR = e_{is}$ of each node i over all available subcarriers $s \in \mathcal{K}$, and with respect to the constraints on the power p_{is} assigned to each node i over the subcarrier s. Moreover, we added a constraint on the sum of the power p_{is} allocated to node i over all subcarriers $s \in \mathcal{K}$ so that it does not exceed the maximum allowable power $\mathcal{P}_i$. Here, the power constraints are defined as follows:

$$\sum_{s} p_{is} \leq \mathcal{P}_i, \quad i \in \mathcal{N} \tag{6.24}$$

where $\mathcal{P}_i$ is the maximum power assigned to node i over the subcarriers $s \in \mathcal{K}$. With respect to that, the subcarrier s is assigned to only one node i at each time slot. In order to use the maximum capacity of the wireless channel, the aim of the cost function is to maximize the achievable rate r_i of node i over subcarrier s with respect to the measured Signal to Noise Ratio $= e_{is}$ as follows:

$$r_i = z_{is} \log\left(1 + p_{is} e_{is}\right), \quad i \in \mathcal{N} \tag{6.25}$$

where r_i is the rate of node i, z_{is} is the set of subcarriers s assigned to node i, p_{is} is the power used to transmit the amount of the data rate r_i, and e_{is} is the SNR of node i over subcarrier s. The following optimization problem for maximizing the data rate of each node i with respect to the SNR measured over all subcarriers $s \in \mathcal{K}$ is formulated as follows:

$$\begin{aligned} \underset{p_{is}, z_{is}}{\text{maximize}} \quad & \sum_{i} w_i \sum_{s} z_{is} \log\left(1 + p_{is} e_{is}\right), \\ \text{subject to} \quad & \sum_{s} p_{is} \leq \mathcal{P}_i, \quad i \in \mathcal{N} \\ & p_{is} \geq 0, \quad i \in \mathcal{N}, s \in \mathcal{K} \\ & X_i \cap X_j = \emptyset, \quad i \neq j, i, j \in \mathcal{N} \end{aligned} \tag{6.26}$$

where w_i is a weight associated with node i. A higher weight implies that the node will be assigned more communication resources. p_{is} is the power assigned to node i over subcarrier $s \in \mathcal{K}$, z_{is} indicating the set of subcarriers s assigned to node i only; X_i and X_j are the set of subcarriers assigned to nodes i and j, and e_{is} is the SNR measured by node i over subcarrier s.

6.4 Simulation and discussion

In order to evaluate the proposed event-triggered OFDMA resource allocation protocol, an extensive simulation study was carried out in a distributed multi-vehicle dynamics scenario. First, we point out that the simulation study considered the convergence of the distributed optimization problem at the application layer and the communication behavior of the interconnected nodes according to the optimal resource allocation algorithm at the AP. The efficiency of the proposed algorithm was evaluated in terms of communication reduction, convergence to the optimal solution, and optimal resource allocation. In the application layer, a set of $V = 1, 2, 3$ clusters was used, each consisting of $n = 4$ nodes, with each node internally solving a distributed vehicle dynamics optimization problem. A lane change maneuver under braking conditions was used as a reference trajectory for the vehicle dynamics optimization problem. The parameterization of the maneuver scenario is defined with a maximum longitudinal deceleration of $a_x \approx 3$ m/s^2 and maximum lateral acceleration of $a_y \approx 8$ m/s^2 at a speed of $v = 120$ km/h under dry road conditions with $\mu_{max} = 1$. The distributed dual subgradient method was used with step size $\alpha = 0.1/\sqrt{k}$ and the triggering parameters of the event-triggered condition were set to $\beta_0 = 0.004$ and $\beta_1 = 0.00001$.

The convergence rate of the distributed optimization problem was computed according to the relative error $= \frac{\|x[k+1]-x^*\|}{\|x^*\|}$. The relative error traces the performance of the

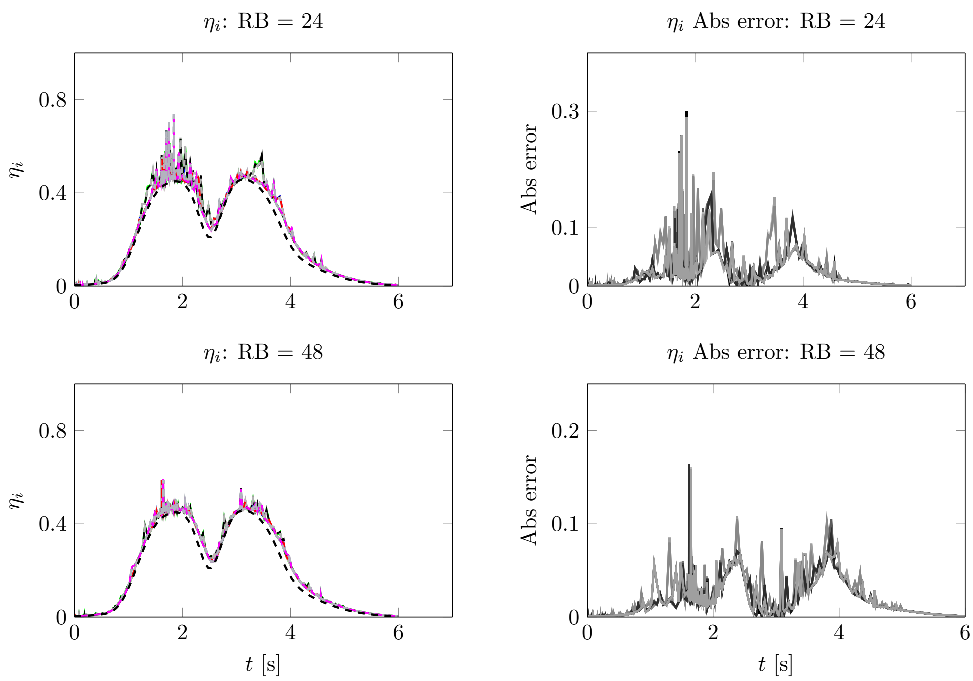

FIGURE 6.6: Performance of the algorithm over the complete maneuver: η_i for all nodes (top) and absolute error (bottom) for the AP for capacities with (RB = 24) and (RB = 48). Naturally, larger resource capacities yield smaller errors and improved performance.

