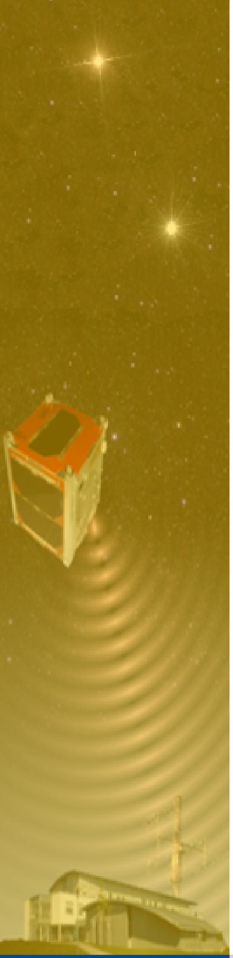


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte
in Robotik und Telematik

Uni Wuerzburg Research Notes
in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 2

Martin Hess

Motion coordination and control in systems of non-holonomic autonomous vehicles

Würzburger Forschungsberichte in Robotik und Telematik

Uni Wuerzburg Research Notes in Robotics and Telematics

Herausgeber

Prof. Dr. K. Schilling
Universität Würzburg
Institut für Informatik
Lehrstuhl für Informatik VII: Robotik und Telematik
Am Hubland
D-97074 Würzburg
Tel.: +49-931-31-86647
Fax.: +49-931-31-86679
email: schi@informatik.uni-wuerzburg.de

ISSN (Internet): 1868-7474
ISSN (Print): 1868-7466

Motion coordination and control in systems of nonholonomic autonomous vehicles

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

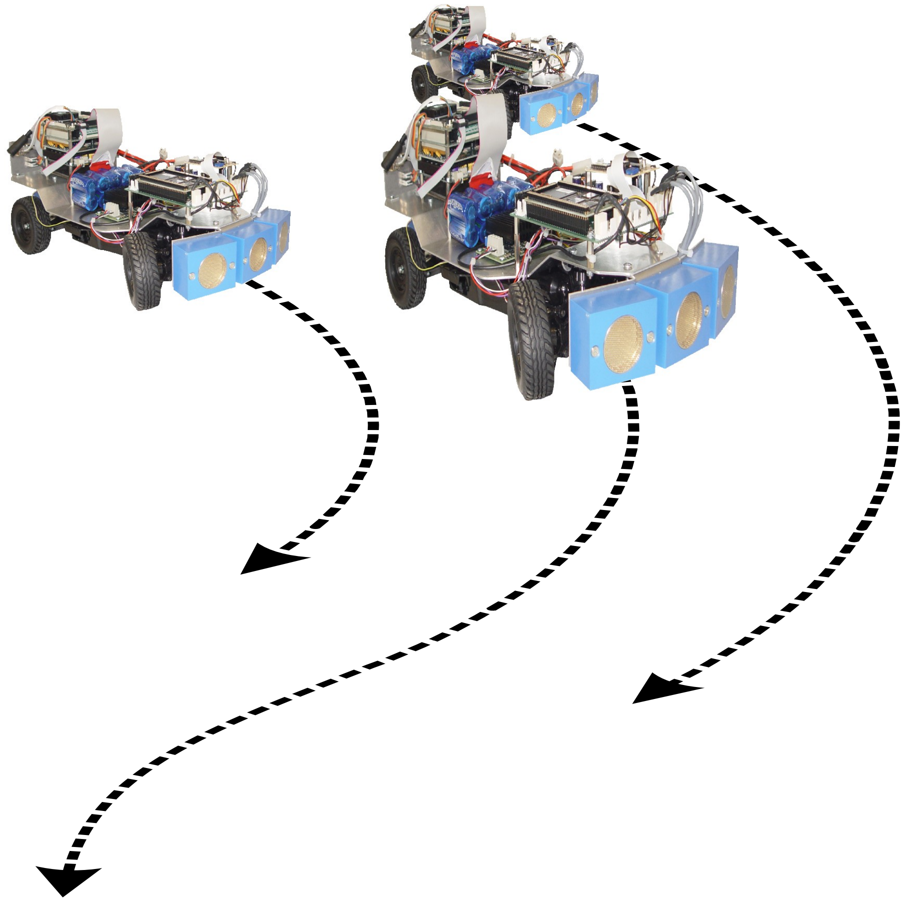
vorgelegt von

Martin Hess

aus

Frankfurt am Main

Würzburg, Juli 2009



Eingereicht am: 15.07.2009

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Klaus Schilling

2. Gutachter: Prof. Dr. Lars Grüne

3. Gutachter: Prof. Dr. Dušan M. Stipanović

Tag der mündlichen Prüfung: 19.02.2010

Acknowledgements

First of all I would like to thank my doctoral supervisor Prof. Dr. Klaus Schilling for giving me the opportunity to work in this interesting field and for providing me with the necessary guidance into scientific research. Especially I appreciate the confidence he put in me during my time as a collaborator at his chair. I would also like to thank Prof. Dr. Lars Grüne and Prof. Dr. Dušan Stipanović who both acted as reviewers of this thesis and provided me with helpful comments and discussions. I am also very grateful for the chance to stay three month at Prof. Stipanović's department at the University of Illinois at Urbana-Champaign (UIUC). Furthermore I want to express my gratitude to Prof. Dr. Jürgen Albert and Prof. Dr. Dietmar Seipel for being available as examiners for my disputation, as well as to Prof. Dr. Hakan Kayal for taking the minutes.

For a little more than three years my research activities have been supported by the international doctorate program "Identification, Optimization and Control with Applications in Modern Technologies" within the Elite Network of Bavaria. Among other things this privilege made it possible for me to take part in international conferences and to stay in the United States as a visiting scholar for three month. I am very proud that I could have been a part of this program and I would like to thank the coordinator Prof. Dr. Günter Leugering and the remaining professors of the program for creating an inspiring atmosphere within the whole group. Further, I am obliged to Dr. Erik Kropat for his competent handling of all the administrative tasks. For giving me insights into other interesting topics besides mobile robots and for their companionship my appreciation also goes to the other PhD-students of the IDP, namely Nils Altmüller, Michael Blume, Izabella Ferenczi, Tobias Gradl, Falk Hante, Anna von Heusinger, Indra Kurniawan, Marcus von Lossow, Thorsten Spickenreuther, Alexander Thekale, Slobodan Veljovic, and Stefan Wendl. Special thanks go to Martin Saska with whom I shared the office and worked in close cooperation for more than three years.

At the chair of Computer Science VII: Robotics and Telematics I enjoyed very much to work in a team of extraordinary people. Therefore I would like to express my gratitude to all my former colleagues and, especially, to Daniel Eck, Marco Schmidt, Christian Herrmann, Markus Sauer, Florian Zeiger, Dr. Frauke Driewer, Dr. Lei Ma, Dr. Peter Hokayem, László Lemmer, Marcus Krauß, Florian Leutert, Robin Heß, Lothar Stolz, Maximilian Drentschew,

Zhihao Xu, Zhongyang Wu, Kaipeng Sun, Rajesh Shankar Priya, Lakshminarasimhan Srinivasan, Dieter Ziegler, Heidi Frankenberger, Edith Reuter, and Heidi Schaber. Besides the pleasant collaborations at work I also enjoyed a lot the private activities off the job. Further I want to thank my colleagues from the UIUC for having made me feel so welcome, especially Juan Mejía who supported me in academic as well as in everyday matters. For their companionship and friendship I also thank my C-U roommates Norman Simburger and Jia Yaodong.

The motivation and the support I received from my parents and from my sister encouraged me many times during the creation of this work, for which I owe them a large debt of gratitude. Moreover, I am deeply grateful for the support of my love Maike. She displayed a lot of patience while she took her time to listen to my scientific problems and to discuss about mathematical questions. Also, she regularly reminded me that there are other important things besides the scientific work as well. For this reason I want to dedicate this thesis to her.

Contents

1 Introduction

Since the first mobile robots have been designed during the late 60's, a lot of research and technological development has been undertaken in the field of intelligent vehicles. Today there exists a variety of different application areas in which mobile robots are successfully utilized. The biggest field is the transportation of materials in the industrial production sector, which is predominantly handled by self-contained mobile robots, often called autonomous transport systems. But also for space exploration mobile robots are successfully applied to investigate planetary surfaces, e.g., on the Mars (http://marsrovers.jpl.nasa.gov). Besides the industrial applications mobile robots also found their way into the private sector like, e.g., vacuum cleaning robots or lawn mowing robots, which accomplish their tasks mostly autonomously. As those single-robot systems proved their usefulness in various areas, one reasonably expects that for various tasks the utilization of multiple cooperating robots can significantly improve the overall efficiency. Naturally, the realization of the more complex multi-vehicle systems induces a variety of challenging problems that need to be considered and solved before a lucrative and robust application can be possible.

The investigation of distributed robotic systems started in the late 80's, when researchers began to focus on systems consisting of multiple mobile robots. One of the first works in this field describes the multi-robot architecture ACTRESS (ACTor-based Robot and Equipment Synthetic System), which consists of three robots and three workstations [AMI89, AOI^{+}91]. Within the testbed the authors evaluated algorithms for specific box-pushing tasks as well as collision avoidance based on dynamic path planning.

Since these early works, the number of publications concerning multi-vehicle systems is steadily growing. One reason for this is the development and the standardization of wireless communication technologies, which are significant for the mobility of distributed systems. Many fields of robotics research, like for example space exploration [DB08] or operations in dangerous environments [KKP^{+}04], can benefit from the application of groups of cooperating robots. Without risking human lives, the robots can access contaminated regions and build up a sensor network that collects important informations or they even might interact with their environment. Further they can be used to provide a mobile communication infrastructure in environments where usual communication systems (e.g., via satellites) are not available. In

[DBS05] a requirement analysis for the cooperation between humans and multiple robots that is based on questionnaires and personal interviews with professionals in the field of search and rescue, has been presented. Besides efficient approaches for the interaction between humans and robots there is a certain demand for autonomous functionalities, where robots accomplish tasks without the need for human intervention. In this monograph we focus on such autonomous functionalities by designing and verifying low-level control algorithms for the motion coordination in systems of multiple networked vehicles.

Depending on the application, a group composed of simple mobile robots often can solve tasks more efficient than it is possible for a single mobile robot even if it is well-equipped. Moreover a group of cooperating robots provides redundancy, which makes the system more robust against failures. A recent overview about directions and advances in multi-robot systems was given in [APP02]. Even though the class of mobile robots among others also includes aerial vehicles, underwater vehicles, spacecraft, as well as legged and chained ground vehicles, this monograph focuses mainly on wheeled ground vehicles and especially on robots of the car-like type. In contrast to many works in this field we attach great importance to the practicability of the developed methods. Thus, besides mathematical analyzes and simulation experiments, the main results of this thesis are also verified within laboratory hardware experiments with real mobile robots.

Today there exists a huge variety of different control architectures for multi-robot systems that can be classified into two groups. On the one hand we have the *centralized systems* in which a core unit is responsible for the overall coordination of the system. It manages environmental data received from the mobile nodes and uses this global information to generate new commands for them. In systems with many nodes a lot of information has to be processed and accordingly the computational resources of the central unit often have to be much higher compared to those of the other units. On the other hand there are the *decentralized systems* in which the resources are typically evenly distributed among the robots of the group. Each robot collects local information about its own state and the environment, which might be shared with other vehicles within a certain neighborhood often defined by the available communication range. As a result the usually global mission goal needs to be attained by robots that in principle rely on local information.

A general conclusion about which of the two control architectures is more efficient cannot be drawn since both structures have advantages and disadvantages. In centralized systems for example it is comparatively easy to optimize the performance of a task (e.g., total time) due to the availability of global information. Also failures of individual robots can be compensated without much difficulty. Nevertheless, a fault in the core unit could result in a complete

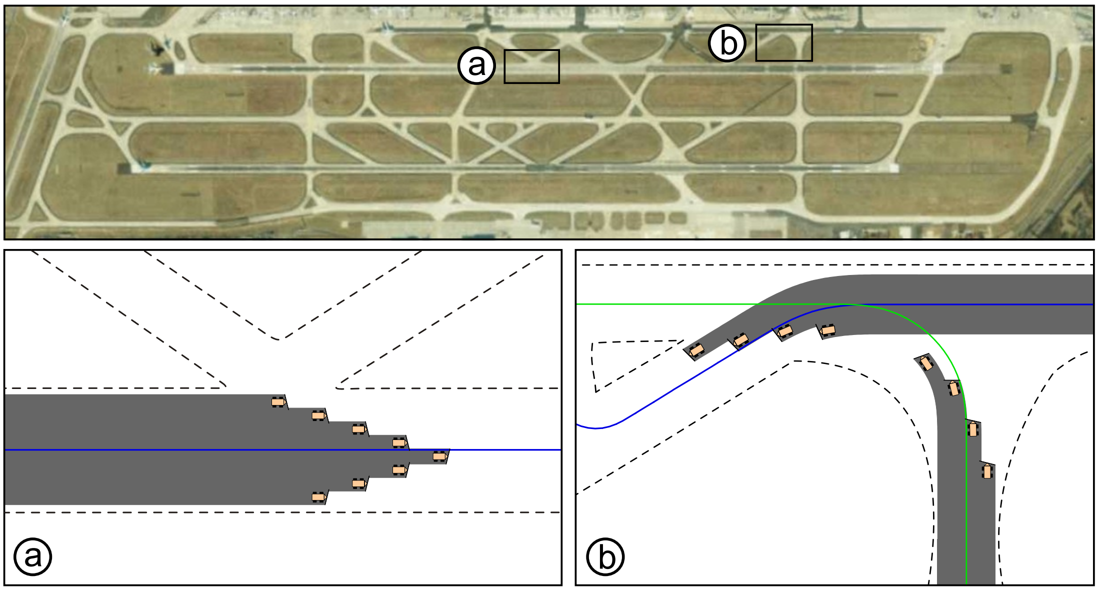


Figure 1.1: Motivating scenario 1: snow shoveling on airfields. Formations of autonomous snowplows remove the snow from the runways and service roads.

breakdown of the system. Due to the distributed nature, decentralized systems generally can cope with the breakdown of several nodes. Further, these systems are better scalable as the amount of information for the individual vehicle does not increase with a higher number of robots. However, it is often difficult to define distributed control laws that efficiently attain the global mission goal. Thus, the choice of the architecture mainly depends on the requirements of the underlying application. Note that in practice most systems are hybrid, i.e., they are neither purely centralized nor decentralized.

An important field when dealing with cooperative multi-vehicle systems is the coordination of motions among the robots. During a mission, various vehicles usually have to maintain a certain predefined relative configuration. To solve this cooperative control problem, the control of each individual vehicle needs to be implemented adequately, i.e., one has to design control laws that regulate the robots' actuators in order to fulfill the collective task. The two realistic application scenarios for cooperative groups of wheeled ground vehicles, described in the following, motivate the algorithms developed in the context of this work.

The first scenario is concerned with the problem of snow shoveling on the airfield of large airports. Today, the snow is typically removed from the runways and roads by utilizing a fleet of human-driven snowplows. Recent developments in the field of autonomous vehicles theoretically enable the application of autonomous snowplow robots for the sweeping task. A control center might coordinate the task by grouping the vehicles into formations and by providing individual plans about which roads will be cleaned next. Figure 1.1 shows

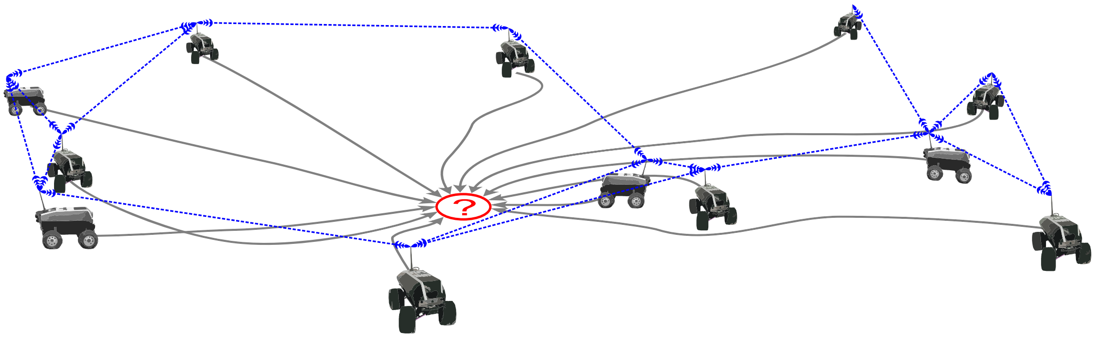


Figure 1.2: Motivating scenario 2: rendezvousing in networks of distributed mobile robots. The vehicles gather (rendezvous) by means of local control laws.

an aerial photo of a part of Frankfurt International airport, one of the biggest airports in Europe. It is not difficult to see that the creation of an optimal sweeping schedule is a complex task. The number of possibilities is immense since the snowplow formations are allowed to divide and unite arbitrarily during the shoveling process. Assuming that grouping information and a sequence of concurrent road segments is provided by the control center it remains to coordinate the motions among the robots. This includes trajectory planning for the coalitions as well as for the individual vehicles while enabling the possibility to divide and unite the formations. Further it is desirable that the tracking control for the motion execution guarantees optimal breadthwise coverage of the vehicles' shovels on the road. Especially during sharp turns, the center of the shovel significantly deviates from the snowplow's barycenter and therefore the application of available tracking controllers for car-like vehicles results in non-optimal coverage of the road.

In the second scenario a group of spatially distributed vehicles are supposed to gather within a certain area. Due to the spatial distance, a mobile robot is usually neither within the communication range nor within the sensing range of all of the other nodes as it is illustrated in Figure 1.2. Thus it makes sense to solve the task by means of decentralized algorithms that exclusively rely on the information from vehicles in range. Besides the robustness against vehicle failures, a decentralized approach allows the implementation within relatively large groups consisting of several hundreds of robots without the need for a significantly higher amount of individual computational power. The gathering task, which is often referred to as multi-robot rendezvous can be beneficial in a variety of applications. Consider for example a group of mobile robots that autonomously fan out in order to collect soil samples or other objects of interest. During the mission the collecting robots might need to come together with a container vehicle or with a robot that carries chemical analysis devices. Further it is desired to gather together the robots at the end of the mission in order to ease their pickup.

Other application scenarios that have been dealt with in the context of cooperating vehicles include but are not limited to traffic coordination [KGCC$^+$08], agricultural robots [ABLR$^+$06], object manipulation [LS05, LMPS05, Sch04], mutual localization [GS03, RB02], as well as exploration and mapping problems [BMSS05, MFP04]. An overview of applications focused especially on cooperative control of multi-vehicle systems can be found in a recent survey presented in [Mur07].

1.1 Contribution

This work focuses on coordination methods and the control of motion in groups of wheeled mobile robots, in particular of the car-like type. We shortly introduce notions of Lyapunov stability for dynamical systems with discontinuous right-hand side and reformulate existing theorems such that they allow stability analysis with respect to part of the state variables. After the introduction of various differential models for wheeled mobile robots we use these tools to analyze the stability of position stabilizing controllers for car-like robots. The main contribution of this work is threefold.

Formation motion planning

We introduce several enhancements to a motion planning approach for formations of nonholonomic mobile robots. First, we enable the approach to handle a time-varying reference speed without affecting the relative position coordinates. Second, with the enhanced version it is now possible to adjust the formation's shape during reference trajectory segments with continuously differentiable curvature in contrast to the need for a constant curvature in the original approach. Third, we introduce functions that allow adjustments of the formation's shape without generating undesired discontinuities in the speed and steering profiles of the vehicles. Finally, all of the enhancements are demonstrated in various simulations.

Applied motion coordination

The scenario of snow shoveling on an airfield by utilizing multiple formations of autonomous snowplows is discussed. We propose solutions to the subproblems of motion planning for the formations and tracking control for the individual vehicles. For the formations' leading vehicles, continuous reference trajectories are generated from a sequence of points. The corresponding paths are composed of straight, circular, and clothoid curve segments. Further, the developed approach can cope with changing formation shapes as well as with vehicles that leave or join a formation. Besides that, the derived tracking controller is designed in order to guarantee optimal breadthwise coverage of the roads to clean. While all situations that might occur have been tested in a simulation environment, we also verified the tracking controller in real robot hardware experiments.

Multi-robot rendezvous

The rendezvous problem for groups of car-like vehicles, which has not yet been addressed in literature, is analyzed. We propose distributed and provably stable control systems that drive all vehicles to a common position in the plane. Besides vehicles with free steering, for which we developed a stabilizing discontinuous control law, we also consider the more realistic case of car-like robots with bounded steering. In this case we combine a discontinuous control law with a specific switching logic that drives certain vehicles out of undesired equilibria via an orientation adjustment maneuver. The convergence properties of both approaches are analyzed by utilizing Lyapunov based techniques. Furthermore, they are evaluated within various simulation experiments, while the bounded steering case is also verified within laboratory hardware experiments. Finally we introduce a modification to the bounded steering system that allows concurrent orientation adjustment maneuvers among neighboring vehicles. Based on extensive simulation results we compare the performance of the modified system with the original one.