proposed distributed algorithm by computing the convergence rate of η_i with respect to the optimal adhesion potential η_i^* computed by the centralized algorithm.For the OFDMA wireless network, we used LTE standard AP parameters. The access point operated on the frequency band $B = \{5, 10\}$ MHz, which provides a set of resource blocks $RB = \{24, 48\}$. The SNR was randomly distributed over the subcarriers with a maximum value up to 130 dB.

Fig. 6.6 presents the computed η_i for $i = 1, ..., 12$ of the three clusters and the absolute error computed with reference to the centralized optimal algorithm. We see that the algorithm's performance is highly improved, with an increasing number of resource blocks ($RB = 48$) and a decrease in the absolute error value. We notice that with small numbers of subcarriers ($RB = 24$), the resource allocation algorithm maintains the system performance with a noticeable increase in the absolute error value, especially in the extreme driving maneuver instance.

Fig. 6.7 presents the effect of the optimal node weights on the resource allocation. Fig. 6.7-(a) shows an example of the computed weights of node 1, while Fig. 6.7-(b) shows the corresponding number of resource blocks assigned to node 1 according to the changes in its weight. Fig. 6.7-(c) presents the sum of the optimal rates assigned to the node over the assigned subcarriers, while Fig. 6.7-(d) depicts the total optimal power assigned to the node over the assigned subcarriers. Here, we comment that the changes in the weight of node 1 highly affected its communication activities with respect to the assigned subcarriers, rate, and power.

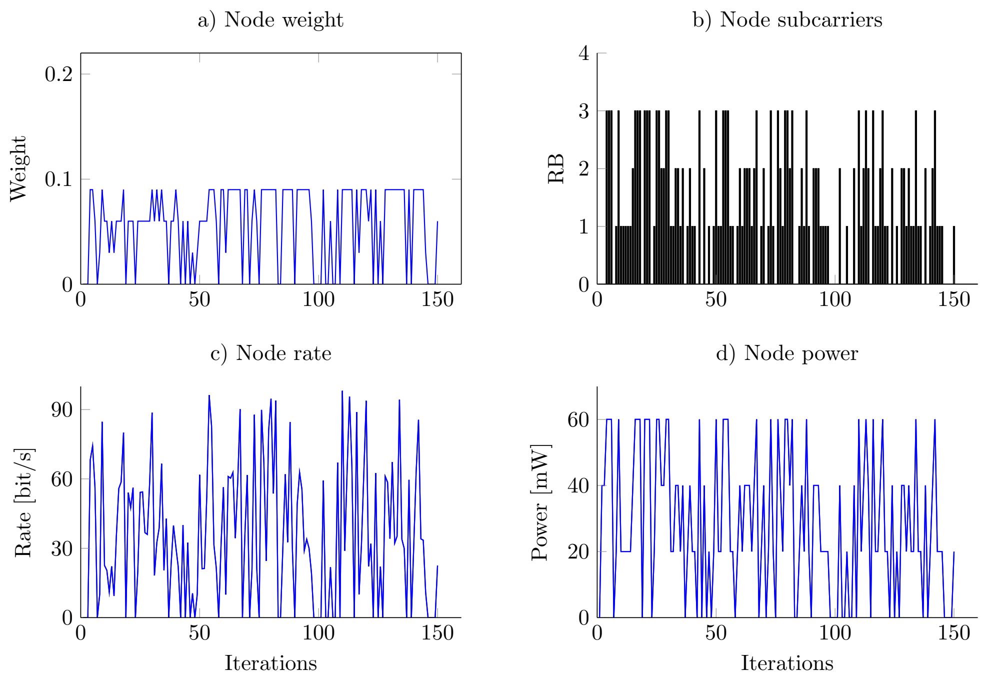

FIGURE 6.7: Weights, assigned rate, and power of Node 1 related to the number of subcarriers for an AP with a capacity of 24 resource blocks.

Fig. 6.8 presents the nodes' communication requests RTS and the communication ac-

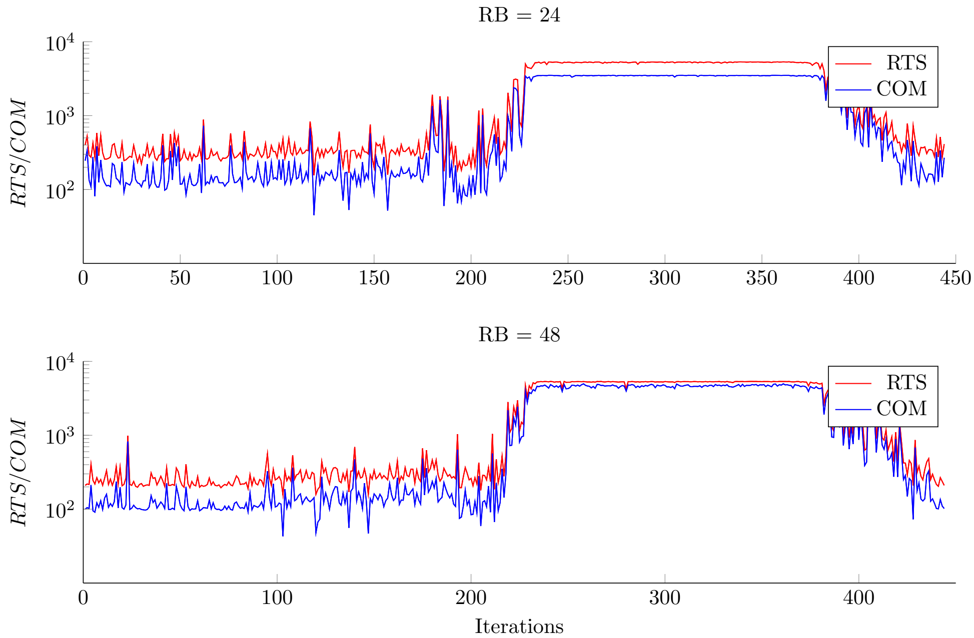

Figure 6.8: RTS and communication activities for all nodes; AP with 24 and 48 resource blocks.

tivities in the case of the AP operating on (RB = 24) and (RB = 48) capacity. Fig. 6.9 shows the communication request analysis of the three clusters over an extreme maneuver instance and the AP operating on (RB = 24) and (RB = 48) capacity. We observe that in the extreme maneuver, the resource demands are increased, and the scheduler assigns the available resources to those nodes that have a higher weight. Fig. 6.9-(a) presents the total number of RTS of all nodes, and Figs. 6.9-(b), 6.9-(c), and 6.9-(d) show the transmission activities of the clusters 1,2, and 3.

Finally, Fig. 6.10 presents the relative error measure of the convergence rate of η_i for each node within the three clusters. The communication is conducted with the AP operating on (RB = 24) and (RB = 48). Observe that the convergence rate is improved when more resource blocks are used, while the performance of the system is maintained even with fewer communication resources by optimally assigning the communication resources to the nodes.

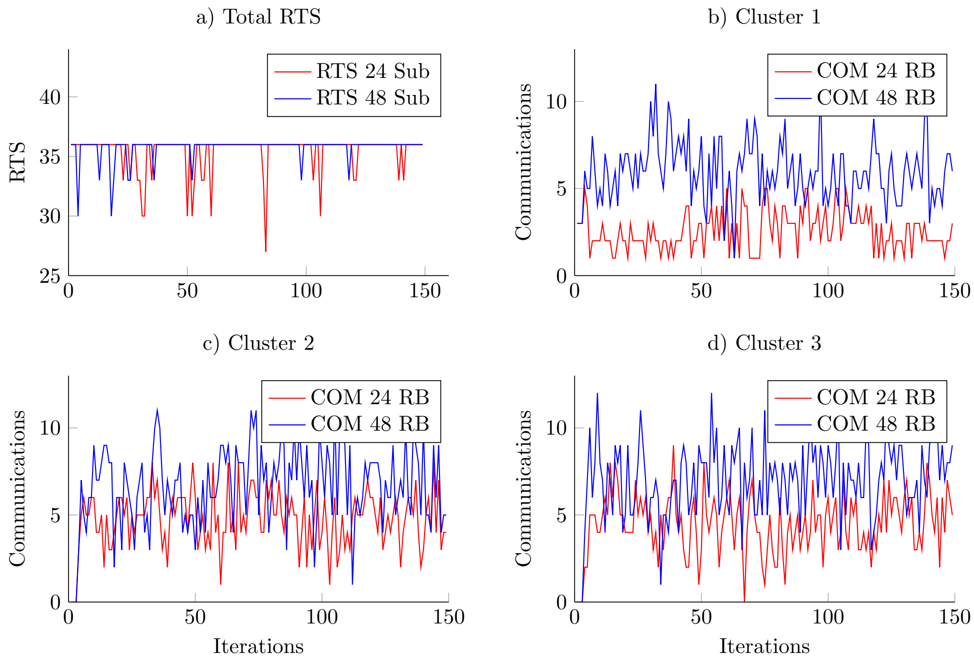

FIGURE 6.9: RTS and communication intensity for the whole network involving three clusters and individual correspondents for an AP with a capacity of (RB = 24) and (RB = 48). Less capacity naturally leads to lower communication intensity.