1.2 Outline

The remainder of this monograph is organized as follows. Chapter 2 lays the foundation for the stability analysis performed for mobile robot control systems throughout the following chapters. First we introduce Filippov's concept of solutions for dynamical systems with discontinuous right-hand side and the corresponding notion of Lyapunov based stability. After recapitulating the concept of stability with respect to a part of the state variables we reformulate the stated results in order to cope with partial stability of discontinuous dynamical systems.

Chapter 3 starts with an overview of several differential models for wheeled mobile robots. After this, we propose two discontinuous feedback controllers for the position stabilization of car-like robots, whose stability properties are analyzed by applying the methods presented in Chapter 2.

Chapter 4 deals with the planning of motion for formations of nonholonomic mobile robots. After a survey of current research in this area, we recapitulate in detail the concept of a given formation motion planning approach that maintains the formation in curvilinear coordinates and also incorporates predefined adjustments of the formation's shape. Then, we propose a variety of enhancements that eliminate different weaknesses in the presented planning method. These improvements result in the relaxation of certain assumptions made on the reference trajectory and further they guarantee continuous speed and curvature profiles for all vehicles even during changes of the formation's shape. Finally the proposed enhancements are demonstrated in various simulation experiments.

The problem of snow shoveling on airports by applying formations of autonomous snowplows is discussed in Chapter 5. We give an overview of related work before we describe the task specific concept of a centrally supervised system architecture with a hierarchical structure on the motion coordination level. Based on the idea of the motion planning approach considered in the previous chapter, reference trajectories are generated for every formation and thereupon also for each individual vehicle. Further, we introduce a dynamic feedback controller in order to track the generated trajectories with optimal breadthwise coverage of the shovel on the roads. After this, we demonstrate the motion related capabilities of the system in a simulated environment and present results from the verification of the controller in laboratory hardware experiments with two mobile robots.

Chapter 6 examines the rendezvous problem for groups of car-like mobile robots. First, we introduce a graph-based representation of the interaction topology in multi-agent systems followed by the formulation of the problem statement. After providing a detailed survey on multi-agent consensus and the rendezvous problem for networked multi-vehicle systems as a special case, we propose a local control law for the rendezvous of car-like robots with free steering and prove its convergence by applying the stability theorems stated in Chapter 2. Then, we focus on the rendezvous problem for groups of car-like vehicles with limited steering. Starting from a local control law as in the free steering case, we introduce orientation adjustment maneuvers that drive certain vehicles out of undesired equilibria. After proving the overall system's convergence we discuss the problem of terminating the rendezvous process in a distributed manner. While both proposed control systems are demonstrated in simulated scenarios, the bounded steering system is also verified in multiple hardware experiments. Finally, we propose an extension to the bounded steering system that enables concurrent orientation adjustment maneuvers among neighboring vehicles without damaging the system's convergence properties. The performance of the extended approach is statistically compared to that of the original one using the results of extensive simulation runs.

The main results presented throughout this thesis are summarized in Chapter 7. Based on this, we give concluding remarks and point out open issues and possible directions for future work.

2 Stability of nonlinear dynamical systems

In the literature there are several well established concepts of stability for dynamical systems. In this chapter we review the Lyapunov stability methods for discontinuous dynamical systems and the notion of partial stability for continuous systems. Then, by combining both concepts we formulate the corresponding theorems that are able to deal with partial stability in discontinuous systems.

2.1 Introduction

Consider the dynamical system given by

$$\dot{x}(t) = f\bigl(x(t)\bigr), \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $n \in \mathbb{N}\backslash\{0\}$ is a locally Lipschitz continuous vector field. Since the right-hand side of (2.1) exclusively depends on the state x, the system is referred to as *autonomous*. The point $x_e \in \mathbb{R}^n$ is an equilibrium of (2.1) if $0 = f(x_e)$. For convenience let us assume that the equilibrium is located at the origin of $\mathbb{R}^n$, i.e., $x_e = 0$. This can be done without loss of generality since any equilibrium can be shifted to the origin by a simple change of variables.

Definition 2.1.1. [Kha02] *The equilibrium point $x_e = 0$ of (2.1) is*

- ***(Lyapunov) stable*** *if, for each $\varepsilon > 0$, there is a $\delta = \delta(\varepsilon) > 0$ such that*

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \varepsilon, \; \forall t \geq 0.$$

- ***unstable*** *if it is not stable.*

- ***asymptotically stable*** *if it is stable and δ can be chosen such that*

$$||x(0)|| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0.$$

The stability of x_e can be analyzed with the aid of the well known stability theorem of Lyapunov also referred to as *Lyapunovs direct method* as well as with LaSalle's *invariance principle* (see, e.g., [Kha02]). The theorems can be utilized to prove Lyapunov stability and asymptotic stability of nonlinear dynamical systems, if one is able to find a suitable

Lyapunov function candidate. For system (2.1) a *classical solution* on an interval $[0, t_1]$ is a continuously differentiable map $x : [0, t_1] \to \mathbb{R}^n$ that satisfies (2.1). The existence of classical solutions is guaranteed by the continuity of the vector field. However, in case the vector field is discontinuous, classical solutions of (2.1) might not exist.

In the remainder of this chapter we present generalized versions of Lyapunov's stability theorem and LaSalle's invariance principle that are capable of dealing with an alternative notion of solutions especially developed for dynamical systems with discontinuous right-hand side (Section 2.2). Further we introduce definitions for stability of dynamical systems with respect to part of the state variables as well as a related Lyapunov-like stability theorem in Section 2.3. Finally in Section 2.4 we reformulate Lyapunov's stability theorem and LaSalle's invariance principle in order to be capable of handling partial stability of discontinuous dynamical systems.

2.2 Dynamical systems with discontinuous right-hand side

There exist different alternative ways to relax the classical notion of solutions in order to consider a discontinuous vector field (cf. [Cor08]). One such solution is the *Carathéodory solution* that follows the direction specified by the vector field except for a set of time instants that has measure zero. In other words, Carathéodory solutions are *absolutely continuous* functions that solve the integral version of (2.1) given by

$$x(t) = x(0) + \int_0^t f\big(x(s)\big)ds.$$

Relaxing the value of the vector field on a set of times of measure zero like in the notion of Carathéodory solutions is not always sufficient to guarantee that such solutions exist. Its value can be subject to significant changes arbitrarily close to a given point due to the discontinuity of the vector field, which might not allow the construction of a Carathéodory solution.

To get around this problem one can consider *Filippov solutions* [Fil88] that follow the idea of looking at a neighborhood of each point instead of focusing on the value of the vector field at individual points. To formalize this let $\mathcal{P}(\mathbb{R}^n)$ be the collection of subsets of $\mathbb{R}^n$ and let $B(x, \delta)$ be the open ball centered at x with radius $\delta > 0$. Now for $f : \mathbb{R}^n \to \mathbb{R}^n$ we define the *Filippov set-valued map* $K[f](x) : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$ by

$$K[f](x) = \bigcap_{\delta>0} \bigcap_{\mu(S)=0} \overline{\text{co}}\left\{f\big(B(x,\delta)\backslash S\big)\right\}, \tag{2.2}$$

where $\overline{\text{co}}$ denotes the convex closure and $\mu(S)$ denotes the *Lebesgue measure* of the set S. An alternative definition that is equivalent to (2.2) is given in [PS87]: There exists $S_f \subseteq \mathbb{R}^n, \mu(S_f) = 0$ such that for all $S \subseteq \mathbb{R}^n, \mu(S) = 0$

$$K[f](x) = \overline{\text{co}}\{\lim f(x_i) : x_i \to x, x_i \notin S_f \cup S\}.$$

Note that by the definition of the set-valued map, the value of $K[f]$ at a point x is independent of the value of the vector field f at x. To deal with the discontinuities of the vector field f we now replace the differential equation (2.1) by the differential inclusion

$$\dot{x}(t) \in K[f](x(t)). \tag{2.3}$$

A Filippov solution of (2.1) on $[0, t_1]$ is an absolutely continuous map $x : [0, t_1] \to \mathbb{R}^n$ that satisfies (2.3) for almost all $t \in [0, t_1]$. In [PS87] a calculus has been developed for computing Filippov's differential inclusion (2.3). We summarize some of the properties in the following proposition.

Proposition 2.2.1.

(i) If $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuous, then $K[f](x) = \{f(x)\}$.

(ii) If $f, g : \mathbb{R}^n \to \mathbb{R}^n$ are locally bounded, then $K[f + g](x) \subseteq K[f](x) + K[g](x)$. If f is also continuous, then $K[f + g](x) = f(x) + K[g](x)$.

(iii) If $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is continuous and $f : \mathbb{R}^n \to \mathbb{R}^m$ is locally bounded, then $K[gf](x) = g(x)K[f](x)$, where $gf(x) := g(x)f(x)$.

Before we can continue towards the statement of the stability theorem we need the definitions of the regularity of a function, the distance between two nonempty closed sets, semi-continuity of multi-valued functions, the generalized gradient, and the set-valued derivative which are stated in the following.

Definition 2.2.2. [Cla83] *A function $V : \mathbb{R}^n \to \mathbb{R}$ is called **regular at x** $\in \mathbb{R}^n$ if for all $v \in \mathbb{R}^n$ its right directional derivative $V'_+(x, v)$ at x in the direction of v exists, and $V'_+(x, v) = V^o(x, v)$. Here, the right directional derivative is defined as*

$$V'_+(x, v) = \lim_{h\to 0^+} \frac{V(x + hv) - V(x)}{h},$$

and $V^o(x, v)$ denotes the Clarke upper generalized derivative of V at x in the direction of v that is defined as

$$V^o(x, v) = \limsup_{y\to x, h\to 0^+} \frac{V(y + hv) - V(y)}{h}.$$

*The function V is called **regular** if it is regular at each $x \in \mathbb{R}^n$.*

It is easy to see that a function that is continuously differentiable at x is also regular at x.

Definition 2.2.3. [Fil88] *The* ***distance between two nonempty closed sets*** *A and B in a metric space, in particular in $\mathbb{R}^n$, is characterized by*

$$\beta(A,B) = \sup_{a\in A} \operatorname{dist}(a,B),$$
$$\beta(B,A) = \sup_{b\in B} \operatorname{dist}(b,A),$$
$$\alpha(A,B) = \max\{\beta(A,B),\beta(B,A)\},$$

where $\operatorname{dist}(x,Y)$ corresponds to the distance between point x and the closest point $y \in Y$ in the metric space.

Definition 2.2.4. [Fil88] *A set-valued function $K(p)$ is called* ***upper semi-continuous*** *(with respect to the inclusion) at the point p if $\beta(K(p'),K(p)) \to 0$ as $p' \to p$. Further, the function $K(p)$ is called upper semi-continuous if it is upper semi-continuous at each point of its domain.*

Definition 2.2.5. [Cla83] *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function, and let $\Omega_f \subset \mathbb{R}^n$ denote the set of points where V fails to be differentiable. The* ***Clarke generalized gradient*** *$\partial V : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$ of V is defined by*

$$\partial V(x) = \overline{\text{co}}\{\lim_{i\to\infty} \nabla V(x_i) : x_i \to x, x_i \notin \Omega_f\}.$$

It is easy to see that $\partial V(x) = \{\nabla V(x)\}$ on the whole domain if $V(x)$ is everywhere differentiable. With the definition of the Clarke generalized gradient, we can now define the set-valued derivative.

Definition 2.2.6. [BC99] *The* ***set-valued derivative of V with respect to the set-valued map K*** *is defined as*

$$\dot{\overline{V}}_{K[f]}(x) = \{a \in \mathbb{R} : \exists v \in K[f](x), p \in \partial V(x) \ \ \textit{such that} \ \ p \cdot v = a\}.$$

Note that if $V(x)$ is everywhere differentiable, the corresponding set-valued derivative can be written as

$$\dot{\overline{V}}_{K[f]}(x) = \nabla V(x) \cdot K[f](x).$$

The set-valued derivative allows to study how the function V evolves along the solutions of a differential inclusion without having to explicitly obtain the solutions. For this purpose the following result from [BC99] can be applied.

Lemma 2.2.7. *Let $x : [0,t_1] \to \mathbb{R}^n$ be a solution of the differential inclusion* (2.3)*, and let $V : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz and regular. Then $\frac{d}{dt}V\big(x(t)\big)$ exists almost everywhere and*

$$\frac{d}{dt}V\big(x(t)\big) \in \dot{\overline{V}}_{K[f]}\big(x(t)\big)$$

almost everywhere.

Proof. The proof can be found in [BC99, Lemma 1]. □

Now, with the aid of Lemma 2.2.7 we can formulate a generalization of Lyapunov's stability theorem for systems of the form (2.3). The following theorem originally appeared in [BC99].

Theorem 2.2.8. *Let $K : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)\backslash\{\emptyset\}$ be an upper semi-continuous multivalued map with compact, convex values. Let x_e be an equilibrium of the differential inclusion* (2.3)*, i.e., $0 \in K[f](x_e)$, and let $D \subseteq \mathbb{R}^n$ be an open and connected set with $x_e \in D$. Furthermore, let the function $V : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz and regular on D as well as $V(x_e) = 0$, and $V(x) > 0$ for $x \in D\backslash\{x_e\}$.*

If $\max \dot{\overline{V}}_{K[f]}(x) \leq 0$ *for each $x \in D$, then x_e is a stable equilibrium of* (2.3) *in Lyapunov's sense. If further* $\max \dot{\overline{V}}_{K[f]}(x) < 0$ *for each $x \in D\backslash\{x_e\}$, then x_e is an asymptotically stable equilibrium of* (2.3)*.*

This kind of stability is often referred to as *strong stability*, which means that stability holds for all solutions starting from each initial condition. *Weak stability* in contrast means that stability holds for at least one solution starting from each initial condition.

In many situations it is difficult or even impossible to find a Lyapunov function candidate whose derivative is strictly negative definite even if the corresponding system is asymptotically stable. If this is the case, it might be possible to apply a version of La Salle's invariance principle that was extended in order to deal with discontinuous dynamical systems. Similar to the continuous case this principle states the convergence of solutions towards a specific set under certain conditions and often allows to conclude asymptotic stability.

Theorem 2.2.9. *Let $K : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)\backslash\{\emptyset\}$ be an upper semi-continuous multivalued map with compact, convex values and let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous and regular function. Let $S \subset \mathbb{R}^n$ be compact and strongly invariant for* (2.3) *and assume that* $\max \dot{\overline{V}}_{K[f]}(y) \leq 0$ *for each $y \in S$. Then all solutions $x : [0, \infty) \to \mathbb{R}^n$ of* (2.3) *starting in S converge to the largest weakly invariant set M contained in*

$$S \cap \overline{\{y \in \mathbb{R}^n : 0 \in \dot{\overline{V}}_{K[f]}(y)\}},$$

where $\overline{\{\cdot\}}$ denotes the closure of the set.

Proof. The proof can be found in [BC99, Theorem 3]. □

Note that a dynamical system with discontinuous right-hand side can also be analyzed in the framework of *switching systems* [Lib03], where the switching is *state-dependent* and without *impulse effects*. For such systems the state space is partitioned into operation regions whose individual vector field can be described by a continuous-time dynamical model. With a continuously differentiable common Lyapunov function for all operation regions one can deduce stability and asymptotic stability of certain equilibria without considering solutions in the Filippov sense. Nevertheless, LaSalle-like invariance principles for switching systems like for example in [GST08, MAG06, HLAS05] usually depend on restrictions of the switching signal (e.g., persistent *dwell-time*) that in general do not hold for the control systems discussed in this monograph.

2.3 Partial stability of continuous systems

For some systems it makes sense to look at stability with respect to part of the state variables [Vor98]. This is, e.g., the case for a car-like robot in the plane that should be stabilized to a specific point $(x_d, y_d)^T$ while its final orientation is considered irrelevant (cf. Section 3.2).

The following definitions and the corollary on sufficient conditions for partial stability were taken from [CH02].

Definition 2.3.1. *Consider the nonlinear dynamical system*

$$\dot{x}_1(t) = f_1\big(x_1(t), x_2(t)\big), \quad x_1(0) = x_{10}, \tag{2.4}$$

$$\dot{x}_2(t) = f_2\big(x_1(t), x_2(t)\big), \quad x_2(0) = x_{20} \tag{2.5}$$

where $x_1 \in \mathcal{D} \subseteq \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$, $0 \in \mathcal{D}$, $n_1, n_2 \in \mathbb{N}\backslash\{0\}$, $f_1 : \mathcal{D} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ is such that for every $x_2 \in \mathbb{R}^{n_2}$, $f_1(0, x_2) = 0$ and $f_1(\cdot, x_2)$ is locally Lipschitz in x_1, $f_2 : \mathcal{D} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$ is such that for every $x_1 \in \mathcal{D}$, $f_2(x_1, \cdot)$ is locally Lipschitz in x_2. Additionally we assume that there exists a unique solution $\big(x_1(t), x_2(t)\big)$ to (2.4), (2.5) *that is defined for all $t > 0$.*

- *The nonlinear dynamical system* (2.4), (2.5) *is called* ***Lyapunov stable with respect to x_1*** *if, for every $\varepsilon > 0$ and $x_{20} \in \mathbb{R}^{n_2}$, there exists a $\delta(\varepsilon, x_{20}) > 0$ such that $||x_{10}|| < \delta$ implies that $||x_1(t)|| < \varepsilon$ for all $t \geq 0$.*
- *The nonlinear dynamical system* (2.4), (2.5) *is called* ***Lyapunov stable with respect to x_1 uniformly in x_{20}*** *if, for every $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$ such that $||x_{10}|| < \delta$ implies that $||x_1(t)|| < \varepsilon$ for all $t \geq 0$ and for all $x_{20} \in \mathbb{R}^{n_2}$.*
- *The nonlinear dynamical system* (2.4), (2.5) *is called* ***asymptotically stable with respect to x_1*** *if it is Lyapunov stable with respect to x_1, and for every $x_{20} \in \mathbb{R}^{n_2}$, there exists a $\delta(x_{20}) > 0$ such that $||x_{10}|| < \delta$ implies that* $\lim_{t\to\infty} x_1(t) = 0$.
- *The nonlinear dynamical system* (2.4), (2.5) *is called* ***asymptotically stable with respect to x_1 uniformly in x_{20}*** *if it is Lyapunov stable with respect to x_1 uniformly in x_{20}, and there exists a $\delta > 0$ such that $||x_{10}|| < \delta$ implies that* $\lim_{t\to\infty} x_1(t) = 0$ *for all $x_{20} \in \mathbb{R}^{n_2}$.*

Note that w.l.o.g. the definition considers $x_1 = 0$ to be the unique partial equilibrium of system (2.4), (2.5). Further note that there is a difference between the partial stability definitions given in Definition 2.3.1, where x_{20} can be arbitrary compared to the more popular definitions of partial stability given in [Vor98], which require that both initial conditions x_{10} and x_{20} lie in a neighborhood of the origin.

The following corollary gives sufficient conditions for partial stability as defined in Definition 2.3.1 that are similar to those from Lyapunov's direct method. Recall that a continuous function $\gamma : [0, a) \to [0, \infty]$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$.

Corollary 2.3.2. *Consider the nonlinear dynamical system given by* (2.4)*,* (2.5) *and let* $V'(x_1)$ *denote the Fréchet derivative of* V *at* x_1*. If there exists a continuously differentiable, positive-definite function* $V : \mathbb{R}^{n_1} \to \mathbb{R}$ *such that*

$$V'(x_1) f_1(x_1, x_2) \leq 0$$

for all $(x_1, x_2) \in D \times \mathbb{R}^{n_2}$*, then the nonlinear dynamical system is Lyapunov stable with respect to* x_1 *uniformly in* x_{20}*. If, in addition, there exists a class* $\mathcal{K}$ *function* $\gamma(\cdot)$ *such that*

$$V'(x_1) f_1(x_1, x_2) \leq -\gamma(||x_1||)$$

for all $(x_1, x_2) \in D \times \mathbb{R}^{n_2}$*, then the nonlinear dynamical system is asymptotically stable with respect to* x_1 *uniformly in* x_{20}*.*

Proof. The proof can be found in [CH02] and the corresponding erratum in [CH03]. □

Note that by setting $n_1 = n$ and $n_2 = 0$ we obtain the special case of nonlinear autonomous systems of the form (2.1). Then, Lyapunov/asymptotic stability with respect to x_1 and Lyapunov/asymptotic stability with respect to x_1 uniformly in x_{20} are equivalent to the classical Lyapunov/asymptotic stability of nonlinear autonomous systems. In case the partial derivatives of V are continuous the Fréchet derivative of V is equal to the gradient ∇V.