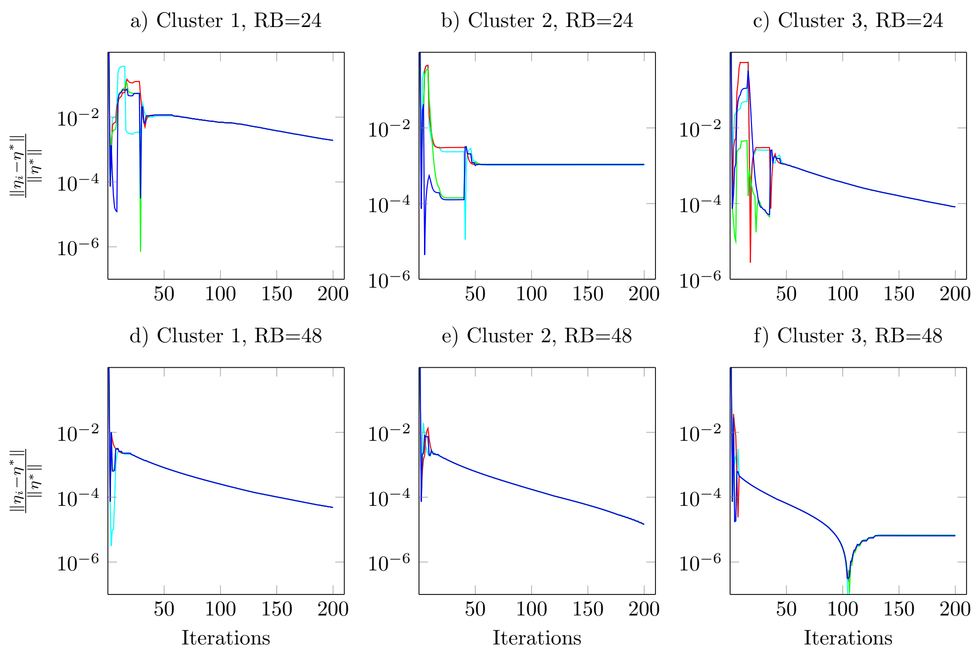

FIGURE 6.10: Convergence rate of η_i for the three clusters with the AP operating on 24 resource blocks (top) and 48 resource blocks (bottom). Obviously, larger resource block capacity produces solutions with smaller errors.

6.5 Chapter conclusion

In this chapter, we presented an event-triggered adaptive OFDMA resource allocation protocol with application to multi-vehicles. In particular, the sensitivity of the distributed optimization problem was mapped to the weighting factor of the node in the resource allocation optimization problem. While this idea has been borrowed from earlier work in the literature, our main contribution consists of its application to the design of adaptive OFDMA protocols. The event-triggered scheme couples the communication and application layers of the proposed OFDMA protocol. We have demonstrated the utilization of the protocol in a multi-vehicle case study, but in principle, it can be readily extended to other network control systems as well. The simulation results demonstrate that the proposed protocol for communication reduction combined with the optimal resource allocation of the available subcarriers maintains the system performance even during critical driving maneuvers, which are typically associated with high resource demands.

Algorithm 7: Adaptive OFDMA protocol with sensitivity based event-triggered resource allocation

initialization;
set $\mathcal{N} = \{1, ..., n\}$; $\mathcal{N}_i = \{1, ..., N_i\}$;
while *subgradient* **do**
  For all nodes;

1. **Update** x_i, ν_q^i, and λ_p^i ;
2. **Compute** sensitivity S_i^j :
   $S_i^j = M_i^{-1} N_i^j$;
3. **Transmit** sensitivity S_i^j, $\forall j \in \mathcal{N}_i$:
4. **Compute** state approximation $\tilde{x}_j^i[k+1]$:
   $\tilde{x}_j^i[k+1] = \hat{x}_j^i[k] + \hat{S}_i^j(\tilde{x}_i[k+1] - \hat{x}_i^j[k])$;
5. **Compute cost function difference** $\Delta\tilde{J}_0^i$:
   $\Delta\tilde{J}_0^i \cong \nabla_{x_i} f_i^0(x_i, \hat{x}_j^i)\Delta x_i + \sum_{j\in\mathcal{N}_i} \nabla_{x_j} f_j^0(\tilde{x}_j^i, \hat{x}_i^j)\Delta\tilde{x}_j^i$;
6. **Assign** node weight $w_i = \frac{\|\nabla_{x_i} f_i^0 \, \Delta x_i\|}{\|\Delta\tilde{J}_0^i\|}$;
7. **Event-triggered condition**:
   if $\left\|\Delta\tilde{J}_0^i\right\| \geq \beta_0 \left\|h_q(x_i, \tilde{x}_j^i)\right\| + \beta_1$ **then**
     $RTS_i \leftarrow True$;

  Access Point:;

1. **Collect** w_i, e_{is}, RTS_i, $\forall i \in \mathcal{N}$, $\forall s \in \mathcal{K}$;
2. **Solve** resource allocation problem:
   $$\underset{p_{is}, z_{is}}{\text{maximize}} \quad \sum_i w_i \sum_s z_{is} \log\left(1 + \frac{p_{is} e_{is}}{z_{is}}\right);$$
3. **Update** the resource allocation table (RB);
4. **Broadcast** the RB table, $\forall\ i \in \mathcal{N}$;
5. **Broadcast** x_i, $\forall\ j \in \mathcal{N}_i$;

end

Chapter 7

Conclusion

In this work, we tackled different research topics in the context of wireless networked control systems as well as the optimal allocation of communication resources with regard to the convergence of the distributed optimization problem. The main topic focused on solving a distributed vehicle dynamics control problem over a wireless communication channel and studying the effect of closing the control loop through a wireless link. We investigated the effect of using different medium access control protocols on the performance of the proposed solution algorithms.

First, we formulated a distributed vehicle dynamics optimal control problem, where we defined a convex optimization problem in the context of a fully decomposed feed-forward control scheme including independent single actuation units for the manipulation of the vehicle dynamics. Basically, each such unit consists of a local optimization task for the optimal control problem that penalizes the maximal utilization of the tire's adhesion for a given driving maneuver. The problem formulation provided equal distribution of the adhesion utilization to the vehicle tires. The proposed convex optimization problem was decomposed into a distributed form and solved within cooperating nodes that conduct data exchange over a uni-directional and bi-directional communication protocol. We proposed a distributed solution algorithm consisting of a projected subgradient consensus method that was implemented in a distributed manner by exchanging the primal variables between the connected nodes. Thus, the algorithm is suitable for power-train systems of electric cars equipped with single-actuating wheel-hub drives. For a prescribed maneuver, the optimization policy provides equal adhesion utilization among the vehicle tires. This prevents the saturation of the tires and yields an increased safety reserve in most driving situations.

In the second phase, we introduced a wireless communication protocol within the system structure, where a set of distributed wireless nodes was equipped with the projected subgradient consensus solver and communicate over a wireless channel. We mainly introduced the concept of using wireless technology for guidance and control of a redundantly actuated electric car supported by an on-board wireless network of sensors, actuators, and control units. The concept was validated by an extensive analysis of TDMA schemes in connection with a vehicle dynamics guidance and stabilization problem. Furthermore, we investigated the effect of the TDMA medium access control protocol on the convergence of the distributed optimization problem. We found that the implementation of the consensus algorithm requires extensive consumption of communication resources and, based on the implementation of the TDMA protocol with sequential time slot assignment, induces a commutative time delay, so serving all nodes will consume much more communication resources. Therefore, an event-triggered scheme was implemented within the solution algorithm in order to reduce the communication activity. This event-triggered scheme allows only nodes whose internal state error exceeds a predefined state error threshold. More specifically, the implementation of the distributed event-triggered algorithm implies that each node broadcasts its state variable only if its internal state error exceeds the predefined threshold. The introduced event-triggered layer was implemented in order to reduce communication between nodes, maintain the system performance, and guarantee convergence of the algorithm to some near-optimal solution. The simulation results showed that a communication reduction of up to 40% was achieved. We believe that this venture represents an important step towards applying the proposed solution algorithm in real time. The proposed event-triggered scheme provides communication control with respect to the internal state error of the node and mainly maintained the convergence of the distributed optimization problem with slight errors, which were regulated by the triggering parameters.

In the third phase, we extended the event-based communication approach with respect to optimal resource allocation, where event-based mechanisms were invoked to introduce purposeful reduction of the communication resource consumption in accordance with the dynamic evolution of the system states. The focus of the event-based wireless communication was on the context of distributed wireless systems where the set of nodes are connected over a wireless network based on OFDMA medium access control. In fact, we here extended thes ideas in two directions. The first direction refers to the usage of OFDMA wireless communication, where we addressed an adaptive OFDMA resource allocation protocol based on the event-triggered approach combined with a sensitivity analysis of the underlying optimization problem. The second extension is related to the out-board application of a control scenario for multi-vehicle clusters.

We considered a multi-vehicle distributed system with a set of vehicles presented as clusters of wireless nodes communicating over an OFDMA-based wireless network. The network structure includes an AP that manages data transmission and assignment of the communication resources to the nodes within the vehicle clusters. We proposed an adaptive OFDMA resource allocation algorithm by utilizing an optimal resource allocation problem with a cost function that maximizes the data rate of each node over the available subcarriers. Our contribution consisted in updating the cost function with the node weight computed according to a sensitivity analysis of the effect of a node's neighbors' state on its own state.

The optimal allocation of the communication resources algorithm combined with the sensitivity-based event-triggered scheme provides a noticeable reduction of the number of information exchange requests within the network, and links the communication layer and the application layer by optimally assigning the communication resources to nodes with respect to the evolution of their state affected by their neighbors. It also maintains the convergence of the overall distributed optimization problem with an acceptable margin of error. Basically, the algorithm assigns subcarriers with a lower SNR and a high data rate to nodes that have a higher effect on their neighbors' state. It was tested by performing an extensive simulation study on a multi-vehicle scenario with three vehicle clusters. The simulation results showed optimal allocation of the communication resources and an improvement of the convergence rate of the distributed optimization problem with respect to the number of communication resources available within the AP coverage area.

Finally, we conclude that the MAC protocols used in the wireless communication network has a high effect on the convergence of the distributed optimization problem, and that the proposed sensitivity-based event-triggered scheme is an efficient method for reducing the data exchange and provides optimal allocation of the communication resources. Moreover, mapping the application layer of the distributed sub-problems to the resource allocation scheme at the communication layer improved the system performance. The simulation study showed that the distributed vehicle dynamics optimization problem, which was used as a benchmark, proved to be a good example for different driving scenarios, and the proposed solution algorithms were efficiently suited for the considered communication protocols.

Bibliography

[1] A. Goldsmith. *Wireless Communications.* Cambridge University Press, New York, NY, USA, 2005.

[2] Jiang T., L. Song, and L. Zhang. *Orthogonal Frequency Division Multiple Access Fundamentals and Applications.* Taylor and Francis Group., Boca Raton, FL, USA, 2010.

[3] A. Molisch. *Wireless Communications.* Wiley-IEEE Press, 2005.

[4] M. Hasan, H. Yu, A. Carrington, and T. Yang. Co-simulation of wireless networked control systems over mobile ad hoc network using simulink and opnet. *Communications, IET*, 3(8):1297–1310, August 2009.

[5] P. Naghshtabrizi and J. Hespanha. Implementation considerations for wireless networked control systems. In S. Mazumder, editor, *Wireless Networking Based Control*, pages 1–27. Springer New York, 2011.

[6] F. Wang and D. Liu. *Networked control systems. Theory and applications. In memory of George Nikolaos Saridis.* London: Springer, 2008.

[7] G. Walsh and Hong Y. Scheduling of networked control systems. *Control Systems, IEEE*, 21(1):57–65, Feb 2001.

[8] W. Zhang, M. Branicky, and S. Phillips. Stability of networked control systems. *Control Systems, IEEE*, 21(1):84–99, Feb 2001.

[9] J. Wang, B. Shi, J. Bai, J. Gao, and X. Fang. Stabilization of a wireless networked control system with packet loss and time delay: An ads approach. *J. Control Sci. Eng.*, 2012:6:6, Jan 2012.