Further information on partial stability and control of dynamical systems can be found in [CH02, Vor98, Vor05].

2.4 Partial stability of discontinuous systems

In this section we incorporate the definitions of partial stability (Definition 2.3.1) into the stability theorem 2.2.8 and the invariance principle 2.2.9 for discontinuous dynamical systems. For this we consider the differential inclusions

$$\dot{x}_1(t) \in K_1\big(x_1(t), x_2(t)\big), \quad x_1(0) = x_{10}, \tag{2.6}$$

$$\dot{x}_2(t) \in K_2\big(x_1(t), x_2(t)\big), \quad x_2(0) = x_{20} \tag{2.7}$$

where $x_1 \in \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$, $n_1, n_2 \in \mathbb{N}\backslash\{0\}$, and $K_i : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathcal{P}(\mathbb{R}^{n_i})\backslash\{\emptyset\}$, $i \in \{1,2\}$, are upper semi-continuous multivalued maps with compact, convex values.

Definition 2.4.1. *Let $V : \mathbb{R}^{n_1} \to \mathbb{R}$ be locally Lipschitz and regular. We define the* ***set-valued derivative of V with respect to the set-valued map K_1*** *as*

$$\dot{\overline{V}}_{K_1}(x_1, x_2) = \{a \in \mathbb{R} : \exists v \in K_1(x_1, x_2), p \in \partial V(x_1) \ \text{ such that } \ p \cdot v = a\}.$$

The following result is a direct consequence of Lemma 2.2.7 since all properties of a solution $x(t)$ needed for the proof (cf. [BC99]) also hold for the semi-trajectory $x_1(t)$.

Lemma 2.4.2. *Let $x : [0, t_1] \to \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, $x(t) = \big(x_1(t), x_2(t)\big)$ be a solution of the dynamical system* (2.6),(2.7)*. Further, let $V : \mathbb{R}^{n_1} \to \mathbb{R}$ be locally Lipschitz and regular. Then $\frac{d}{dt}V\big(x_1(t)\big)$ exists almost everywhere and*

$$\frac{d}{dt}V\big(x_1(t)\big) \in \dot{\overline{V}}_{K_1}\big(x_1(t), x_2(t)\big)$$

almost everywhere.

Note that even though V is a function of $x_1(t)$, its time-derivative usually is a function of the full state $x(t)$. With respect to Definition 2.3.1 we can now use Lemma 2.4.2 to formulate the following stability result which generalizes the first part of Theorem 2.2.8.

Theorem 2.4.3. *Let $K_1 : \mathbb{R}^{n_1} \to \mathcal{P}(\mathbb{R}^{n_1})\backslash\{\emptyset\}$ be an upper semi-continuous multivalued map with compact, convex values. Let x_e be a partial equilibrium of the differential inclusion* (2.6)*, i.e., $0 \in K_1(x_e, x_2)$ for all $x_2 \in \mathbb{R}^{n_2}$, and let $D \subseteq \mathbb{R}^{n_1}$ be an open and connected set with $x_e \in D$. Furthermore, let $V : \mathbb{R}^{n_1} \to \mathbb{R}$ be locally Lipschitz and regular on D as well as $V(x_e) = 0$, and $V(x_1) > 0$ for $x_1 \in D\backslash\{x_e\}$. If $\max \dot{\overline{V}}_{K_1}(x_1, x_2) \leq 0$ for all $(x_1, x_2) \in D \times \mathbb{R}^{n_2}$, then the partial equilibrium x_e of system* (2.6),(2.7) *is Lyapunov stable with respect to x_1 uniformly in x_{20}.*

Finally we reformulate the invariance principle for discontinuous systems stated in Theorem 2.2.9 such that it can also deal with the semi-trajectories of a solution.

Theorem 2.4.4. *Let $K_1 : \mathbb{R}^{n_1} \to \mathcal{P}(\mathbb{R}^{n_1})\backslash\{\emptyset\}$ be an upper semi-continuous multivalued map with compact, convex values and let $V : \mathbb{R}^{n_1} \to \mathbb{R}$ be a locally Lipschitz continuous and regular function. Let $S \subset \mathbb{R}^{n_1}$ be compact and strongly invariant for* (2.6) *and assume that $\max \dot{\overline{V}}_{K_1}(y_1, y_2) \leq 0$ for all $(y_1, y_2) \in S \times \mathbb{R}^{n_2}$. Then the semi-trajectories $x_1(t)$ of all solutions $x(t) = \big(x_1(t), x_2(t)\big)$ of system* (2.6),(2.7) *with $x_{10} \in S$ converge to the largest weakly invariant set M contained in*

$$\overline{S \cap \{y_1 \in \mathbb{R}^{n_1} : 0 \in \dot{\overline{V}}_{K_1}(y_1, y_2)\}}.$$

Proof. The proof is along the lines of the proof of Theorem 3 in [BC99]. We just have to replace solutions of the system $x(t)$ by the corresponding semi-trajectory $x_1(t)$. □

3 Modeling and control of wheeled mobile robots

Today, basic control problems for a single wheeled mobile robot are considered as solved and the systems are quite well understood. As a consequence of this, a collection of differential models for wheeled mobile robots has been established over the years. The aim of this chapter is to review some of the well-known differential models and to point out the difficulties when dealing with car-like vehicles. After this we introduce and prove the stability of two discontinuous control laws for the car-like model that move the robot to or close to an arbitrary position in the plane.

3.1 Differential models for wheeled mobile robots

In this section we introduce differential models for the state transition of different types of wheeled mobile robots moving in the plane. Note that the dependence on the time t is often omitted, if it is clear from the context.

For the simplest model, the *single-integrator*, we consider the two state variables $x, y \in \mathbb{R}$ that denote absolute Cartesian coordinates in the plane and the corresponding linear equations of motion

$$\begin{aligned} \dot{x} &= u_1, \\ \dot{y} &= u_2, \end{aligned} \tag{3.1}$$

where the control input $u = (u_1, u_2)^T \in \mathbb{R}^2$ corresponds to the velocity vector of the system. The system is *fully actuated*, which means, it has as many independent control inputs as degrees of freedom. Systems like (3.1) with a maximum of one integration from input to state are called *kinematic* or *first order systems*. These kind of systems allow instant velocity changes, which can be considered as unrealistic, especially for heavy vehicles moving at high speed. To get around this problem, system (3.1) can be extended to second order dynamics. Physically that means the acceleration vector, which can be interpreted as a force, replaces the velocity vector as the control input. By considering Newton's law $F = ma$, the corresponding second order system often referred to as *double-integrator* in the plane yields

$$\begin{aligned} \dot{x} &= v_1, \\ \dot{y} &= v_2, \\ \dot{v}_1 &= u_1/m, \\ \dot{v}_2 &= u_2/m, \end{aligned} \tag{3.2}$$

where $v = (v_1, v_2)^T \in \mathbb{R}^2$ is the velocity vector, $u = (u_1, u_2)^T \in \mathbb{R}^2$ is the input force/acceleration vector and $m > 0$ corresponds to the mass of the vehicle. Models that incorporate Newton's law are called *dynamic models.* Note that the mass m is often incorporated in the control input and therefore it does not occur in the model. Moreover, it is natural for the proposed models to bound the input vector by some constant, such that $|u| \leq u_{\max}$. A mobile robot that exclusively drives on spherical wheels can be described by model (3.1) or (3.2). It is easy to see that such a robot can accelerate in an arbitrary direction of the planar state space.

This is not the case for all kinds of mobile robots and therefore one distinguishes between *holonomic* and *nonholonomic* vehicles. The classification is based on model-specific *kinematic constraints*, which mathematically describe restrictions of the robot's mobility. Holonomic constraints can be expressed as explicit function

$$F(q_1, \ldots, q_m, t) = 0,$$

that exclusively depends on position variables. In contrast, nonholonomic constraints cannot be expressed without the dependence on the velocity, i.e., the time-derivative of at least one position variable. Formally a nonholonomic constraint can be expressed as

$$G(q_1, \ldots, q_m, \dot{q}_1, \ldots, \dot{q}_m, t) = 0,$$

which is not integrable such that the resulting expression exclusively depends on position variables. A mobile robot is called nonholonomic if its model contains at least one nonholonomic constraint, otherwise the robot is holonomic.

Compared to the holonomic models described above, the *differential drive robot*, whose schematics are depicted in Figure 3.1, is a vehicle that is subject to a nonholonomic constraint. Its equations of motion can be written as

$$\begin{aligned} \dot{x} &= \frac{r}{2}(u_l + u_r)\cos\theta, \\ \dot{y} &= \frac{r}{2}(u_l + u_r)\sin\theta, \\ \dot{\theta} &= \frac{r}{L}(u_r - u_l), \end{aligned} \tag{3.3}$$

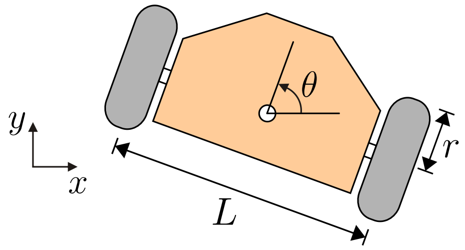


Figure 3.1: Schematics of a differential drive mobile robot. The vehicle's (x, y)-position is depicted by the white circle centered on the axle of the fixed wheels.

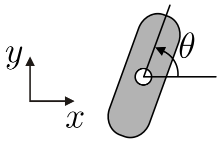


Figure 3.2: Schematics of a unicycle. The vehicle's (x, y)-position is depicted by the white circle in the middle of the wheel.

where $x, y \in \mathbb{R}$ are the Cartesian coordinates, $L > 0$ is the distance between the fixed wheels with radius $r > 0$, and $\theta \in [0, 2\pi)$ denotes the vehicle's orientation with respect to the x-axis. The input vector $u = (u_l, u_r)^T$ consists of the angular speeds of the left and the right wheel. Since there are less control inputs than state variables, (3.3) is referred to as *underactuated* system. This kind of robot is often equipped with one or more Castor wheels that stabilize the vehicle and prevent it from turning over. It is easy to see that when $u_l = u_r > 0$, the robot moves forward on a straight trajectory. Further, the robot rotates on the spot without any translation when an input with $u_l = -u_r \neq 0$ is applied. Due to its design, the differential drive robot cannot accelerate in arbitrary directions like it is the case for the introduced linear models. Especially it cannot move sideways against the direction of the fixed wheels, which is expressed in the vehicle's nonholonomic constraint

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0. \tag{3.4}$$

It is possible to perform a transformation on the action space of model (3.3). For this we set $u_\psi = (u_l + u_r)/2$ and $u_\omega = (u_r - u_l)$ and the model becomes

$$\begin{aligned} \dot{x} &= r u_\psi \cos\theta, \\ \dot{y} &= r u_\psi \sin\theta, \\ \dot{\theta} &= \frac{r}{L} u_\omega. \end{aligned} \tag{3.5}$$

In the transformed model, u_ψ can be interpreted as the robot's translational input and u_ω as its rotational input respectively.

Another model that is closely related to the differential drive model is the so called *unicycle* (cf. Figure 3.2), which is perhaps due to its simplicity the most frequently investigated nonholonomic model in publications concerning wheeled mobile robots. The equations of motion for the unicycle are

$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \omega, \end{aligned} \tag{3.6}$$

where v corresponds to the translational speed in the direction of the vehicle's front and ω controls its change of orientation. The unicycle simply consists of one wheel, which does not seem appropriate for practical terms. But note that under the transformation $v = ru_\psi, \omega = ru_\omega/L$, the models (3.5) and (3.6) are equivalent, which means a differential drive robot can imitate a unicycle and *vice versa*. Under the well defined, origin-preserving state and control coordinate transformation

$$\begin{aligned} x_1 &= x\cos\theta + y\sin\theta, \\ x_2 &= \theta, \\ x_3 &= 2(x\sin\theta - y\cos\theta) - \theta(x\cos\theta + y\sin\theta), \\ u_1 &= v - \omega(x\sin\theta - y\cos\theta), \\ u_2 &= \omega, \end{aligned}$$

model (3.6) takes the form

$$\begin{aligned} \dot{x}_1 &= u_1, \\ \dot{x}_2 &= u_2, \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1. \end{aligned} \tag{3.7}$$

This representation of model (3.6) is well known as *Brockett's nonholonomic integrator* and can be found quite frequently in the literature dealing with nonholonomic systems. Since it does not contain the trigonometric functions, it is usually more convenient to work with model (3.7) rather than with model (3.6).

Finally we introduce the kinematic model for the *rear-wheel driven car-like mobile robot* (cf. Figure 3.3), which will often be referred to throughout this thesis. It can be described with the following coupled nonlinear equations:

$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= vK = v\frac{\tan\phi}{L}. \end{aligned} \tag{3.8}$$

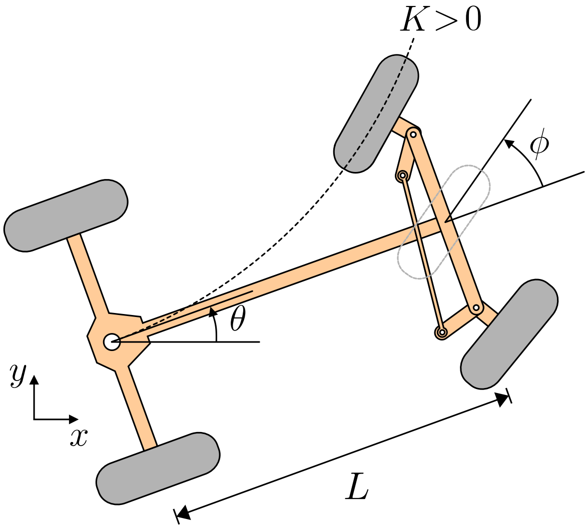


Figure 3.3: Schematics of a car-like robot. The vehicle's (x, y)-position is depicted by the white circle centered on its rear axle. The dashed curve denotes the robot's path, given a constant steering and a positive speed.

Like for the other introduced nonholonomic models the state vector consists of the robot's position coordinates $p = (x, y)^T \in \mathbb{R}^2$, and its heading $\theta \in [0, 2\pi)$. The control vector consists of speed $v \in \mathbb{R}$ and curvature $K \in \mathbb{R}$ or equivalently of speed and steering angle $\phi \in (-\pi/2, \pi/2)$, where ϕ corresponds to the angle of a virtual steering wheel centered on the robot's front axle and the constant L denotes the distance between front and rear axle.

Like the differential drive robot and the unicycle, the car-like model is subject to the nonholonomic constraint (3.4). Physically this can be explained by the fact that both vehicles are equipped with one or more fixed wheels on a single axle. As a result, the differential equations referring to the position variables x and y are equivalent. In the car-like model, the steering angle ϕ cannot take angles with an absolute value of $\pi/2$, since its tangent is not defined. This would correspond to a situation, where the steering wheels are oriented perpendicular to the orientation direction. Naturally the rear-wheel driven vehicle cannot move without skidding over its front wheels.

As it was mentioned before, the absolute value of the robot's speed and especially for car-like robots the absolute value of its curvature are often limited to an upper bound. In the context of model (3.8) this means

$$\begin{aligned} |v| &\leq v_{\max}, \\ |K| &\leq K_{\max}. \end{aligned} \tag{3.9}$$

where $v_{\max}$ and $K_{\max}$ are positive constants. From an upper bound for K automatically follows that $|\phi| \leq \phi_{\max}$ with $\phi_{\max} = \arctan(LK_{\max})$.

From the similarity of the car-like model (3.8) and the unicycle (3.6) we can conclude that the unicycle as well as the differential drive can imitate a car-like robot. The other way around is not possible, since for the differential drive and the unicycle, the rotation rate can be set independently of the translational speed. For the car-like robot the rotation rate also depends on the translational speed v, which makes it impossible to rotate the car without a translational motion. In other words, it cannot turn on the spot. Note that a car-like vehicle is still able to reach the complete (obstacle-free) configuration space, but possibly not without ever changing its movement direction.

Further information on differential models for wheeled mobile robots can be found, e.g., in [LaV06, Lau98, SN04].

3.2 Position stabilization of the car-like robot via discontinuous feedback

In the literature two major motion control problems for mobile ground robots situated in an obstacle-free plane are considered. On the one hand we have *trajectory following* which is also referred to as *trajectory tracking*. In this problem the robot has to follow a usually feasible reference trajectory defined in the Cartesian space. Such a controller will be introduced in Section 5.4 for the specific problem of snowplows tracking a certain reference trajectory with the center of their shovel. On the other hand there is the problem of *point-to-point motion* whose goal it is to reach a specific target configuration from a given initial one. The task is also often referred to as *point stabilization* or *position stabilization* if the orientation of the vehicle at the desired Cartesian position can be arbitrary.

Both problems can be solved by application of feedforward or feedback control or by a combination of both. As with real robot hardware uncertainties cannot be neglected, usually a certain level of feedback is desired in order to introduce robustness into the system. In general, novel control algorithms initially are designed and analyzed for the kinematic model. In a second step, they could be extended to a dynamic model that might be necessary when dynamic effects cannot be neglected. In this section we introduce feedback control laws that stabilize the position coordinates $p = (x, y)^T$ of a car-like robot with free as well as with bounded steering angle described by the kinematic equations in (3.8). W.l.o.g. we set the desired position to $(x_d, y_d)^T = (0, 0)^T$. So the problem is to design a control law that moves the car such that $(x, y)^T \to 0$ as $t \to \infty$.

As the proposed feedback control laws are discontinuous, we apply the stability theorems that have been introduced in Section 2.4 in order to analyze the convergence properties of the resulting closed-loop systems. While for the following systems stability can be also shown by applying the original direct method of Lyapunov to the individual continuous subspaces, the application of the discontinuous versions are meant as an exemplary introduction for the more complicated systems considered in Chapter 6.

3.2.1 Free steering angle

For the case of free steering angle, i.e., $\phi < \pi/2$, we propose the following discontinuous control law

$$v = -c_1 \operatorname{sgn}(x\cos\theta + y\sin\theta)\sqrt{x^2+y^2}, \tag{3.10}$$

$$\phi = \arctan\left(c_2\omega\frac{L}{v}\right) \tag{3.11}$$

with

$$\omega = -\frac{\pi}{2} + \left(\operatorname{atan2}(-y,-x) - \theta + \frac{\pi}{2}\right) \textbf{ modulo } \pi$$

and

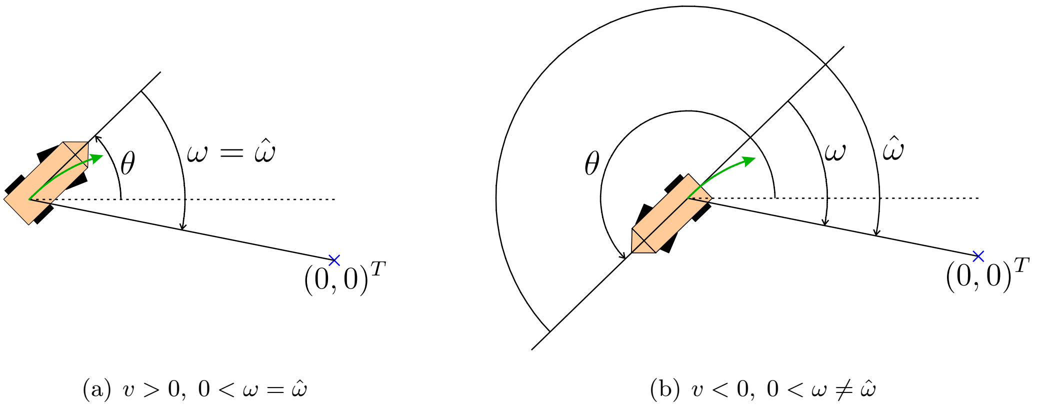
$$\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{otherwise} \end{cases}$$

where c_1 and c_2 are positive constants and the function $\operatorname{atan2}(\cdot,\cdot)$ denotes the arctangent of two variables which is defined as

$$\begin{aligned} & y \neq 0: \\ & \operatorname{atan2}(y,x) = \begin{cases} \arctan|y/x| \cdot \operatorname{sgn}(y) & x > 0 \\ \pi/2 \cdot \operatorname{sgn}(y) & x = 0 \\ (\pi - \arctan|y/x|) \cdot \operatorname{sgn}(y) & x < 0 \end{cases} \\ & y = 0: \\ & \operatorname{atan2}(y,x) = \begin{cases} 0 & x \geq 0 \\ \pi & x < 0 \end{cases} . \end{aligned} \tag{3.12}$$

For $v = 0$ the value of ϕ is not defined. But since in this case the system state of the robot does not change, we assume ϕ remains unchanged whenever $v = 0$. If we substitute (3.11) for ϕ in the kinematic model (3.8), we obtain $\dot{\theta} = c_2\omega$. Thus ω can be interpreted as the rate of turning. The sign of ω and of v both depend on the quadrant of the x-y-plane, in which the vehicle is currently located, as well as on its current orientation θ. Figure 3.4 shows two example configurations at a fixed position in the second quadrant resulting in different values for v and identical values for ω.

The following theorem states the stability of the closed-loop system defined by equations (3.8), (3.10) and (3.11) under the assumption that $v \neq 0$ at *unstable weak equilibria.* These equilibria refer to configurations with $0 \in K[f]\big((x,y,\theta)^T\big)$ to which the system does not



(a) $v > 0,\ 0 < \omega = \hat{\omega}$

(b) $v < 0,\ 0 < \omega \neq \hat{\omega}$

Figure 3.4: Illustration of the relation between the vehicle's configuration and the values for v and ω, where $\hat{\omega} := \operatorname{atan2}(-y, -x) - \theta$. The resulting movement direction is depicted by the arrow originating in the car's barycenter.

return if disturbed, where f corresponds to the function representing the vector field of the kinematic model (3.8). For the considered closed-loop system the set of unstable weak equilibria is defined by $D = \{(x, y, \theta)^T : x\cos\theta + y\sin\theta = 0\}$. Resulting from the design of the control law these equilibria do not have a stable manifold for certain choices of the controller constants c_1 and c_2 as we will show in the following. As a result, every solution can be only in an unstable weak equilibrium at its initial configuration. In those cases the removal of the zero from the Fillipov inclusion guarantees the convergence to the desired position.

Theorem 3.2.1. *Consider a car-like vehicle described by* (3.8)*. If*

$$c_1, c_2 \in \mathbb{R}^+,\ c_2 > \frac{2}{\pi}c_1 \tag{3.13}$$

holds and we assume that $v \neq 0$ at unstable weak equilibrium configurations, then under the control law defined by equations (3.10) *and* (3.11) *the partial equilibrium $(x, y) = (0, 0)$ is asymptotically stable uniformly in θ.*

Proof. The set-valued map corresponding to the car-like model can be written as

$$\dot{p} \in K_1(p, \theta) = K\begin{bmatrix} v\cos\theta \\ v\sin\theta \end{bmatrix},$$
$$\dot{\theta} \in K_2(p, \theta) = K[v\tan\phi/L].$$

Lyapunov stability of the partial equilibrium can be shown by applying Theorem 2.4.3. For this we define the quadratic candidate Lyapunov function

$$V(p) = \frac{1}{2}\left(x^2 + y^2\right).$$

Since V is continuous, its set-valued derivative can be computed as

$$\dot{\overline{V}}_{K_1}(p,\theta) = \nabla V(p) \cdot K_1(p,\theta) = \begin{pmatrix} x \\ y \end{pmatrix} \cdot K \begin{bmatrix} v\cos\theta \\ v\sin\theta \end{bmatrix}. \tag{3.14}$$

Substituting (3.10) for v in equation (3.14) and considering the calculus introduced in Proposition 2.2.1 yields $K[\operatorname{sgn}(z)]z = \{|z|\}$ as well as

$$\dot{\overline{V}}_{K_1}(p,\theta) = K[v](x\cos\theta + y\sin\theta) = \left\{-c_1\sqrt{x^2+y^2}|x\cos\theta + y\sin\theta|\right\} \leq 0.$$

As a result we have $V\big(p(t)\big) \leq V\big(p(0)\big)$ for all $t \geq 0$, i.e., the partial equilibrium $(x,y) = (0,0)$ of the closed-loop system is Lyapunov stable uniformly in θ.

Now we define a bounded set of positions as

$$S_k = \{p \in \mathbb{R}^2 : V(p) \leq k\}.$$

with $k > 0$. The set is closed due to the continuity of V and strongly invariant for the closed-loop system. Therefore we can apply the modified version of LaSalle's invariance principle introduced in Theorem 2.4.4 which states that the semi-trajectories $p(t)$ converge to the largest invariant set within

$$S = S_k \cap \overline{\{p \in \mathbb{R}^2 : 0 \in \dot{\overline{V}}_{K_1}(p,\theta)\}} = \{p \in S_k : x\cos\theta + y\sin\theta = 0,\ \theta \in [0,2\pi)\}.$$

Assuming that $V > 0$ we show now that the time-derivative of the set predicate is unequal zero, which renders all trajectories starting from points in S_k to be non-invariant except for the origin. The time-derivative of the set predicate can be calculated as

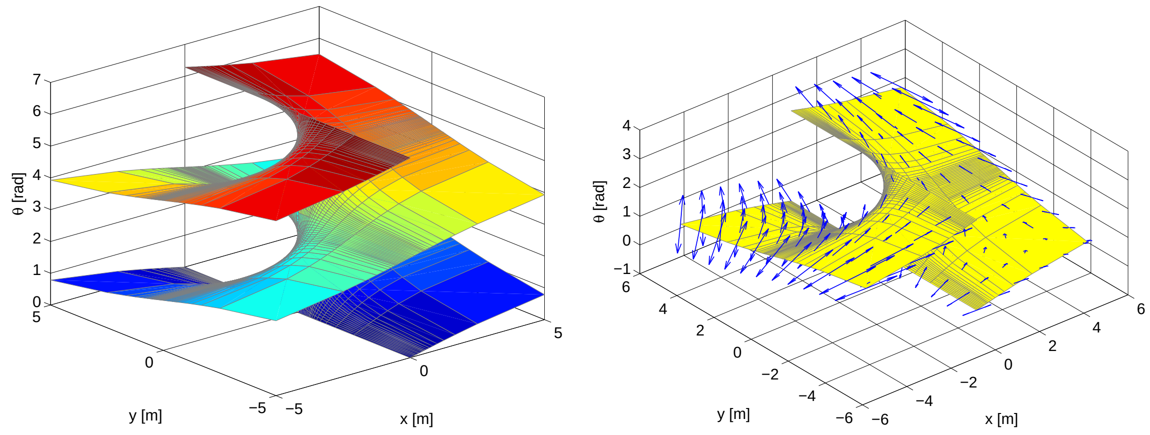
$$\frac{d}{dt}(x\cos\theta + y\sin\theta) = \cos\theta(\dot{x} + y\dot{\theta}) + \sin\theta(\dot{y} - x\dot{\theta}).$$

W.l.o.g. we can assume that $\theta = 0$, so we have $\omega = \pi/2 \operatorname{sgn} y$ from the definition of the control law. Remember that $v \neq 0$ at unstable weak equilibria by assumption and therefore we have $v \in [-c_1|y|, c_1|y|]\backslash\{0\}$ in terms of Filippov solutions. For the derivative we now obtain

$$\frac{d}{dt}(x\cos\theta + y\sin\theta) = \dot{x} + y\dot{\theta} = v + yc_2\omega = \underbrace{v}_{|\cdot|\leq c_1|y|} + \underbrace{\frac{\pi}{2}c_2|y|}_{|\cdot|>c_1|y|} \neq 0. \tag{3.15}$$

□

Note that since $V(p)$ is continuously differentiable it would have been sufficient to calculate $\dot{V}(p)$ directly. In Theorem 3.2.1 we assumed that $v \neq 0$ at unstable weak equilibrium configurations. This is because even though the control law (3.10) yields $v \neq 0$ for $V > 0$ the definition of Filippov solutions yields that at points of discontinuity, i.e., when $(x,y,\theta)^T \in D$, v might take any value in the interval $[-c_1\sqrt{x^2+y^2}, c_1\sqrt{x^2+y^2}]$. Since the interval includes



(a)

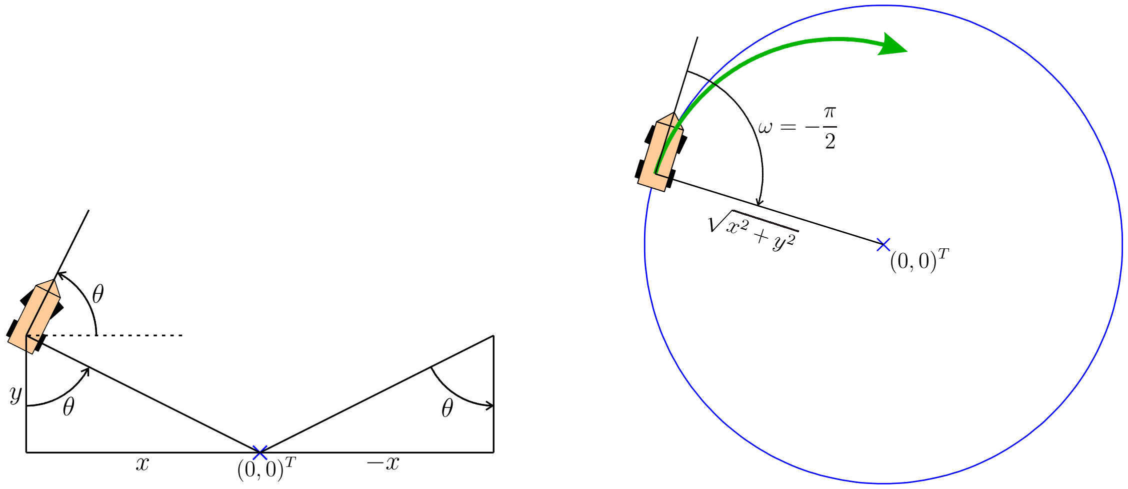
(b)

Figure 3.5: The set of discontinuous points D in a neighborhood of the origin (a) and the corresponding vector field on both sides of the surface (b).

zero, it is possible that the vehicle remains at this configuration. From the definition of (3.10) it is clear that any small perturbation results in a velocity unequal zero and the system drives away from the undesired, unstable weak equilibrium. Geometrically, the points in D are the configurations for which it does not matter if v is positive or negative in order to decrease the V-function. Compared to configurations outside of D for which the necessary movement direction is well-defined, the vehicle has to decide whether to move forward or backward. Figure 3.5(a) illustrates the set D in a neighborhood of the origin, while Figure 3.5(b) shows the vector field on both sides of the surface restricted to orientations in $[0, \pi]$. Since all opposing vectors are parallel, it is not possible to form a linear combination that is tangent to the discontinuous surface. Therefore we can conclude that sliding motions along the surface will not happen in this system.

With the modification of the set of Filippov solutions the set-valued map is neither compact nor convex anymore. Fortunately in our case the results in Section 2.4 still hold, since we just removed the solutions that remain in a point of D for more than an instant of time. As a result the largest weakly invariant set M in Theorem 2.4.4 becomes a subset of the one that corresponds to the standard Filippov solutions. In [YTN05] a related, even more restrictive modification of the Filippov set-valued map has been introduced. This modification eliminates trajectories that stay on the discontinuous surface as long as there exists at least one solution that leaves the surface.

In the simple case of stabilizing one vehicle towards a static point the reason for condition



(a)

(b)

Figure 3.6: Vehicle configuration from the set S. The car is oriented perpendicular to the line through position (x, y) and the origin (a). Thus, the vehicle has to drive into the disc to lower the value of the V-function (b).

(3.13) can be visually illustrated. For this we solve the set predicate of S for θ resulting in

$$(x, y, \theta) \in S \Leftrightarrow x = y = 0 \vee \theta = k\pi - \operatorname{atan2}(x, y), \qquad k \in \mathbb{Z}\backslash\{0\}$$

which means, if $(x, y)^T \neq (0, 0)^T$ the vehicle is oriented perpendicular to the line through position $(x, y)^T$ and the origin $(0, 0)^T$, i.e., $\omega \in \{-\pi/2, \pi/2\}$ (cf. Figure 3.6(a)). From (3.10) it follows that $v \neq 0$ and the steering angle needs to be such that the vehicle drives into the disk centered at the equilibrium with radius $\sqrt{x^2 + y^2}$ in order to get V decreased (cf. Figure 3.6(b)). Assuming $\omega = -\pi/2$ this requires ϕ to be such that

$$-\sqrt{x^2 + y^2} < r < 0 \tag{3.16}$$

where

$$r := \frac{1}{K} = \frac{L}{\tan\phi}, \qquad \phi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)\backslash\{0\}$$

is the momentary signed curve radius of the vehicle. Successively inserting ϕ, ω and v yields

$$\frac{L}{\tan\phi} = \frac{v}{c_2\omega} = -\frac{2v}{c_2\pi} = -\frac{2}{c_2\pi}c_1\sqrt{x^2 + y^2}.$$

It is easy to see that the constraint in (3.16) is satisfied whenever (3.13) holds. The case when $\omega = \pi/2$ can be shown in the same way.

In order to move the vehicle to an arbitrary position $(x_d, y_d)^T$ in the plane, one simply needs to shift the momentary coordinates appropriately resulting in the control law

$$\begin{aligned} v &= -c_1 \operatorname{sgn}\big((x - x_d)\cos\theta + (y - y_d)\sin\theta\big)\sqrt{(x - x_d)^2 + (y - y_d)^2}, \\ \phi &= \arctan\left(\frac{c_2 L}{v}\left(-\frac{\pi}{2} + \left(\operatorname{atan2}(y_d - y, x_d - x) - \theta + \frac{\pi}{2}\right) \textbf{ modulo } \pi\right)\right). \end{aligned}$$

The proposed controller was not designed to control the final orientation. For vehicles of the unicycle type, which have no limitations on the turning radius, full state stabilization can easily be achieved (see, e.g., [dWS92, ACBB95, MM97, ODV02]). The full state point stabilization for car-like vehicles sometimes also referred to as parallel parking problem is more complicated, since the vehicle cannot turn on the spot. Nevertheless there exist a couple of approaches that provide a reasonable solution (see, e.g., [Sam93, DOS98, LYC99]).

3.2.2 Bounded steering angle

Due to mechanical and physical limitations real car-like vehicles usually provide a smaller interval of applicable steering angles than the system considered in Section 3.2.1. That means there exists a constant $0 < \phi_{\max} < \pi/2$ that bounds the steering angle such that $\phi \in (-\phi_{\max}, \phi_{\max})$. As a result the vehicle cannot follow curves with arbitrary small curve radius. Actually the minimum traceable curve radius cannot get below $L/\tan\phi_{\max}$, which is equal to the reciprocal of the maximum curvature $K_{\max}$. It is clear that in contrast to cars with free steering angle, vehicles with bounded steering cannot reach every point in the plane without having to move away from it first. In order to avoid this problem we ease the task and consider an enhanced version of control law (3.10)-(3.11) that drives the car-like robot with limited steering angle into a bounded neighborhood of the desired destination, which for simplicity we again choose to be $(x_d, y_d)^T = (0, 0)^T$. For this we define

$$v = \begin{cases} -c_1\left(x\cos\theta + y\sin\theta + \operatorname{sgn}(x\cos\theta + y\sin\theta)\cdot b\right) & e > \frac{L}{\tan\phi_{\max}} \\ -c_1(x\cos\theta + y\sin\theta) & \text{otherwise} \end{cases}, \tag{3.17}$$

$$\phi_i = \sigma_{\phi_{\max}}\left(\arctan\left(c_2\omega\frac{L}{v}\right)\right), \tag{3.18}$$

where ω, c_1, c_2 are defined as in Section 3.2.1, and

$$e = \sqrt{x^2 + y^2}, \qquad b = e - \frac{L}{\tan\phi_{\max}}, \qquad \sigma_\alpha(x) = \begin{cases} \alpha & x > \alpha \\ -\alpha & x < -\alpha \\ x & \text{otherwise} \end{cases}.$$

Like in the free steering case we assume that ϕ remains unchanged whenever $v = 0$. The convergence properties of the bounded steering closed-loop system are stated in the following theorem.

Theorem 3.2.2. *Consider a car-like vehicle described by* (3.8) *that has a bounded steering angle* $\phi \in (-\phi_{\max}, \phi_{\max})$*, with* $0 < \phi_{\max} < \pi/2$*. If the constants in the control law are chosen such that*

$$c_1, c_2 \in \mathbb{R}^+, \qquad c_2 > \frac{2}{\pi} c_1$$

holds and we assume that $v \neq 0$ *at unstable weak equilibrium configurations, then the control law defined by equations* (3.17) *and* (3.18) *drives the vehicle towards a point* $(\tilde{x}, \tilde{y})^T$ *with* $\sqrt{\tilde{x}^2 + \tilde{y}^2} \leq L/\tan\phi_{\max}$*.*

Proof. Like in the proof of Theorem 3.2.1 we apply Theorem 2.4.3 to show stability in the sense of Lyapunov. Therefore we define the quadratic candidate Lyapunov function

$$V(p) = \frac{1}{2}(x^2 + y^2).$$

Since $V(p)$ is continuously differentiable we calculate its time-derivative directly instead of the set-valued derivative receiving

$$\dot{V}(p) = x\dot{x} + y\dot{y} = v(x\cos\theta + y\sin\theta).$$

Substituting the control law (3.17) for v and considering that $\mathrm{sgn}(z)z = |z|$ yields

$$\dot{V}(p) = \begin{cases} -c_1\big((x\cos\theta + y\sin\theta)^2 + |x\cos\theta + y\sin\theta| \cdot b\big) & e > \frac{L}{\tan\phi_{\max}} \\ -c_1(x\cos\theta + y\sin\theta)^2 & \text{otherwise} \end{cases}.$$

It is easy to see that $\dot{V}(p) \leq 0$ and therefore $V\big(p(t)\big) \leq V\big(p(0)\big)$ for all $t \geq 0$, so the partial equilibrium $(x, y) = (0, 0)$ of the closed-loop system is Lyapunov stable uniformly in θ.

We define now a bounded set of positions as

$$S_k = \{p \in \mathbb{R}^2 : V(p) \leq k\}.$$

with $k > 0$ and apply Theorem 2.4.4. So we define the subset of S_k for which $\dot{V} = 0$ as

$$S = \{p \in S_k : x\cos\theta + y\sin\theta = 0\}.$$

To complete the proof it has to be shown that the subset of S, for which the inequality $\sqrt{x^2 + y^2} > L/\tan\phi_{\max}$ holds, is non-invariant with respect to the closed-loop dynamics. Therefore we choose a $p \in S$ such that $\sqrt{x^2 + y^2} > L/\tan\phi_{\max}$ and assume w.l.o.g. that $\theta = 0$. This yields $x = 0$ and we have

$$\sqrt{x^2 + y^2} = |y| > \frac{L}{\tan\phi_{\max}}. \tag{3.19}$$

In the case when the steering angle bound is not reached we have $\dot{\theta} = c_2\omega$. Then, the time derivative of the set predicate of S is equal to (3.15) and therefore known to be nonzero.

In the second case we have to consider situations in which the steering angle bound is reached such that $\dot{\theta} = v \tan\phi_{\max} \operatorname{sgn}(-y)/L$. From the fact that $v \neq 0$ by assumption, the following statement holds for the set predicate of S:

$$\begin{aligned}\frac{d}{dt}(x\cos\theta + y\sin\theta) &= v + y\dot{\theta} = v + y\operatorname{sgn}(-y)\frac{\tan\phi_{\max}}{L}v \\ &= v\left(1 - \underbrace{|y|\frac{\tan\phi_{\max}}{L}}_{>1\ (3.19)}\right) \\ &\neq 0.\end{aligned}$$

□

Note that in most cases the vehicle will get closer to the goal than $L/\tan\phi_{\max}$. This depends on the angle at which the vehicle enters the circle with radius $L/\tan\phi_{\max}$ centered at the desired position. At this point of entrance the control law (3.17) switches and a movement speed different from zero is no more guaranteed. In literature there exist approaches that achieve point stabilization for bounded steering vehicles that definitely converge to the desired point by means of switching movement direction (see, e.g., [Sam93]) or by initially driving away from the goal (see, e.g., [Ind99]).

4 Motion planning for formations of nonholonomic mobile robots

The aim of this chapter is to analyze the problem of motion planning for formations of nonholonomic vehicles. In particular we want to find an approach that is not exclusively designed for differential drive vehicles but also applicable for mobile robots of the car-like type. After an overview of related work, a promising approach from literature is presented in more detail. Then we point out a number of weak points in the approach and improve them by proposing several enhancements. Finally the developed enhancements are demonstrated in various simulated scenarios.