[10] B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 676–681, Dec 2003.

[11] F. Yang and H. Fang. Control structure design of networked control systems based on maximum allowable delay bounds. *Journal of the Franklin Institute*, 346(6):626 – 635, 2009.

[12] Jing Wang, Bingxin Shi, Jianjun Bai, Jinfeng Gao, and Xiaosheng Fang. Stabilization of a wireless networked control system with packet loss and time delay: an ads approach. *Journal of Control Science and Engineering*, 2012, 2012.

[13] L. Xiangheng and A. Goldsmith. Wireless network design for distributed control. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2823–2829 Vol.3, Dec 2004.

[14] X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2823–2829 Vol.3, 2004.

[15] G. Goodwin, H. Haimovich, D. Quevedo, and J. Welsh. A moving horizon approach to networked control system design. *Automatic Control, IEEE Transactions on*, 49(9):1427–1445, Sept 2004.

[16] M. HASAN, H. Yu, A. Griffiths, and T. Yang. Co-simulation framework for networked control systems over multi-hop mobile ad-hoc networks. In *IFAC 17th World Congress The International Federation of Automatic Control*, 2008.

[17] D. Westhoff, J. Girao, and M. Acharya. Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation. *IEEE Transactions on Mobile Computing*, 5(10):1417–1431, Oct 2006.

[18] A. Zomaya and Y. Lee. *Energy Efficient Distributed Computing Systems*. Wiley-IEEE Computer Society Pr, 1st edition, 2012.

[19] J. Baillieul and P. Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9–28, Jan 2007.

[20] V. Liberatore. Integrated play-back, sensing, and networked control. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[21] T. Aysal, M. Coates, and M. Rabbat. Rates of convergence of distributed average consensus using probabilistic quantization. In *in Proc. of the Allerton Conference on Communication, Control, and Computing*, 2007.

[22] B. O'Hara and A. Petrick. *IEEE 802.11 Handbook : a designer's companion*. IEEE standards wireless networks series. IEEE, New York, 2005.

[23] H. Chen and M. Guizani. *Next Generation Wireless Systems and Networks*. John Wiley and Sons., New Jersey, USA, 2006.

[24] J. Chen, K. Sivalingam, and P. Agrawal. Performance comparison of battery power consumption in wireless multiple access protocols. *Wireless Networks*, 5 (6):445–460, 1999.

[25] M. Ilyas, I. Mahgoub, and L. Kelly. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, Inc., Boca Raton, FL, USA, 2004.

[26] Nick LaSorte, W Justin Barnes, and Hazem H Refai. The history of orthogonal frequency division multiplexing. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pages 1–5. IEEE, 2008.

[27] L. Zhang and P. Sun. Introduction to ofdma. In T. Jiang, L. Song, and Y. Zang, editors, *Orthogonal Frequency Division Multiple Access Fundamentals and Applications*, chapter 6, pages 131–164. Auerbach Publication, Boston, MA, USA, 2010.

[28] Michele Morelli, C.-C. Jay Kuo, and Man-On Pun. Synchronization techniques for orthogonal frequency division multiple access (ofdma): A tutorial review. *Proceedings of the IEEE*, 95(7):1394–1427, 2007. doi: 10.1109/JPROC.2007.897979.

[29] R. Parasad. *OFDM for wireless communications systems*. Artech House, 2004.

[30] D. Hristu-Varsakelis and W. Levine, editors. *Handbook of networked and embedded control systems*. Control engineering. Birkhäuser, Boston, 2005.

[31] A. Hac. *Wireless Sensor Network Designs*. Wiley, addr:wiley, Oct 2003.

[32] N. Ploplys, P. Kawka, and A. Alleyne. Closed-loop control over wireless networks. *Control Systems, IEEE*, 24(3):58–71, Jun 2004.

[33] Y. Tipsuwan and M. Chow. Control methodologies in networked control systems. *Control engineering practice*, 11(10):1099–1111, 2003.

[34] Mayank Kumar Gautam, Avadh Pati, Sunil Kumar Mishra, Bhargav Appasani, Ersan Kabalci, Nicu Bizon, and Phatiphat Thounthong. A comprehensive review of the evolution of networked control system technology and its future potentials. *Sustainability*, 13(5):2962, 2021.

[35] G. Zheng, W. Zeng, and F. Xu. Design on the optimal feedback control law of networked control systems with multiple switching modes. In *Computational*

Engineering in Systems Applications, IMACS Multiconference on, volume 1, pages 564–569, Oct 2006.

[36] Y. Blind, U. Münz, and F. Allgöwer. Modeling, analysis, and design of networked control systems using jump linear systems (modellierung, analyse und entwurf vernetzter regelsysteme mithilfe schaltender systeme). *Automatisierungstechnik*, 56(1):20–28, 2008.

[37] M. Gopal. *Modern control system theory.* New Age International (P)Ltd, New Delhi, 2005.

[38] S. Misra, S. Misra, and I. Woungang. *Guide to wireless sensor networks*, volume 7. Springer New York, NY, 2009.

[39] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and V. Poor. *MIMO Wireless Communications.* Cambridge University Press, New York, NY, USA, 2007.

[40] M. Fitz, C. Shen, and M. Samuel. The wireless communications physical layer. In G. Stornelli, editor, *Wireless Networks*, pages 13 – 94. Academic Press, Burlington, 2006.

[41] Giorgio Franceschetti and Sabatino Stornelli. *Wireless Networks: From the Physical Layer to Communication, Computing, Sensing and Control.* Elsevier, 2006.