4.1 Problem statement and related work

Motion planning deals with the problem to produce a feasible continuous motion that connects an initial configuration with a goal configuration while avoiding collisions with obstacles. A motion planning algorithm takes these configurations as input, and outputs appropriate control inputs (e.g., speed and steering commands) for the mobile robot. The research in robot motion planning can be traced back to the early stages of the development of computer-controlled robots, which started in the late 60's. Even though, most of the foundational achievements are more recent and have been discovered during the 80's. For instance, the *cell decomposition* method divides the configuration space into cells that lay inside, partially inside, or outside the free space. After this, one searches for a collision-free path by exploring the adjacency graph of free cells. Details on these classical motion planning approaches can be found in [Lat91].

From the 90's the motion planning problem has been considered in the presence of kinematic constraints. When the degrees of freedom of a mobile robot are not independent, like, e.g., for a car-like robot that cannot alter its orientation without changing its position, the searching for a motion is referred to as *nonholonomic motion planning.* In that case, not every path in the free configuration space is necessarily feasible. Another difficulty arises on the motion execution level, since the existence of stabilizing smooth feedback is no more guaranteed for nonholonomic systems. Detailed information about motion planning in the presence of kinematic constraints can be found, e.g., in [Lau98, LaV06].

The majority of motion planning approaches in multi-vehicle systems can be classified into two fields: *dynamic assignment* and *formation motion planning*. Given n identical agents and $m \geq n$ destinations, the task of dynamic assignment is to derive distributed control laws for every agent such that, for any initial configuration, all agents move to distinct destinations and remain there. In [ZP08], the authors propose a distributed hybrid approach to the assignment problem that simultaneously addresses the discrete allocation of destinations to agents as well as the continuous control strategies for driving the individual agents to the destinations. Optimization based strategies for large groups of integrator-agents that solve the problem for a given formation shape without considering location, scale, and rotational effects have been proposed in [DS07]. The problem of finding collision-free trajectories for multiple robots moving toward individual goals (static assignment) within a common roadmap environment has been discussed in [PCM08].

In contrast to the assignment problem, the task of formation motion planning (sometimes also referred to as *open-loop formation control*) is to generate a trajectory for each robot of a group that guarantees the maintenance of a predefined spatial pattern, the formation's shape. For the planning, a desired goal configuration or a reference trajectory is usually given beforehand. To achieve a coordinated motion, the generated trajectories can be tracked with a feedback controller by each robot individually. Thus in principle the task of formation control is split into the tasks of trajectory planning and tracking control. Solutions to the formation control problem can also be partitioned into three groups of approaches: *Virtual structures*, *behavioral* and *leader following*.

In the virtual structure approach the entire formation is treated as a single rigid body. In order to derive the control, firstly the dynamics of the virtual structure are defined. After this, the motion of the virtual structure is translated into the desired motion for each vehicle. Finally each vehicle individually tracks its corresponding desired trajectory [LT97]. In [EH01] the authors propose a coordination strategy for multi-agent formation control applicable to models that support independent control of rotational and translational velocity. The problem is defined by a rigid body constraint in combination with a desired reference path. Sufficient conditions for the existence of a control Lyapunov function for the robot formation have been derived in [ÖEH02]. The "Lyapunov Formation Function" is constructed as a weighted sum of the robots individual Lyapunov functions. [BK04] introduces an approach for the generation of optimal trajectories for rigid formations of fully actuated mobile robots towards a desired goal configuration. By tuning a parameter, the trajectory is chosen from a family that includes optimal trajectories for rigid formations as well as independent trajectories that are optimal for each individual robot. This allows the robots to cluster together in order to avoid obstacles. In [BTK04], a geometric method for motion planning for virtual structure formations of differential-drive wheeled mobile robots is presented that explicitly takes into account the vehicles' nonholonomic constraints.

The behavioral approach for formation control is inspired by the natural behavior of animal herds like flocks of birds or schools of fish. Each member of the group implements a set of prescribed desired behaviors like, e.g., move-to-goal, avoid-robot, avoid-static-obstacle, maintain-formation, etc. The control action is obtained as a weighted average of the control for each behavior [BA98]. In contrast to virtual structures, behavioral approaches are naturally decentralized as a result of their local design. With "social potential fields" the authors of [RW99] introduce a variant of the behavioral scheme, in which artificial force laws are defined between neighboring robots inducing a potential field of attraction and repulsion into the system. Also closely related to this method is the physics-based control scheme described in [SSHH04]. In this specific approach the mobile agents sense and react to virtual forces which are inspired by natural physics laws.

In the leader following approach, the formation consists of leading and following robots. Usually the control inputs of the followers are calculated with respect to position and orientation of one or multiple designated leaders [DOK01]. The leaders do not necessarily need to be physical vehicles as long as their configuration is not determined by means of sensor measurements, like it is the case in [DFK$^+$02] where the authors use the information from omnidirectional cameras to provide feedback to the applied controllers. Another vision based approach that uses a generic panned camera combined with a laser range finder on each robot has been presented in [FM02]. The key idea is that each robot keeps a single leader at a desired angle and distance. By panning the sensor and by moving themselves if necessary, the robots simply have to center their leader in the sensor's field of view. It is a fact that in leader-following systems of this type the formation error is amplified during the signal propagation, which has been formally justified in [TPK04]. A method for designing robust decentralized control laws for leader-follower type formations of unmanned aerial vehicles (unicycle model) with an overlapping information structure has been presented in [SİTT04]. After expanding the interconnected system into a higher dimensional space, a static state feedback controller is designed for the subsystems that afterwards can be contracted back to the original system for implementation. In [MSG$^+$08] a feedback law using Lyapunov-type analysis is derived that guarantees collision avoidance and tracking of a reference trajectory for a group of unicycles. Further, the authors present extensive experimental results that validate the effectiveness of the proposed approach.

The choice of which method to be utilized mainly depends on the underlying application. For large groups of vehicles the behavioral approach might be the best choice due to the scalability properties resulting from its decentralized nature. When more control over the individual vehicles is desired, usually the virtual structure and the leader following approaches are the better choice. Nevertheless, the distinction between these three groups of approaches is not always possible. For instance, [BLH01] addresses the problem of coordinating multiple spacecraft to fly in tightly controlled formations. For this the authors introduce a coordination architecture that merges leader-following, behavioral, and virtual-structure approaches. Besides space exploration, applications for formation control can be found, e.g., in cooperative box pushing [Sch04, LT97] as well as in cooperative load transportation [ASTBK06]. In both tasks the mobile robots have to fulfill certain geometric constraints that are similar to the problem of moving in a rigid formation. This is also the case in cooperative airfield snow shoveling as it will be discussed in Chapter 5 of this monograph. Formation control might also play a role in the design of automated highway systems as discussed, e.g., in [HV00].

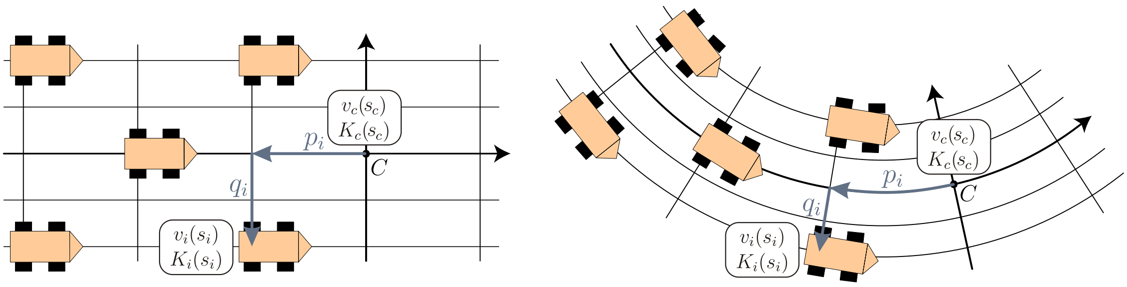
For some applications it is useful to separate the motion planning from the controller, giving more transparency and therefore an increased control over the overall group behavior. Such an approach, where the trajectory of every robot in the formation is explicitly planned, was introduced in [BC04]. Further, a controller is implemented on each robot to ensure the tracking of its planned trajectory in contrast to maintaining some fixed distance to a leader. In [GPFZ07] a decentralized strategy for car-like robots with bounded control inputs is presented, where each vehicle computes an optimal trajectory using only locally generated information. Moreover, the authors propose reactive adaption of the trajectory during the planning phases enabling obstacle avoidance and communication maintenance. [HSB$^+$07] discusses the specific motion planning problem for two-spacecraft interferometry based on geometric optimal control. Compared to single-telescope observation, astronomical interferometry with multiple spacecraft telescopes results in increased resolution capabilities due to an increased observatory baseline. A centralized approach for motion planning for multiple formations of disc-shaped robots in dynamic virtual environments has been proposed in [LG08]. The planning task is solved by combination of a flexible virtual structure approach and a model for crowd simulation.

4.2 Trajectory generation with piecewise constant reference curvature

In this section we describe the formation motion planning approach by Barfoot and Clark [BC04, BCRD02] in more detail. Depending on a reference trajectory, the method generates trajectories for the nonholonomic mobile robots of a formation, whose kinematics were introduced in (3.8). Further it allows to specifically alter the formation's shape during the motion, which is accomplished by ensuring that the relative distances between robots are maintained in curvilinear coordinates rather than in the usual Cartesian coordinate system. The curvature of the curvilinear coordinate system corresponds to the momentary curvature of the reference trajectory. As a consequence, during turns the method automatically plans for robots on the outside to speed up and for robots on the inside to slow down.

It is assumed that each robot can sense its own location relative to a common global reference frame, thus inter-robot communication is not necessary. However, communication is still necessary in order to transfer information about the planned motion (reference trajectory) to each of the robots. In the following we refer to the leader as the reference point, which might be either a real or a virtual vehicle moving on the reference trajectory. The task is now to calculate suitable control inputs for the following vehicles in order to maintain a certain formation structure, which can be subject to intended modifications. Due to the nonholonomic equations of motion it is often inappropriate to compute these control inputs from the momentary position and orientation of a reference point. In the considered method this problem is omitted by maintaining the formation in curvilinear coordinates and calculating speed and curvature from control inputs of the reference point at identical traveled distances. It can be seen in Figure 4.1 that the distances between robots vary for different curvatures of the trajectory in the Cartesian but not in the curvilinear coordinate system. For this reason we assume that the initial distances between robots are chosen big enough in order to avoid inter-vehicle collisions within the group during the coordinated motion.

Let the formation consist of N nonholonomic vehicles that evolve according to the equations introduced in (3.8). To demonstrate that the followers' control inputs do not explicitly depend on the time we perform a change in variable from time t to the traveled distance s. Hence the control inputs of the reference point are defined as $v_c(s_c)$ and $K_c(s_c)$ at a traveled distance $s_c(t)$ (cf. Figure 4.1). In the same manner $v_i(s_i)$ denotes the speed and $K_i(s_i)$ denotes the curvature of the ith robot's trajectory at a traveled distance $s_i(t)$ respectively. Note that $s_i(t)$ is meant to be measured along the reference trajectory. Furthermore $p_i(s_i)$ and $q_i(s_i)$ are the curvilinear coordinates of robot i relative to the reference point C. Therefore, the



(a) $K_c = 0 \Rightarrow K_i = 0, v_i = v_c$

(b) $K_c > 0 \Rightarrow K_i < K_c, v_i > v_c$

Figure 4.1: 5 Robots driving in formation on a straight (a) and on a curved (b) trajectory. v_i denotes the speed and K_i denotes the curvature of the ith robot's trajectory at the current traveled distance s_i. p_i and q_i are the curvilinear coordinates of the ith robot measured relatively to the reference point C whose momentary control inputs are denoted by v_c and K_c.

relationship between the reference point's traveled distance s_c and the ith vehicle's traveled distance s_i can be written as

$$s_i(t) = s_c(t) + p_i(t). \tag{4.1}$$

As long as the reference point is located in front of the vehicles, i.e., p_i is opposite in sign to v_c, the reference trajectory can be generated in real-time, e.g., by a high level path planner or a teleoperator. This results from the fact that the shape of the formation is completely defined by the coordinates (p_i, q_i) and therefore it does not depend on the reference trajectory.

The derivation of the control inputs for the followers is based on the relationship between the Cartesian coordinates (x_i, y_i) and the curvilinear coordinates (p_i, q_i) (cf. Figure 4.2). Also it is assumed that the curvature K_c of the reference trajectory is piecewise constant. From Figure 4.2 it can be seen that

$$\begin{aligned} x_i(t) &= \left(\frac{1}{K_c} - q_i(t)\right) \sin\big(s_i(t)K_c\big), \\ y_i(t) &= \frac{1}{K_c} - \left(\frac{1}{K_c} - q_i(t)\right) \cos\big(s_i(t)K_c\big). \end{aligned} \tag{4.2}$$

Note that due to readability the dependency on time t is omitted in the following equations. Calculating the first time derivatives of (4.2) we obtain

$$\begin{aligned} \dot{x}_i &= -\dot{q}_i \sin(s_iK_c) + \dot{s}_i(1 - q_iK_c)\cos(s_iK_c), \\ \dot{y}_i &= \dot{q}_i \cos(s_iK_c) + \dot{s}_i(1 - q_iK_c)\sin(s_iK_c), \end{aligned} \tag{4.3}$$

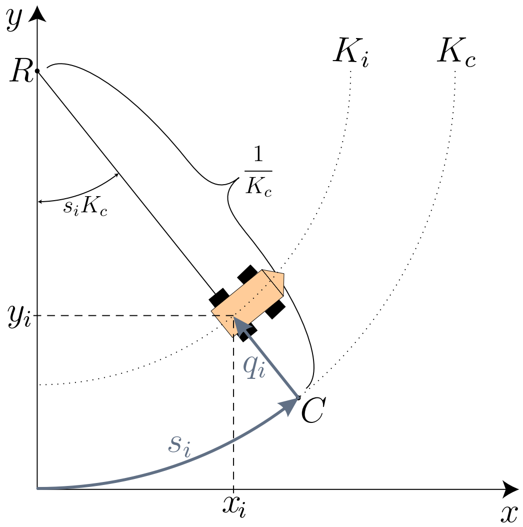


Figure 4.2: Relationship between Cartesian coordinates (x_i, y_i) and curvilinear coordinates (p_i, q_i). The point R denotes the center of curvature for vehicle i as well as for the reference C.

and for the second time derivatives we obtain

$$\begin{aligned}\ddot{x}_i &= -\big(\ddot{q}_i + \dot{s}_i^2 K_c(1-q_iK_c)\big)\sin(s_iK_c) + \big(\ddot{s}_i(1-q_iK_c) - 2\dot{s}_i\dot{q}_iK_c\big)\cos(s_iK_c),\\ \ddot{y}_i &= \big(\ddot{q}_i + \dot{s}_i^2 K_c(1-q_iK_c)\big)\cos(s_iK_c) + \big(\ddot{s}_i(1-q_iK_c) - 2\dot{s}_i\dot{q}_iK_c\big)\sin(s_iK_c).\end{aligned} \tag{4.4}$$

It is well known from basic physics that the time derivative of the position results in the speed and therefore we can write the overall speed of vehicle i along a path in the plane as

$$v_i = \sqrt{(\dot{x}_i^2 + \dot{y}_i^2)}. \tag{4.5}$$

With equation (4.5) and the component-wise velocities given by (4.3) we can express the speed of vehicle i as

$$v_i = \sqrt{\big(\dot{q}_i^2 + \dot{s}_i^2(1-q_iK_c)^2\big)}. \tag{4.6}$$

Considering that $dq_i/ds_i = \dot{q}_i/\dot{s}_i$ and $\dot{s}_i = v_c$, equation (4.6) can be reformulated as

$$v_i = \mathcal{Q}_i v_c, \tag{4.7}$$

where

$$\mathcal{Q}_i := \sqrt{\left(\frac{dq_i}{ds_i}\right)^2 + \Big(1 - q_iK_c\Big)^2}. \tag{4.8}$$

Note that equation (4.7) depends on local variables, i.e., on variables that concern vehicle i as well as on v_c and K_c which characterize the reference trajectory.

The curvature at a point (x, y) on a curve in parameter notation can be calculated as

$$K = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\left(\dot{x}^2 + \dot{y}^2\right)^{3/2}}. \tag{4.9}$$

In order to receive the curvature for robot i, we simply substitute (4.3) and (4.4) for the first and second derivatives in expression (4.9) yielding

$$K_i = \frac{K_c \dot{s}_i \dot{q}_i^2 + (1 - q_i K_c)(\dot{s}_i \ddot{q}_i - \ddot{s}_i \dot{q}_i) + K_c \dot{s}_i \big(\dot{q}_i^2 + \dot{s}_i^2 (1 - q_i K_c)^2\big)}{\sqrt{\big(\dot{q}_i^2 + \dot{s}_i^2 (1 - q_i K_c)^2\big)^3}}. \tag{4.10}$$

Reducing the fraction by $\dot{s}_i^3$ and considering that

$$\frac{d^2 q_i}{d s_i^2} = \frac{\dot{s}_i \ddot{q}_i - \ddot{s}_i \dot{q}_i}{\dot{s}_i^3}, \tag{4.11}$$

equation (4.10) can be written as

$$K_i = \frac{1}{\mathcal{Q}_i} \left(K_c + \frac{\Big(1 - q_i K_c\Big) \frac{d^2 q_i}{d s_i^2} + K_c \left(\frac{d q_i}{d s_i}\right)^2}{\mathcal{Q}_i^2} \right). \tag{4.12}$$

Like equation (4.7) this expression depends on local variables as well as on the curvature of the reference trajectory K_c.

Until now, the case when the center of curvature is located between the robot itself and the reference trajectory has been neglected. As can be seen in Figure 4.3 this happens when q_i is greater/less than the momentary positive/negative signed curve radius of the reference trajectory given by $1/K_c$ or equivalently whenever

$$1 - q_i K_c < 0.$$

It can be seen that in such a case the values for speed v_i and curvature K_i need to be opposite in sign to v_c and K_c in order to maintain the formation. This is achieved, e.g., by multiplying equations (4.7) and (4.12) with the factor

$$\mathcal{S}_i = \operatorname{sign}(1 - q_i K_c). \tag{4.13}$$

As such we finally receive

$$v_i = \mathcal{S}_i \mathcal{Q}_i v_c, \tag{4.14}$$

$$K_i = \frac{\mathcal{S}_i}{\mathcal{Q}_i} \left(K_c + \frac{K_c \left(\frac{d q_i}{d s_i}\right)^2 + \Big(1 - q_i K_c\Big) \frac{d^2 q_i}{d s_i^2}}{\mathcal{Q}_i^2} \right). \tag{4.15}$$

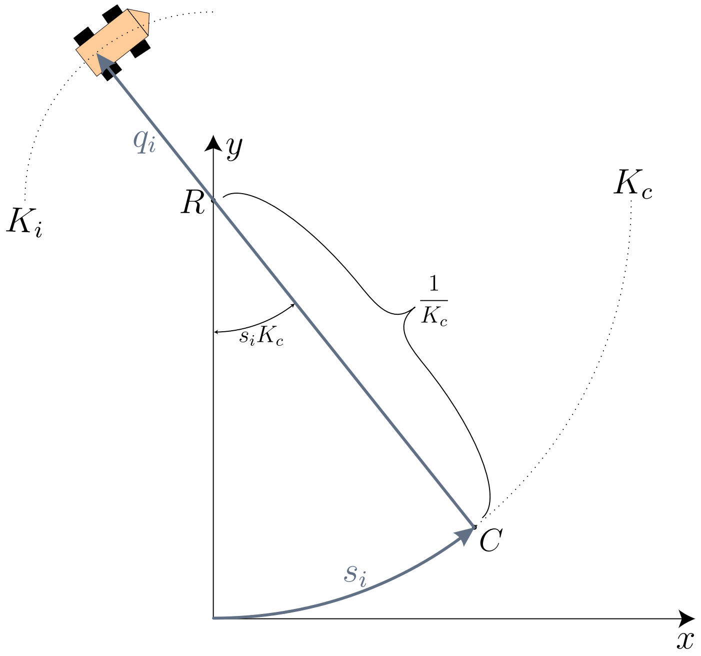


Figure 4.3: Illustration of the special case when q_i is greater than the momentary positive signed curve radius of the reference trajectory given by $1/K_c$.

The coordinate p_i and, during trajectory segments with constant curvature ($K_c(s_i) = K_c =$ const.), also the coordinate q_i might be continuously altered in order to effect modifications of the formation's shape. Among other things this can be especially helpful for reactive obstacle avoidance as it is described, e.g., in [HSS06, SHS07]). A change of p_i is simply accomplished by a temporary decrease or increase of the ith robot's driving speed. To vary the q-coordinate of robot i the following function has been proposed:

$$\psi_{s_o,s_f,\psi_o,\psi_f}(s) = \begin{cases} \psi_o & 0 \leq s \leq s_o \\ \psi_o + (\psi_f - \psi_o)b^2(3-2b) & s_o < s \leq s_f \\ \psi_f & s_f < s \end{cases} \tag{4.16}$$

where $b = (s - s_o)/(s_f - s_o)$. It can be seen that ψ_o and ψ_f denote the initial and the final value of function (4.16), which is meant to be adjusted during the interval $[s_o, s_f]$ of traveled distance measured along the reference trajectory. To apply this function for a change of q_i while the ith robot's traveled distance resides between $s_{i,o}$ and $s_{i,s}$ one simply sets

$$q_i(s_i) = \psi_{s_{i,o},s_{i,f},q_{i,o},q_{i,f}}(s_i). \tag{4.17}$$

Figure 4.4 shows an example of the function and its first two derivatives. A benefit of this approach lies in the fact that the adjustments of p_i and q_i do not depend on the shape of the reference trajectory. Therefore one might offline create a catalog that contains a collection of

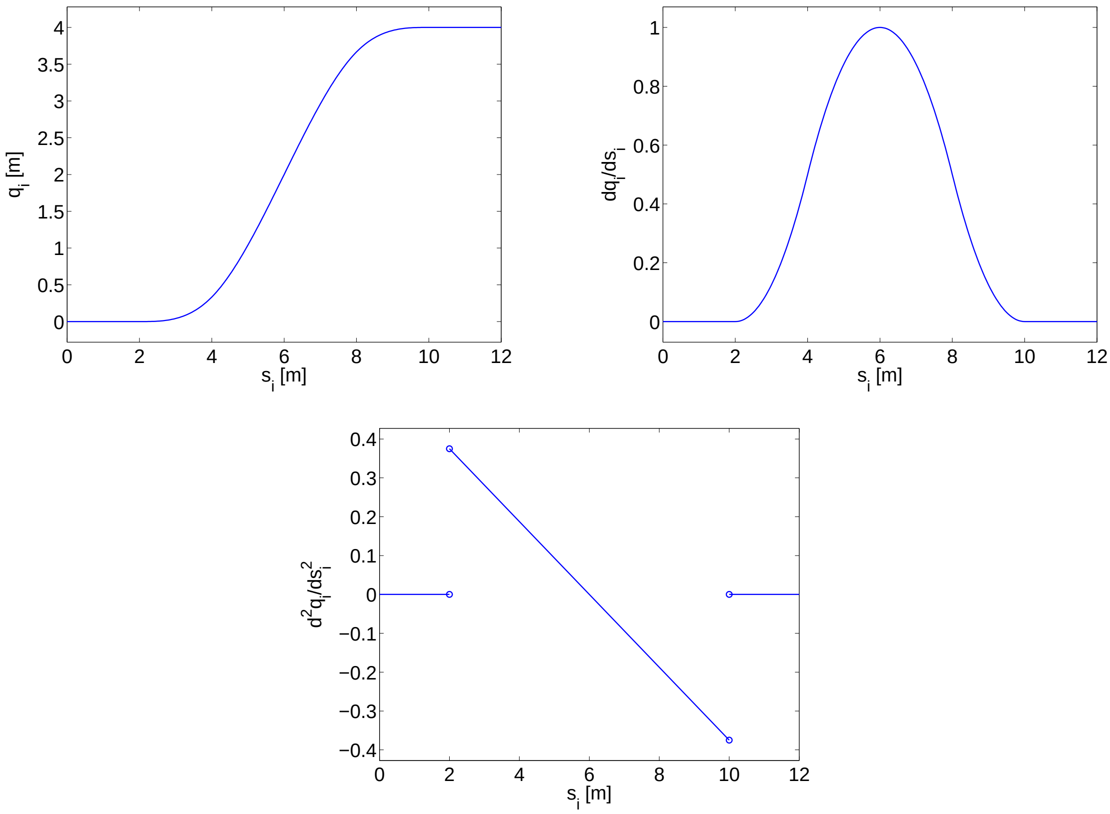


Figure 4.4: Example of function (4.17) and its first two derivatives. The constants were chosen as $s_{i,o} = 2\text{m}$, $s_{i,f} = 10\text{m}$, $q_{i,o} = 0\text{m}$ and $q_{i,f} = 4\text{m}$.

different maneuvers for the present formation. Thus, a higher level entity that might rely on sensor measurements can trigger adequate maneuvers from the catalog in specific situations. This could be for example to switch to a column formation in order to overcome a narrow passage that was just detected. As a result the formations arrangement is changed with minimal computational effort.

If the coordinates (p_i, q_i) remain unchanged in order to maintain the structure of the formation, the control inputs of the ith follower (4.14)-(4.15) at a traveled distance of $s_i(t)$ simplify to

$$v_i(s_i) = v_c(s_i)\big(1 - q_i K_c(s_i)\big),$$
$$K_i(s_i) = \frac{K_c(s_i)}{1 - q_i K_c(s_i)}.$$

Since the formation's structure remains constant there is no need for keeping K_c constant and we can specify the dependency on the traveled distance s_i. Note that this is not the case for equations (4.14) and (4.15), where we omitted to specify a dependency on s_i as in general it does not hold. This problem will be taken on in Section 4.3 where an extended equation for K_i will be derived.

As the geometrical distance between robots with a different value for p_i varies slightly during turns, the approach can be classified as flexible rather than rigid, which makes it inappropriate for cooperative load transport. But since vehicles with the same value for p_i maintain a fixed distance to each other, the approach is well suited for convoying and for the cooperative box pushing task. The possibility to alter the formation's shape during the movement allows reactive response to unforeseen obstacles and narrow passages.

Further information on this method about motion planning for robot formations can be found in [BC04, BCRD02].

4.3 Enhanced trajectory generation

Before we point out weaknesses and propose different enhancements to the motion planning approach described in Section 4.2, we recapitulate the definitions of polynomial splines and of the differentiability classes of a function.

Definition 4.3.1. *Consider a real valued function $f : \mathbb{R} \to \mathbb{R}$ and let $k \in \mathbb{N}^+$. The function f is said to be of **class $\mathbf{C^k}$** or **$\mathbf{C^k}$-continuous**, if the derivatives $f', f'', \ldots, f^{(k)}$ exist and are continuous. If $k = \infty$, the function f is referred to as **smooth**.*

Definition 4.3.2. [HV94] *Let*

$$\Delta_N = \{a = \tau_0 < \tau_1 < \ldots < \tau_N = b\}, \quad N \geq 2$$

*be a fixed subdivision of the interval $[a, b]$. Then, a **polynomial spline** of degree $r \in \mathbb{N}^+$ and defect $k \in \{1, \ldots, r\}$ with respect to Δ_N is a function s composed of algebraic polynomials of degree r 'tied together' at the points $\tau_1, \ldots, \tau_{N-1}$ in such a way that on the whole interval $[a, b]$ the spline s is continuous together with its derivatives up to and including order $r - k$. Thus $s \in \mathrm{C}^{r-k}$ on $[a, b]$ and on each interval (τ_{i-1}, τ_i), $i = 1, \ldots, N$ s is an algebraic polynomial of degree at most r.*

The approach for generating trajectories for formations as described in Section 4.2 is unsuitable for practical usage due to several reasons. The first problem results from the algorithm's behavior for non-constant reference speed. If a non-constant function is applied for v_c the values of p_i change with time causing unintentional deformations in the formation's arrangement. The second problem comes from the fact that K_c is assumed to be piecewise constant. Thus the equations that define the control inputs for the followers are not able to deal with changes of q_i during reference trajectory segments with non-constant curvature. That means, modifications of the formation's shape are very limited and therefore cannot be applied reactively, e.g., to avoid recently detected obstacles. Moreover, if the reference trajectory is generated in real-time, the complete segment for a maneuver might not be known in time by the corresponding robot. A third reason for impracticalness is that if one utilizes the C^1-continuous function (4.17) to effect a change of the q-coordinate, this results in a discontinuous profile for the vehicle's curvature K_i. Looking at equation (4.17) it can be seen that its second derivative is not defined in the connection points (cf. Figure 4.4). In terms of practical application this requires an instantaneous change of the steering angle during the motion, which is physically impossible. In the remainder of this section we describe how to modify the approach that was described in Section 4.2 in order to overcome the just mentioned problems. We assume from now on that the curvature of the reference trajectory $K_c(s_i)$ is of class C^1 or higher since we need the first derivative of $K_c(s_i)$ in order to compute the followers' curvature input K_i as will be seen later. Note that this requirement is much less strict than the demand for piecewise constant curvature in the original approach. Parts of the following results have been published in [HSS07b].

4.3.1 Time varying reference speed

Concerning the first problem it happens, especially for fast changing v_c, that the formation's shape is stretched or compressed along the reference trajectory and collisions between robots might occur. This phenomenon can be easily avoided by calculating v_i as

$$v_i(t) = \mathcal{S}_i \mathcal{Q}_i v_c(t), \tag{4.18}$$

which depends on the momentary speed of the reference point rather than on $v_c(s_i)$. On the other hand this requires that the entity that generates the reference trajectory immediately transmits the current speed v_c to all of the vehicles, making the approach inappropriate for groups of vehicles in which a communication link from the "leader" to every follower lacks to exist. Another solution would be to limit the change of v_c and therefore the acceleration for a certain interval of traveled distance. As a result the deformations along the reference trajectory will be bounded, but on the other hand the space of feasible reference trajectories is further constrained. Which of the modifications is more suitable depends on the intended application.

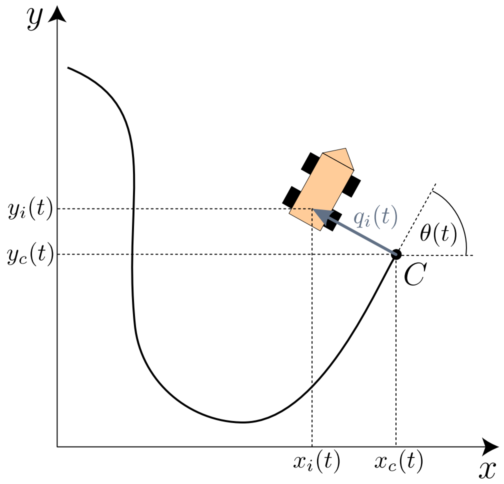


Figure 4.5: Position and orientation of the ith vehicle with respect to the reference point C.

4.3.2 Time varying reference curvature

To resolve the second problem it is necessary to derive the equations for v_i and K_i without the assumption that the curvature of the reference trajectory is piecewise constant. In the following we will show how to enhance the method in order to be capable of handling reference trajectories with time-varying curvature. In contrast to Section 4.2, in which the derivation of v_i and K_i was based on the relationship between Cartesian and curvilinear coordinates, we will start directly from the equations of motion (3.8). As will be seen in the following, this results in similar control input equations that are capable of handling reference trajectories with time-varying, continuously differentiable curvature.

Integration of the kinematic equations (3.8) results in a canonical expression of the equations of odometric dead reckoning. With these formulas and the assumption that $p_i(t) = 0$ for the ith vehicle, its configuration at time t can be written as:

$$\begin{aligned}
x_i(t) &= x_c(t) - q_i(t)\sin\theta(t) \\
&= \int_0^t v_c(\tau)\cos\theta(\tau)d\tau - q_i(t)\sin\theta(t), \\
y_i(t) &= y_c(t) + q_i(t)\cos\theta(t) \\
&= \int_0^t v_c(\tau)\sin\theta(\tau)d\tau + q_i(t)\cos\theta(t), \\
\theta(t) &= \int_0^t K_c(\tau)v_c(\tau)d\tau.
\end{aligned}$$

Figure 4.5 schematically depicts the momentary configuration of the ith vehicle relative to the reference point C.

From here, the remaining steps in order to get to the equations for v_i and K_i are very similar to the steps described in Section 4.2. For readability, we again omit the dependency on t in the notation of the following equations. This time we obtain

$$\begin{aligned}
\dot{x}_i &= \cos\theta(v_c - q_i\dot{\theta}) - \sin\theta\dot{q}_i,\\
\dot{y}_i &= \sin\theta(v_c - q_i\dot{\theta}) + \cos\theta\dot{q}_i,\\
\dot{\theta} &= K_c v_c,
\end{aligned}$$

for the first derivatives in time and

$$\begin{aligned}
\ddot{x}_i &= \cos\theta(\dot{v}_c - 2\dot{q}_i\dot{\theta} - q_i\ddot{\theta}) - \sin\theta(\dot{\theta}(v_c - q_i\dot{\theta}) + \ddot{q}_i),\\
\ddot{y}_i &= \sin\theta(\dot{v}_c - 2\dot{q}_i\dot{\theta} - q_i\ddot{\theta}) + \cos\theta(\dot{\theta}(v_c - q_i\dot{\theta}) + \ddot{q}_i),\\
\ddot{\theta} &= v_c\dot{K}_c + K_c\dot{v}_c,
\end{aligned}$$

for the second derivatives in time. If we insert the above equations into the general formulas for velocity (4.5) and curvature (4.9) along a path in parametric form, we obtain the control inputs for the ith vehicle at time t as

$$\begin{aligned}
v_i &= \sqrt{\big(K_c q_i - 1\big)^2 v_c^2 + \dot{q}_i^2},\\
K_i &= \frac{1}{v_i^3}\Big(K_c^3 q_i^2 v_c^3 - 2K_c^2 q_i v_c^3 + \ddot{q}_i v_c\\
&\qquad + \dot{q}_i(q_i v_c \dot{K}_c - \dot{v}_c) + K_c\big(v_c^3 + v_c(2\dot{q}_i^2 - q_i\ddot{q}_i) + q_i\dot{q}_i\dot{v}_c\big)\Big).
\end{aligned}$$

It is straightforward to perform a change of variable from time t to the traveled distance s along the reference trajectory. Considering that $\dot{s}_i = v_c(s_i)$ as well as equation (4.11) and the equalities

$$\frac{\dot{q}_i}{\dot{s}_i} = \frac{dq_i}{ds_i}, \qquad \frac{\dot{K}_c}{\dot{s}_i} = \frac{dK_c}{ds_i},$$

we obtain the following control inputs:

$$v_i(s_i) = \mathcal{Q}_i(s_i) v_c(s_i), \tag{4.19}$$

$$K_i(s_i) = \frac{1}{\mathcal{Q}_i(s_i)}\left(K_c(s_i) + \frac{K_c(s_i)\big(\frac{dq_i}{ds_i}(s_i)\big)^2}{\mathcal{Q}_i(s_i)^2} + \frac{\big(1 - q_i(s_i)K_c(s_i)\big)\frac{d^2q_i}{ds_i^2}(s_i) + q_i(s_i)\frac{dq_i}{ds_i}(s_i)\frac{dK_c}{ds_i}(s_i)}{\mathcal{Q}_i(s_i)^2}\right), \tag{4.20}$$

where s_i is calculated just like in equation (4.1) and

$$\mathcal{Q}_i(s_i) := \sqrt{\left(\frac{dq_i}{ds_i}(s_i)\right)^2 + \Big(1 - q_i(s_i)K_c(s_i)\Big)^2}$$

is the same as in equation (4.8) except that K_c is allowed to be varying.

After incorporating the factor $\mathcal{S}_i$ defined in (4.13) to these equations, it can be seen that the varying curvature has no influence on the robot's speed, because the derived formula for the speed is the same as equation (4.14) deduced for the piecewise constant case. In comparison to equation (4.15), there exists an additional term in the expression for K_i, which takes care of the changing curvature of the reference trajectory. Assuming that $dK_c/ds_i = 0$, i.e., the curvature K_c along the reference trajectory is constant, equation (4.20) simplifies to equation (4.15) which corresponds to the piecewise constant case.

By utilizing equations (4.20) and (4.18) it is now possible to move a group of nonholonomic robots with kinematics described by (3.8) in a possibly changing formation along trajectories with time-varying curvature. Of course, for the derivative dK_c/ds_i to be defined, the reference curvature needs to be at least C^1-continuous. Furthermore, the constraints in equation (3.9) must be satisfied for all vehicles at every point of their trajectory in order to be able to maintain the formation.

4.3.3 Adequate adjustment of relative position coordinates

The adjustment of the ith vehicle's relative coordinates p_i and q_i as it is described in Section 4.2 results in a non-continuous trajectory. But since a real vehicle cannot adjust its speed nor its curvature from one value to another in an instant of time, continuous profiles for v_i and K_i are desired. In the following, one possible way is described how to obtain C^1-continuous speed profiles and at least C^0-continuous curvature profiles for the follower vehicles.

In order to effect changes of the ith robot's p-coordinate, which is measured along the reference trajectory, one simply needs to increase or decrease the robot's current speed v_i as it was shortly mentioned in Section 4.2. To formalize this we introduce the variables $t_{i,o}$ and $t_{i,f}$ denoting the adjustment's starting and ending time, where $t_{i,f} > t_{i,o}$. Further, we set $p_{i,o}$ to be the initial value of p_i and $p_{i,f}$ to be its final value respectively. With these values we now define the function

$$v_i(t) = \begin{cases} \mathcal{Q}_i\big(s_i(t)\big)\big(v_c(t) + \psi_{t_{i,o},(t_{i,o}+\Delta t/4),0,\tilde{v}_c}(t)\big) & t_{i,o} < t < t_{i,o} + \frac{\Delta t}{4} \\ \mathcal{Q}_i\big(s_i(t)\big)\big(v_c(t) + \tilde{v}_c\big) & t_{i,o} + \frac{\Delta t}{4} \leq t < t_{i,o} + \frac{3\Delta t}{4} \\ \mathcal{Q}_i\big(s_i(t)\big)\big(v_c(t) + \psi_{(t_{i,o}+3\Delta t/4),t_{i,f},\tilde{v}_c,0}(t)\big) & t_{i,o} + \frac{3\Delta t}{4} \leq t < t_{i,f} \\ \mathcal{Q}_i\big(s_i(t)\big)v_c(t) & \text{otherwise,} \end{cases} \tag{4.21}$$

where

$$\tilde{v}_c = \frac{4}{3}\frac{p_{i,f} - p_{i,o}}{t_{i,f} - t_{i,o}}, \qquad \Delta t = t_{i,f} - t_{i,o},$$

and ψ corresponds to function (4.16). In principle this enhancement of v_i results in a C^1-continuous decrease (increase) of the speed relative to the vehicle's reference in the first quarter of the time interval $[t_{i,o}, t_{i,f}]$. During the second and the third quarter, the speed's modification is kept equal to the constant $\tilde{v}_c$, and finally it returns to zero along a continuously differentiable curve in the fourth quarter. If the modification increases or decreases depends on the desired change of p_i, i.e., if $p_{i,o} < p_{i,f}$ it will increase in the first quarter of $[t_{i,o}, t_{i,f}]$ and decrease in the fourth quarter and *vice versa.* During such maneuvers p_i is evolving as described by

$$\dot{p}_i = \frac{v_i}{\mathcal{Q}_i} - v_c. \tag{4.22}$$

To verify the behavior of equation (4.21) we can calculate the integral of (4.22) during the adjustment:

$$\begin{aligned}
\int_{t_{i,o}}^{t_{i,f}} \dot{p}_i \, dt &= \int_{t_{i,o}}^{t_{i,f}} \frac{v_i}{\mathcal{Q}_i} - v_c \, dt \\
&= \int_{t_{i,o}}^{t_{i,o}+\frac{\Delta t}{4}} \psi_{t_{i,o},(t_{i,o}+\Delta t/4),0,\tilde{v}_c}(t) \, dt + \int_{t_{i,o}+\frac{\Delta t}{4}}^{t_{i,o}+\frac{3\Delta t}{4}} \tilde{v}_c \, dt \\
&\quad + \int_{t_{i,o}+\frac{3\Delta t}{4}}^{t_{i,f}} \psi_{(t_{i,o}+3\Delta t/4),t_{i,f},\tilde{v}_c,0}(t) \, dt \\
&= \Bigg(- 64t(p_{i,o} - p_{i,f}) \\
&\qquad \cdot \frac{-2t^3 - 3t_{i,o}t(3t_{i,o} + t_{i,f}) + t^2(7t_{i,o} + t_{i,f}) + t_{i,o}^2(5t_{i,o} + 3t_{i,f})}{3(t_{i,o} - t_{i,f})^4} \Bigg) \Bigg|_{t_{i,o}}^{t_{i,o}+\frac{\Delta t}{4}} \\
&\quad + \frac{4t(p_{i,f} - p_{i,o})}{3\Delta t} \Bigg|_{t_{i,o}+\frac{\Delta t}{4}}^{t_{i,o}+\frac{3\Delta t}{4}} \\
&\quad + \Bigg(64t(p_{i,o} - p_{i,f}) \\
&\qquad \cdot \frac{-2t^3 - 3t_{i,f}t(t_{i,o} + 3t_{i,f}) + t^2(t_{i,o} + 7t_{i,f}) + t_{i,f}^2(3t_{i,0} + 5t_{i,f})}{3(t_{i,o} - t_{i,f})^4} \Bigg) \Bigg|_{t_{i,o}+\frac{3\Delta t}{4}}^{t_{i,f}} \\
&= \frac{1}{6}(p_{i,f} - p_{i,o}) + \frac{2}{3}(p_{i,f} - p_{i,o}) + \frac{1}{6}(p_{i,f} - p_{i,o}) = (p_{i,f} - p_{i,o}).
\end{aligned}$$