[42] M. Villella. Nonlinear modeling and control of automobiles with dynamic wheel-road friction and wheel torque inputs. In *IFAC Symposium on Advances in Automotive Control*, 2004.

[43]

[44] M. Stone and M. Demetriou. Modeling and simulation of vehicle ride and handling performance. *Proceedings of the 2000 IEEE International Symposium on*, pages 85–90, 2000.

[45] R. Orend. Steuerung der ebenen Fahrzeugbewegung mit optimaler Nutzung der Kraftschlusspotentiale aller vier Reifen (vehicle dynamics feedforward control with optimal utilisation of the adhesion potentials of all four tyres). *Automatisierungstechnik*, 53(1):20–27, 2005.

[46] D. Kirk. *Optimal control theory an introduction.* Prentice Hall electrical engineering series. Prentice-Hall, Englewood Cliffs, 1970.

[47] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2009.

[48] B. Johansson. *On Distributed Optimization in Networked Systems.* PhD thesis, Royal Institute of Technology (KTH), December 2008.

[49] H. Hindi. A tutorial on convex optimization. In *American Control Conference, 2004. Proceedings of the 2004*, volume 4, pages 3252–3265, June 2004.

[50] J. Nocedal and S. Wright. *Numerical Optimization, Second Edition.* Springer New York, 2006.

[51] C. Langbort, L. Xiao, R. D'Andrea, and S. Boyd. A decomposition approach to distributed analysis of networked systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 3980–3985, Dec 2004.

[52] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[53] D. Palomar and M. Chiang. A tutorial on decomposition methods for network utility maximization. *Selected Areas in Communications, IEEE Journal on*, 24 (8):1439–1451, 2006.

[54] D. Mosk-Aoyama, T. Roughgarden, and D. Shah. Fully distributed algorithms for convex optimization problems. *SIAM Journal on Optimization*, 20(6): 3260–3279, 2010.

[55] Reinhard Diestel. Graph theory. 2005. *Grad. Texts in Math*, 101:867, 2005.

[56] S. Boyd and A. Mutapcic. Subgradient methods. *Lecture notes of EE364b, Stanford University, Winter Quarter*, 2007, 2006.

[57] K. Shor, N.and Kiwiel and A. Ruszcayǹski. *Minimization Methods for Non-differentiable Functions.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[58] A. Nedic and A. Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):4711–4716, Jan 2007.

[59] P. Urban and A. Schiper. Comparing distributed consensus algorithms. *Proc. of the 2004 IASTED Int'l Conf. on Applied Simulation and Modelling*, 57(7): 474–480, July 2004.

[60] T. Aysal, A. Sarwate, and A. Dimakis. Reaching consensus in wireless networks with probabilistic broadcast. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 732–739, Sept 2009.

[61] Ercan M. Sarwate A. Aysal, T. and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 57(7): 2748–2761, July 2009.

[62] R. Olfati and R. Murray. Consensus protocols for networks of dynamic agents. In *American Control Conference, 2003. Proceedings of the 2003*, volume 2, pages 951–956, June 2003.

[63] D. Spanos, R. Olfati-Saber, and R. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*. Prague Czech Republic, 2005.

[64] K. Avrachenkov, M. ElChamie, and G. Neglia. A local average consensus algorithm for wireless sensor networks. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–6, June 2011.

[65] X. Lin, S. Boyd, and S. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67: 33–46, 2005.

[66] R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, 1997.

[67] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4185–4190, Dec 2008.

[68] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, Nov 2009.

[69] A. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, Feb 2008.

[70] H. Bauschke and J. Borwein. Dykstra's alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3):418–443, 1994.

[71] L. Gubin, B. Polyak, and E. Raik. The method of projections for finding the common point of convex sets. *Computational Mathematics and Mathematical Physics*, 7(6):1 – 24, 1967.

[72] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, Jan 2009.

[73] U. Rasheed, M. Ahmed, M. Afridi, and F. Kunwar. Road trajectory mining and autonomous steering control for vision-based unmanned vehicles. In *2010 10th International Conference on Intelligent Systems Design and Applications*, pages 197–202, Nov 2010.

[74] M. Galvani, F. Biral, B. Nguyen, and H. Fujimoto. Four wheel optimal autonomous steering for improving safety in emergency collision avoidance manoeuvres. In *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pages 362–367, March 2014.

[75] R. Marino, S. Scalzi, G. Orlando, and M. Netto. A nested pid steering control for lane keeping in vision based autonomous vehicles. In *2009 American Control Conference*, pages 2885–2890, June 2009.

[76] Ren Sakata, Katsuyoshi Naka, H Nurata, and Susumu Yoshida. Performance evaluation of autonomous decentralized vehicle-grouping protocol for vehicle-to-vehicle communications. In *VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No. 00CH37026)*, volume 1, pages 153–157. IEEE, 2000.

[77] T. Furuyama, Y. Hirayama, and M. Sawada. Performance evaluation of prioritized csma protocol for single-channel roadside-to-vehicle and vehicle-to-vehicle communication systems. In *The 17th Asia Pacific Conference on Communications*, pages 461–466, Oct 2011.

[78] H. Li and F. Nashashibi. Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to see through front vehicle. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 242–247, Oct 2011.

[79] C. Fulford, N. Lie, E. Earon, R. Huq, and C. Rabbath. The vehicle abstraction layer: A simplified approach to multi-agent, autonomous uav systems development. In *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference - 7th International Conference on*, pages 483–487, Oct 2008.

[80] K. Kawano. Applicability of multi-vehicle scheduling problem based on gps tracking records. In *2010 18th International Conference on Geoinformatics*, pages 1–4, June 2010.