From the value of the integral it can be seen that the change of p_i in the interval $[t_{i,o}, t_{i,f}]$ is equal to the desired amount. Figures 4.6 and 4.7 show two examples of adjustments of the relative position coordinate p_i while maintaining a C^1-continuous speed profile. In both examples q_i was set to 0, which results in $v_i = v_c$ for all $t \leq t_{i,o}$ and $t \geq t_{i,f}$. In Figure 4.6 the reference trajectory follows a root-function and the maneuver constants were chosen as $p_{i,o} = -2$m, $p_{i,f} = -3$m, $t_{i,o} = 1$s, and $t_{i,f} = 4$s. Note the negative value of v_i, which makes

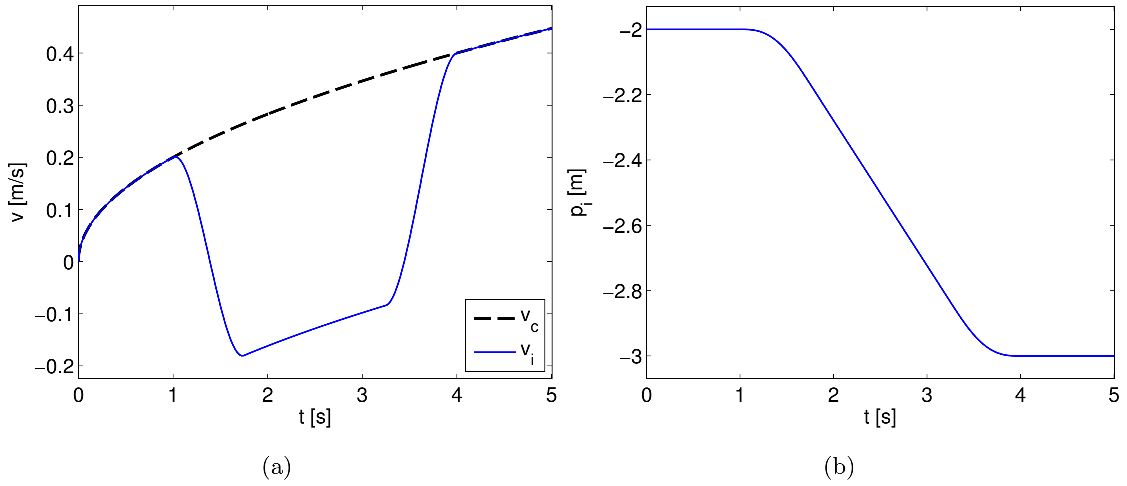


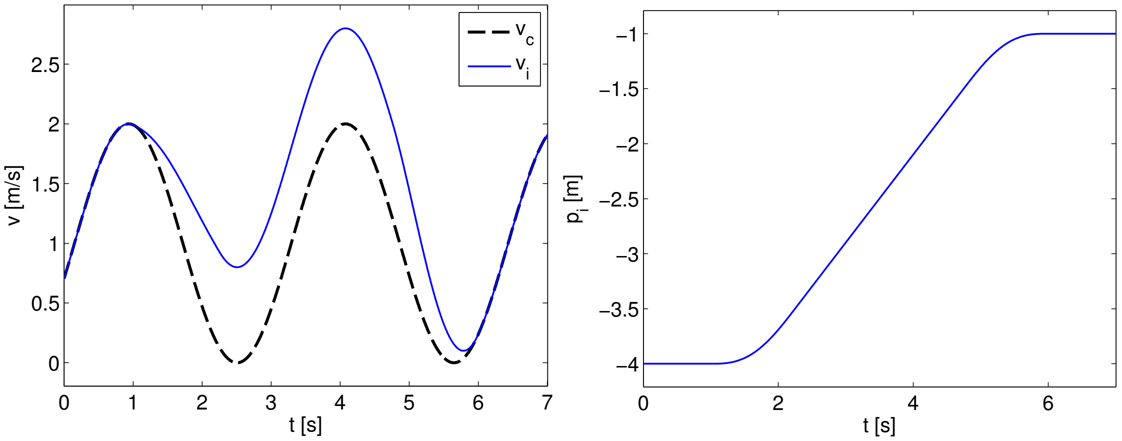
Figure 4.6: Example 1: adjustment of the relative position coordinate p_i while the reference speed follows a root-function. (a) shows the speed profiles of the reference trajectory and of vehicle i and (b) shows the development of p_i.

the robot temporarily move backwards. This is necessary in order to achieve a decrease of 1m within 3 seconds at such a low reference speed v_c. In the second example shown in Figure 4.7 the reference speed has a sinusoidal profile and the maneuver constants were chosen as $p_{i,o} = -4$m, $p_{i,f} = -1$m, $t_{i,o} = 1$s, and $t_{i,f} = 6$s.

It should be mentioned that a linear change of speed in the first and in the fourth quarter of interval $[t_{i,o}, t_{i,f}]$ results in the same final value for p_i. Nevertheless, in the connection points at times $t_{i,o} + \Delta t/4$ and $t_{i,o} + 3\Delta t/4$ the speed function v_i would be just of class C^0. Physically this means that the vehicle will encounter abrupt changes in acceleration which are known to be uncomfortable for possible occupants.

In Section 4.2 the function (4.17) was introduced with the intention to be utilized for adjustments of the relative position variable q_i. Since the function is of class C^1 its second derivative, which is needed to compute K_i as defined in (4.20) or in (4.15), is not defined everywhere. The central part of equation (4.17) is a third order polynomial function connecting the points $(s_0, \psi_0)^T$ and $(s_f, \psi_f)^T$. At these points ψ is only once differentiable. In the remainder of this section we concentrate on how to construct a higher order spline function as replacement for the third order polynomial in (4.17) that features a better overall smoothness.

In order to compute K_i as in equation (4.20) the first derivative of the reference curvature dK_c/ds_i needs to exist everywhere. If this derivative is C^0, i.e., $K_c \in$ C^1, then the profile of K_i cannot be smoother than C^0. Hence it would be sufficient if the spline function will be of class C^2. As a result K_i would always be C^0-continuous in the starting and ending points



(a) (b)

Figure 4.7: Example 2: adjustment of the relative position coordinate p_i along a sinusoidal reference speed profile. (a) shows the speed profiles of the reference trajectory and of vehicle i and (b) shows the development of p_i.

of adjustments of q_i. A guaranteed continuously differentiable curvature profile is obtained, if K_c is at least of class C^2 and the q_i-adjusting spline function is at least of class C^3. In order to construct a C^3-continuous spline that connects the points $(s_0, \psi_0)^T$ and $(s_f, \psi_f)^T$ we define

$$\xi_{s_o,s_f,\xi_0,\xi_f}(s) = \begin{cases} \tilde{\xi}_1(s) & s_o \le s \le S_1 \\ \tilde{\xi}_2(s) & S_1 < s \le S_2 \\ \tilde{\xi}_3(s) & S_2 < s \le s_f \end{cases}, \tag{4.23}$$

with

$$\begin{aligned} \tilde{\xi}_i(s) &= a_i s^5 + b_i s^4 + c_i s^3 + d_i s^2 + e_i s + f_i, \\ S_1 &= s_o + (s_f - s_o)/4, \\ S_2 &= s_o + 3(s_f - s_o)/4. \end{aligned}$$

To figure out the equation's coefficients we will now set up a set of equality constraints. At first we have to assure that the function values at the starting point s_0 and the ending point s_f of the function are equal to ξ_0 and ξ_f and the first three derivatives in these points are equal to zero. Thus, for the starting point we have

$$\tilde{\xi}_1(s_0) = a_1 s_0^5 + b_1 s_0^4 + c_1 s_0^3 + d_1 s_0^2 + e_1 s_0 + f_1 = \xi_0, \tag{4.24}$$

$$\tilde{\xi}_1'(s_0) = 5a_1 s_0^4 + 4b_1 s_0^3 + 3c_1 s_0^2 + 2d_1 s_0 + e_1 = 0, \tag{4.25}$$

$$\tilde{\xi}_1''(s_0) = 20a_1 s_0^3 + 12b_1 s_0^2 + 6c_1 s_0 + 2d_1 = 0, \tag{4.26}$$

$$\tilde{\xi}_1^{(3)}(s_0) = 60a_1 s_0^2 + 24b_1 s_0 + 6c_1 = 0, \tag{4.27}$$

and for the ending point

$$\tilde{\xi}_3(s_f) = \xi_f, \tag{4.28}$$

$$\tilde{\xi}'_3(s_f) = 0, \tag{4.29}$$

$$\tilde{\xi}''_3(s_f) = 0, \tag{4.30}$$

$$\tilde{\xi}^{(3)}_3(s_f) = 0, \tag{4.31}$$

respectively. Another eight equations result from the need to connect the polynomials piecewise to a C^3-continuous spline. Thus, for S_1 we have

$$\tilde{\xi}_1(S_1) = \tilde{\xi}_2(S_1), \tag{4.32}$$

$$\tilde{\xi}'_1(S_1) = \tilde{\xi}'_2(S_1), \tag{4.33}$$

$$\tilde{\xi}''_1(S_1) = \tilde{\xi}''_2(S_1), \tag{4.34}$$

$$\tilde{\xi}^{(3)}_1(S_1) = \tilde{\xi}^{(3)}_2(S_1), \tag{4.35}$$

and for S_2 we have

$$\tilde{\xi}_2(S_2) = \tilde{\xi}_3(S_2), \tag{4.36}$$

$$\tilde{\xi}'_2(S_2) = \tilde{\xi}'_3(S_2), \tag{4.37}$$

$$\tilde{\xi}''_2(S_2) = \tilde{\xi}''_3(S_2), \tag{4.38}$$

$$\tilde{\xi}^{(3)}_2(S_2) = \tilde{\xi}^{(3)}_3(S_2). \tag{4.39}$$

As there are 16 equations in 18 unknown variables, we have to add two more equations in order to obtain a unique solution. For this we chose to decrease the order of the first and the third polynomial by one, which means

$$a_1 = 0, \tag{4.40}$$

$$a_3 = 0. \tag{4.41}$$

Now we can solve the linear equation system (4.24)-(4.41) in order to obtain the coefficients of (4.23). See Appendix A for the solution. Note that a C^2-continuous spline function connecting the points $(s_0, \psi_0)^T$ and $(s_f, \psi_f)^T$ can be constructed in the same way from three third order polynomials.

With the constructed spline function of class C^3 we can now set

$$q_i(s_i) = \begin{cases} q_{i,o} & 0 \leq s_i \leq s_{i,o} \\ \xi_{s_{i,o}, s_{i,f}, q_{i,0}, q_{i,f}}(s_i) & s_{i,o} < s_i \leq s_{i,f} \\ q_{i,f} & s_{i,f} < s_i \end{cases}, \tag{4.42}$$

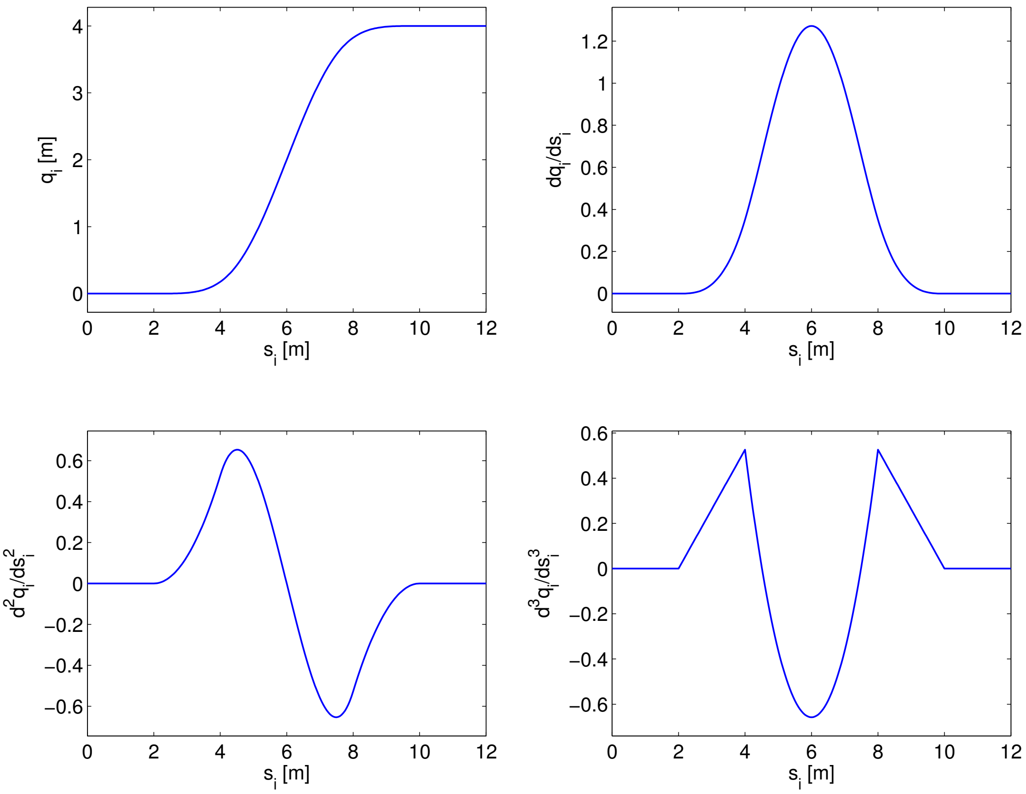


Figure 4.8: Plot of the C^3-continuous function (4.42) intended for adjustments of the ith robot's q-coordinate and its first three derivatives. The constants were chosen as $s_{i,o} = 2$m, $s_{i,f} = 10$m, $q_{i,o} = 0$m, and $q_{i,f} = 4$m.

which can be utilized to adjust the ith robot's q-coordinate while generating a C^1 or C^0-continuous profile for K_i depending on the continuity class of the reference curvature K_c. An exemplary plot of equation (4.42) can be found in Figure 4.8, where the constants were chosen as $s_{i,o} = 2$m, $s_{i,f} = 10$m, $q_{i,o} = 0$m, and $q_{i,f} = 4$m.

4.4 Simulation results

In this section we present the simulation results of two exemplary scenarios to highlight the abilities of the enhanced motion planning approach. Therefore we define a reference trajectory as well as a collection of maneuver constants that describe the desired modifications of the vehicles p and q-coordinates during the motion. The results of the simulations are depicted in Figures 4.9-4.11. The topmost plot in each figure shows the trajectories of the vehicles and the reference in the x-y-plane as well as snapshots of the vehicles' configuration at specific points in time. The remaining figure contains graphs of the speeds v_c and v_i and the relative coordinates p_i and q_i plotted against the time as well as the curvatures K_c and K_i plotted

i	$\tilde{s}^{I}_{i,o}$	Δt^{I}	$p^{I}_{i,o}$	$p^{I}_{i,f}$	$\tilde{s}^{II}_{i,o}$	Δt^{II}	$p^{II}_{i,o}$	$p^{II}_{i,f}$
	[m]	[s]	[m]	[m]	[m]	[s]	[m]	[m]
1	9	3	-1	-2	32	3	-2	-1
2	21	3	-2	-1	40	3	-1	-2

i	$s^{I}_{i,o}$	$s^{I}_{i,f}$	$q^{I}_{i,o}$	$q^{I}_{i,f}$
	[m]	[m]	[m]	[m]
1	8	11	0	-1
2	8	11	0	1

i	$s^{II}_{i,o}$	$s^{II}_{i,f}$	$q^{II}_{i,o}$	$q^{II}_{i,f}$	$s^{III}_{i,o}$	$s^{III}_{i,f}$	$q^{III}_{i,o}$	$q^{III}_{i,f}$	$s^{IV}_{i,o}$	$s^{IV}_{i,f}$	$q^{IV}_{i,o}$	$q^{IV}_{i,f}$
	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
1	25	28	-1	0	32	35	0	1	40	44	1	0
2	25	28	1	0	32	35	0	-1	40	44	-1	0

Table 4.1: Maneuver constants for formation motion planning simulation scenario 1.

against the traveled distance along the reference trajectory. The vertical dashed lines in these plots denote the points in time at which the vehicle snapshots in the x-y-plot were taken. The bold black curves in the speed and the curvature plot correspond to the values of the reference point v_c and K_c.

4.4.1 Scenario 1

In the first scenario we compare the motion planning approach that assumes a piecewise constant reference curvature described in Section 4.2 with the enhanced motion planning approach proposed in Section 4.3. Therefore we evaluated both methods in the same scenario, which means both methods have to cope with the same reference trajectory defined by $v_c(s)$ and $K_c(s)$ as well as identical modifications of p_i and q_i. The maneuver constants specifying the adjustments of the relative coordinates are listed in Table 4.1. We decided to let the starting time $t_{i,o}$ of a p-modification be the time when the corresponding car reaches a certain distance $\tilde{s}_{i,o}$ measured along the reference trajectory for the first time. The trigger has to be limited to the first time, because the adjustment might cause the vehicle to move backwards, thus decreasing its traveled distance possibly below $\tilde{s}_{i,o}$. So the vehicle might pass by the same point at different times. The reason for not choosing fixed times is simply the easier synchronization with q-modifications which have to depend on the traveled distance. Note that it is important to apply adjustments of q_i every time the vehicle's traveled distance s_i resides in the corresponding interval $[s_{i,o}, s_{i,f}]$.

The simulation results for the method that assumes a piecewise constant reference curvature are depicted in Figure 4.9 and the results for the enhanced method are depicted in Figure 4.10 respectively. At the beginning of the simulation run the reference trajectory was set to follow a sinusoidal curve for the speed v_c while keeping the curvature K_c constant. This shows

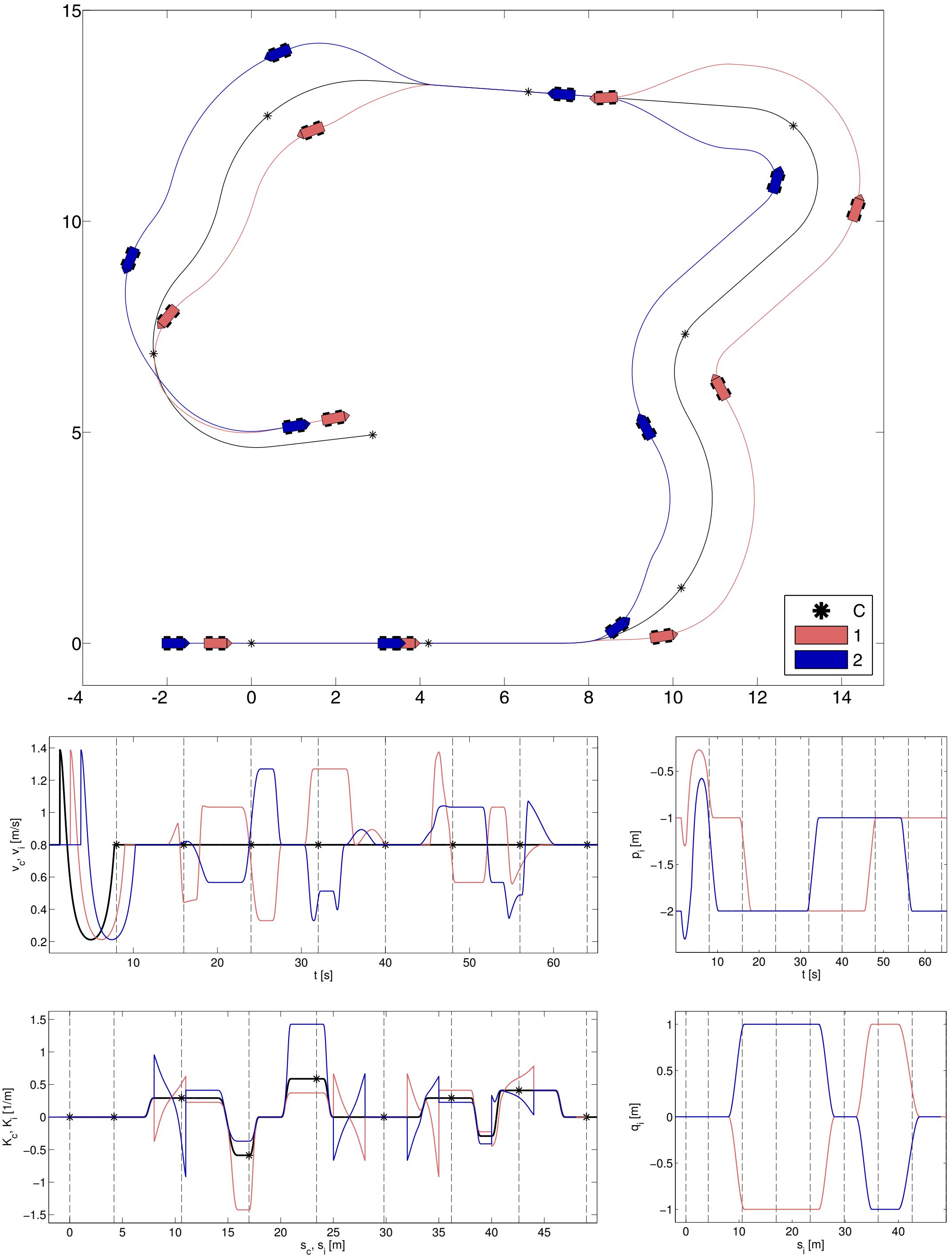


Figure 4.9: Motion planning simulation scenario 1: method assuming a piecewise constant reference curvature. The figure depicts the vehicle trajectories in the plane, the speed and the curvature profile as well as the values of p_i and q_i.