[81] N. Bajcinca and Y. Kouhi. Distributed optimization for feedforward global chassis control. In *6th IFAC Symposium" Advances in Automotive Control"*, 2010.

[82] W. Heemels, K. Johansson, and P Tabuada. An introduction to event-triggered and self-triggered control. In *CDC*, pages 3270–3285, 2012.

[83] D. Dimarogonas and E. Frazzoli. Distributed event-triggered control strategies for multi-agent systems. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 906–910, Sept 2009.

[84] W. Heemels, J. Sandee, and P. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International journal of control*, 81(4):571–590, 2008.

[85] P. Wan and M. Lemmon. Event-triggered distributed optimization in sensor networks. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 49–60, April 2009.

[86] P. Wan and M. Lemmon. Distributed network utility maximization using event-triggered augmented lagrangian methods. In *2009 American Control Conference*, pages 3298–3303. IEEE, 2009.

[87] P. Wan and M. Lemmon. Optimal power flow in microgrids using event-triggered optimization. In *American Control Conference (ACC), 2010*, pages 2521–2526, June 2010.

[88] T. Gommans and W. Heemels. Resource-aware mpc for constrained nonlinear systems: A self-triggered control approach. *Systems & Control Letters*, 79:59–67, 2015.

[89] F. Brunner, T. Gommans, W. Heemels, and F. Allgöwer. Communication scheduling in robust self-triggered mpc for linear discrete-time systems. *IFAC-PapersOnLine*, 48(22):132–137, 2015.

[90] D. Dimarogonas, E. Frazzoli, and K. Johansson. Distributed event-triggered control for multi-agent systems. *Automatic Control, IEEE Transactions on*, 57 (5):1291–1297, May 2012.

[91] Wan. P. and M. Lemmon. Optimal power flow in microgrids using event-triggered optimization. In *American Control Conference (ACC), 2010*, pages 2521–2526, June 2010.

[92] Y. Kouhi, S. Guma, and N. Bajcinca. Real-time allocation of tire adhesion forces for electric vehicles. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 172–177, 2012.

[93] R. Madan and S. Lall. Distributed algorithms for maximum lifetime routing in wireless sensor networks. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 2, pages 748–753, Nov 2004.

[94] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[95] P. Saengudomlert. *Optimization for Communications and Networks.* CRC Press, 2011.

[96] H. Khalil. *Nonlinear systems.* Prentice Hall, Upper Saddle River (New Jersey), 2002.

[97] T. Herman and S. Tixeuil. *A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[98] X. Wang and M. Lemmon. Event-triggering in distributed networked control systems. *IEEE Trans. Automat. Contr.*, 56(3):586–601, 2011.

[99] D. Gross, M. Jilg, and O. Stursberg. Event-based communication in distributed model predictive control. *Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik*, 61(7):457–466, Jan 2013.

[100] J. Huang, V. Subramanian, R. Berry, and R. Agrawal. Scheduling and resource allocation in ofdma wireless systems. In T. Jiang, L. Song, and Y. Zang, editors, *Orthogonal Frequency Division Multiple Access Fundamentals and Applications*, chapter 6, pages 131–164. Auerbach Publication, Boston, MA, USA, 2010.

[101] A. Wilton and T. Charity. *Deploying Wireless Networks.* Cambridge University Press, New York, NY, USA, 1st edition, 2008.

[102] A. Ghosh, J. Zhang, J. Andrews, and R. Muhamed. *Fundamentals of LTE.* Prentice Hall Press, Upper Saddle River, NJ, USA, 1st edition, 2010.

[103] H. Junhong and Z. Yongxing. Enhanced rate adaptive resource allocation scheme in downlink ofdma system. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 5, pages 2464–2468, May 2006.

[104] I. Wong and B. Evans. Adaptive downlink ofdma resource allocation. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2203–2207, Oct 2008.

[105] D. Kivanc, L. Guoqing, and H. Liu. Computationally efficient bandwidth allocation and power control for ofdma. *Wireless Communications, IEEE Transactions on*, 2(6):1150–1158, Nov 2003.

[106] C. Mohanram and S. Bhashyam. A sub-optimal joint subcarrier and power allocation algorithm for multiuser ofdm. *Communications Letters, IEEE*, 9(8): 685–687, Aug 2005.

[107] A. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Mathematics in science and engineering. Academic press, New York, 1983.

[108] V. Subramanian, R. Berry, and R. Agrawal. Joint scheduling and resource allocation in cdma systems. *Information Theory, IEEE Transactions on*, 56(5): 2416–2432, May 2010.

[109] J. Huang, V. Subramanian, R. Agrawal, and R. Berry. Joint scheduling and resource allocation in uplink ofdm systems for broadband wireless access networks. *Selected Areas in Communications, IEEE Journal on*, 27(2):226–234, February 2009.

CV: Shaban Guma

Education

2016 – 2023	Ph.D. Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau, Germany
2010 – 2015	Scholarship research, Technical University Berlin, Germany
2001 - 2003	M.Sc. Telematics Application in Education and Training, Twente University, The Netherlands
2003 - 2004	M.Sc. Control System Engineering, HAN University, The Netherlands
1990 - 1995	B.Sc. in Computer Science, Al-Fateh University, Libya

Work Experience

2017 – present	Research assistant and projects leader, Department of Mechatronics in Mechanical and Automotive Engineering (MEC), RPTU University
2004 – 2009	Research assistant, Aljabel Algarbi University, Libya
1996 – 2000	Research assistant, Nalut high institute for Training and Education, Libya