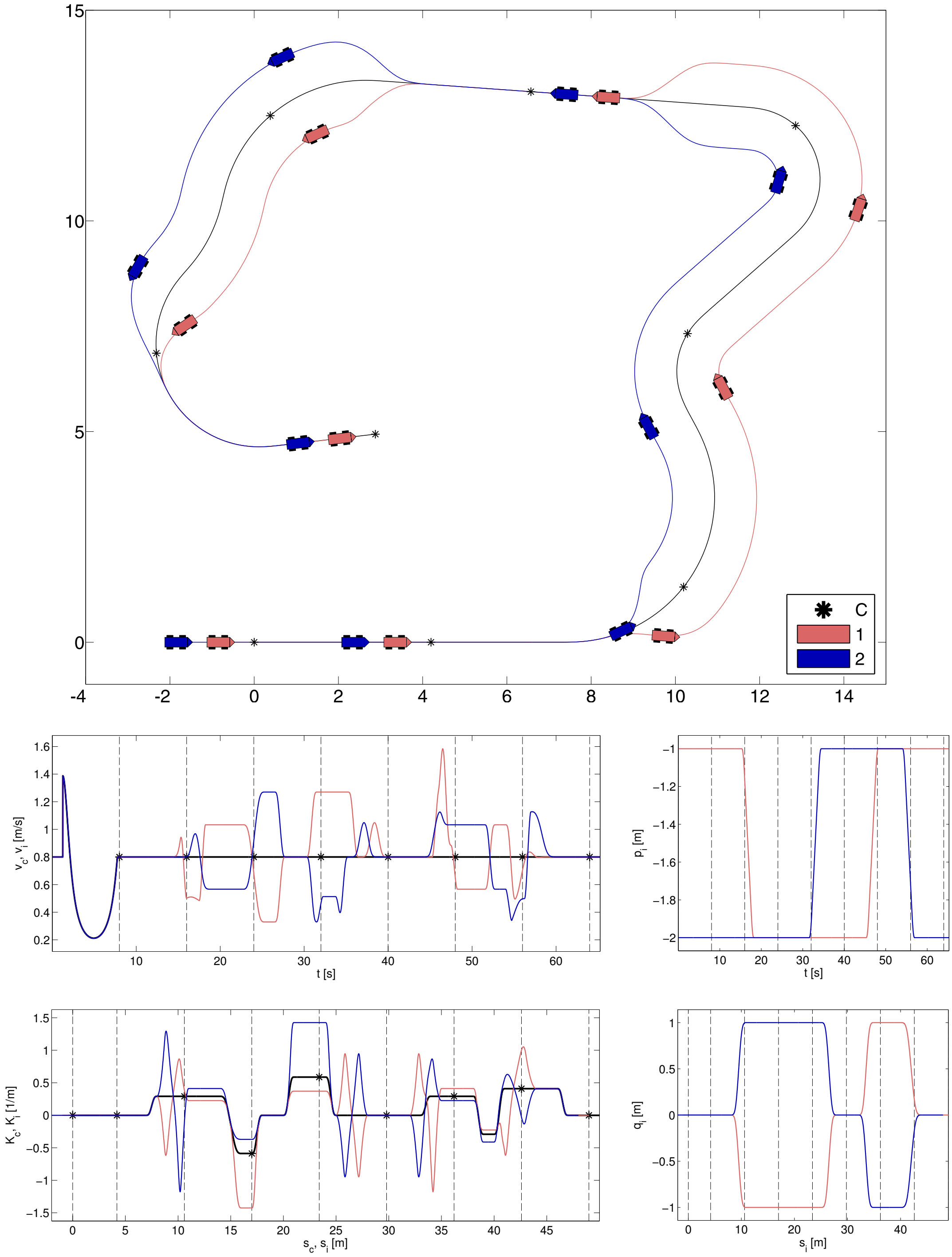


Figure 4.10: Motion planning simulation scenario 1: enhanced method. The figure depicts the vehicle trajectories in the plane, the speed and the curvature profile as well as the values of p_i and q_i.

well how the p-coordinates of the followers change with the original approach even though no modifications were projected as can be seen from Table 4.1. The enhanced approach in contrast maintains the formation, since all vehicles are applying the same reference speed at the same time. In order not to penalize the first method, the remainder of the reference trajectory was set to maintain a constant speed. The bends in the trajectory are composed of a continuously increasing (decreasing) curvature part, a part with constant curvature and finally a part where the curvature moves back to zero such that the profile of K_c is C^2-continuous. The first two desired modifications of q_i were placed on trajectory intervals with constant curvature, while p_i is changed one time during and one time before the q-adjustment. From the x-y-plots one can see that both methods manage the task well. Note the jumps (vertical parts) in the curvature profile of Figure 4.9 resulting from the application of equation (4.17) in comparison to the continuously differentiable curvature profiles of the enhanced approach in Figure 4.10.

In contrast to the first two adjustments of q_i the third and fourth were carried out on trajectory intervals with varying curvature, namely when a straight trajectory part passes into a part with nonzero constant curvature or *vice versa*. As expected, the method that assumes a piecewise constant reference curvature fails to manage these modifications of the formation's arrangement (cf. x-y-diagram in Figure 4.9). On the other hand, one can see in the x-y-diagram in Figure 4.10 that the enhanced approach handles the q-adjustments without problems. Note that the function values of p_i and q_i in Figure 4.9 do not correspond to the real relative positions seen in the x-y-plot. They rather describe the position relative to the reference point where the robots "think" they are located, because they are obtained as the output of equation (4.17) and the integral of equation (4.22). But in case of calculating K_i as in equation (4.15) on trajectory intervals with varying curvature this does not hold since the assumption of a constant reference curvature during q-adjustments is not satisfied.

4.4.2 Scenario 2

In the second scenario we demonstrate the abilities of the enhanced approach with a formation of six vehicles on a more complicated reference trajectory. In the results depicted in Figure 4.11 one can see that the reference speed v_c and the reference curvature K_c follow a heterodyne beat which naturally varies all the time. While in motion the vehicles will change from the initial column formation to a wedge shaped formation, which is followed by a formation of two lines each with three vehicles. After this, the vehicles rotate clockwise to the position of their neighbor and finally they get back to a column formation on the reference trajectory. All these maneuvers are defined by the maneuver constants found in Table 4.2.

i	$\tilde{s}^I_{i,o}$	Δt^I	$p^I_{i,o}$	$p^I_{i,f}$	$\tilde{s}^{II}_{i,o}$	Δt^{II}	$p^{II}_{i,o}$	$p^{II}_{i,f}$	$\tilde{s}^{III}_{i,o}$	Δt^{III}	$p^{III}_{i,o}$	$p^{III}_{i,f}$
	[m]	[s]	[m]	[m]	[m]	[s]	[m]	[m]	[m]	[s]	[m]	[m]
1	30	10	-1	-3	N/A				N/A			
2	12	8	-2	-1	20	10	-1	-3	28	10	-3	-6
3	4	8	-3	-2	12	8	-2	-1	N/A			
4	3	8	-4	-3	28	10	-3	-4	N/A			
5	2	8	-5	-3	18	10	-3	-1	30	10	-1	-2
6	1	8	-6	-3	28	10	-3	-5	N/A			

i	$s^I_{i,o}$	$s^I_{i,f}$	$q^I_{i,o}$	$q^I_{i,f}$	$s^{II}_{i,o}$	$s^{II}_{i,f}$	$q^{II}_{i,o}$	$q^{II}_{i,f}$	$s^{III}_{i,o}$	$s^{III}_{i,f}$	$q^{III}_{i,o}$	$q^{III}_{i,f}$
	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
1	20	25	0	-1	30	35	-1	0	N/A			
2	5	8	0	-0.5	12	16	-0.5	-1	28	33	-1	0
3	4	8	0	0.5	12	16	0.5	1	20	25	1	0
4	3	7	0	-1	18	23	-1	0	N/A			
5	2	7	0	1	30	35	1	0	N/A			
6	18	25	0	1	28	33	1	0	N/A			

Table 4.2: Maneuver constants for formation motion planning simulation scenario 2.

Here, like in the first scenario, the beginning of the p-adjustments is triggered by the first time the corresponding vehicle reaches a certain traveled distance $\tilde{s}_{i,o}$ measured along the reference trajectory. As can be seen, the applied motion generation approach manages the complete scenario without problems. For $t \geq 0$ and accordingly for $s \geq 0$ the resulting speed and curvature profiles in Figure 4.11 are continuously differentiable.

4.5 Summary and discussion

In this chapter we reviewed a motion planning approach for formations of nonholonomic mobile robots that is also applicable to vehicles of the car-like type. After pointing out weaknesses of the method, we introduced several enhancements that close the identified gaps. In the original method, a time-varying reference speed results in an undesired modification of the formation's shape that can be avoided by simply synchronizing the vehicles' speeds. Moreover, an extended equation has been derived for the curvature input calculation of the follower robots. This modification allows changes of the formation's shape along trajectory segments with continuously differentiable curvature in comparison to the need for constant curvature in the original approach. Finally, specific polynomial spline functions have been proposed

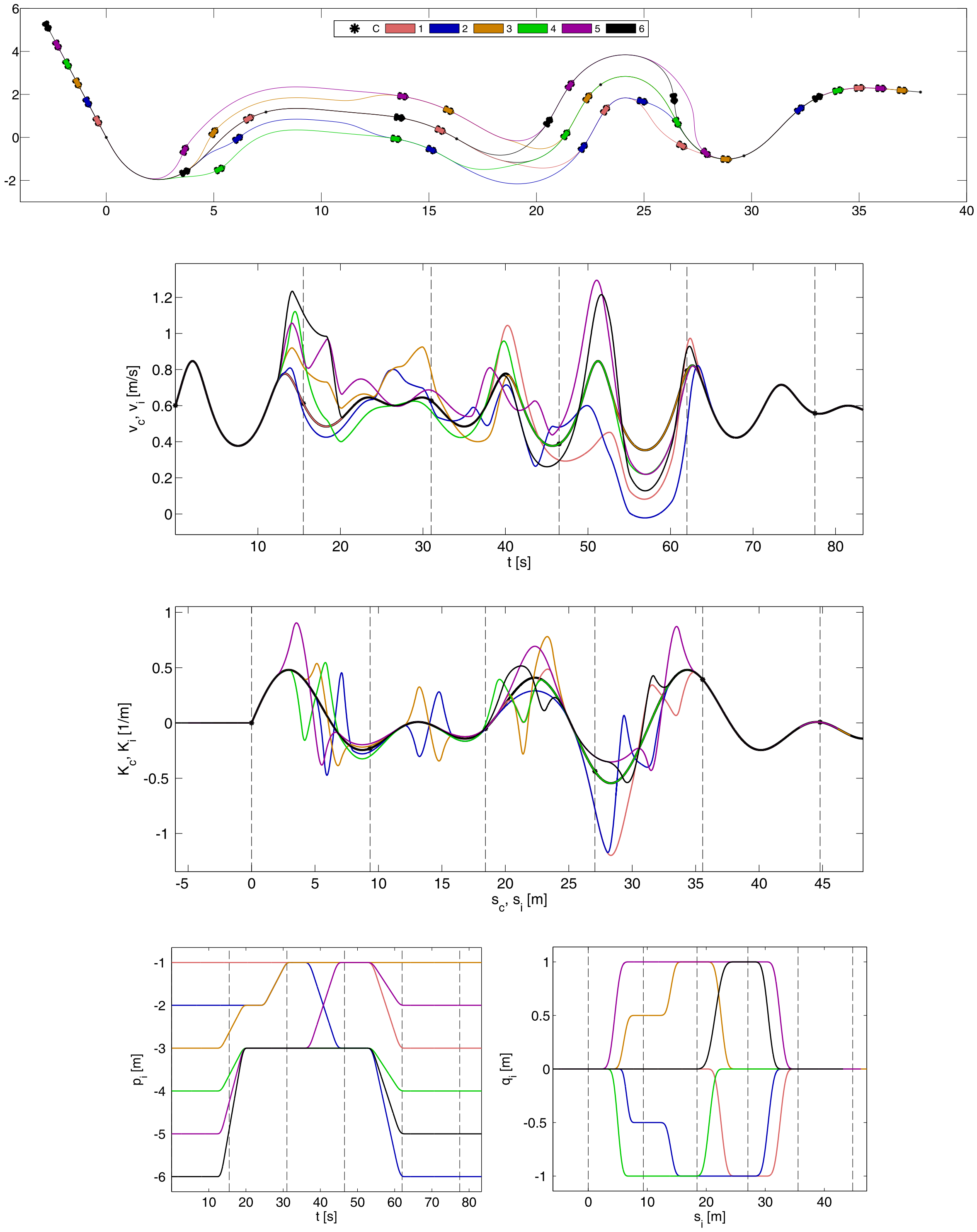


Figure 4.11: Motion planning simulation scenario 2: enhanced method. The figure depicts the vehicle trajectories in the plane, the speed and the curvature profile as well as the values of p_i and q_i.

for the relative position coordinates' alteration. Given C^0-continuous reference speed and C^1-continuous reference curvature, their application results in guaranteed continuous speed and steering profiles for the follower vehicles.

The motion planning approach calculates values for the vehicles' control inputs based on the corresponding values of a reference trajectory. However, in real world applications the control inputs cannot be applied without perturbations. In order to prevent the error from uncontrolled increase one has to include some kind of position feedback into the calculation of the control inputs. One way to achieve this, is to numerically integrate the kinematic model with the corresponding inputs and to apply a controller that tracks the resulting trajectory in the state space. In that case it is also possible to do the planning directly in absolute coordinates as it will be shown in Chapter 5 in the context of snow shoveling on airfields via formations of autonomous plows.

5 Airfield snow shoveling utilizing multi-vehicle formations

This chapter is concerned with the scenario of snow shoveling on airfields by utilizing multiple formations of autonomous snowplow robots. First, we describe the problem and review the related topic of cooperative sweeping. Then, we introduce the task specific concept of a centrally supervised system architecture with a leader-follower hierarchy on the motion coordination level. The basic idea of the method is to dynamically form formations of snowplows whose size depends on the width of the roads to clean. In the main part of this chapter, we develop solutions to the subproblems of motion planning for the formations and tracking control for the individual vehicles. For this we focus on suitable motion generation techniques that rely on the idea of maintaining the formations in curvilinear coordinates as it was proposed in Chapter 4. After this we derive a feedback controller in order to track the individually generated trajectories with the snowplows effector. Finally, implementation details and results from the simulations carried out are presented followed by the tracking controller's verification in laboratory hardware experiments.

5.1 Problem statement and related work

During the winter months, snowy weather has a huge influence on an airport's usual workflow. The air traffic needs to be interrupted from time to time in order to sweep away the snow from the airfield. This is crucial since snow is a potential risk for takeoff and landing aircrafts. Today the tracks of an airport are usually cleaned from snow by utilizing a fleet of human driven snowplows. Recent technological advances in the field of mobile robotics enable the setup of a multi-vehicle system consisting of groups of autonomous snowplow robots for this task. By arranging the vehicles in formations and applying a coordinated task allocation algorithm, such a system could significantly reduce the time of interrupted airport traffic.

The snow shoveling task is, from an application point of view, related to the field of cooperative sweeping, whose aim is to find and execute a motion for the sweeping robots in order to cover a predefined area by their effectors, which in our case are the snowplows' shovels. For cooperative sweeping it is often desired to coordinate the vehicles in a way that time optimality is achieved. Kurabayashi *et al.* addressed the problem of cooperative sweeping by generating a path for a single robot, which is then segmented and distributed among the vehicles [KOAY96]. The same group extended the approach with the robot's ability to relocate movable obstacles in [KOA$^+$96, AKOI96]. In contrast to this, Wagner *et al.* proposed an ant-like strategy for the cooperative cleaning of an unknown non-convex grid-map region [WB97]. The homogeneous robots with a limited amount of memory follow only local rules, which makes the method fully decentralized. Another decentralized approach using an on-line negotiation mechanism to resolve the task sharing was proposed in [MY98]. For this market-based strategy the authors assume that all-to-all communication is available. In [JN02] the authors use a partitioning algorithm that divides the area to be cleaned into a finite number of polygons. These are then allocated among the robots in a decentralized fashion. Both, the determination of the subareas and their assignment are done during the cleaning. Luo *et al.* developed a real-time method based on biologically inspired neural networks where each robot treats the other robots as moving obstacles [LY02, LYS03]. The sweeping of dynamic materials is investigated in [ABW05]. Here it is assumed that the material to clean spreads within the grid-based map. In [AS06] the authors propose a decomposition of the sweeping task by partitioning the area that is supposed to be cleaned. The approach is characterized by its focus on realizability, which is demonstrated in an experiment with real robots.

Projecting the problem of cooperative sweeping onto the structured environment of an airfield, the snow shoveling task generates some problems that have not yet been addressed in that context. The mentioned sweeping approaches lack robustness or rely on simplifications of the problem that makes them unusable for the airport sweeping problem. In particular it is assumed that the robots move on a grid or that they are able to move towards a certain goal point. In contrast, the approach described in the following considers the nonholonomic kinematics of usual snowplows as well as position and orientation of the plows' shovels. Most of the above mentioned methods also lack the ability to react on sudden changes in the environment which is important for the robust and safe execution of the sweeping process. Furthermore it has to be considered that the working space is more structured due to the airfield environment.

5.2 System overview

In this section we describe the overall system structure of our snow shoveling approach. We decided to rely on central supervision for the high level coordination of the system. The main

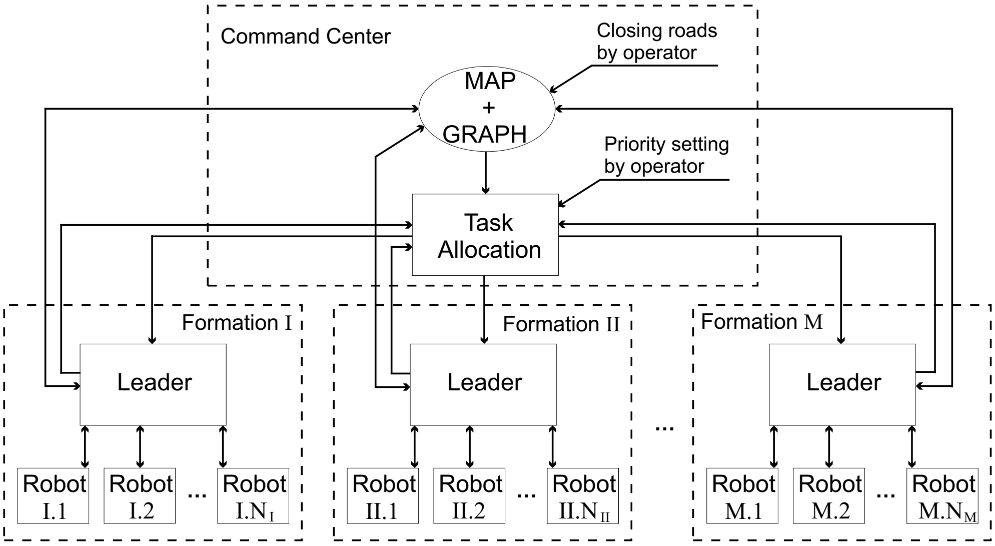


Figure 5.1: Scheme of the complete snow shoveling system. The arrows denote communication links between the different modules.

reason for this is safety, since a central command center has a complete overview of the whole system. The single-point-of-failure problem, which rises from a single command center can be dealt with one or two redundant command center units that take over in case of a failure. Another reason for the centralized approach is that the workspace of the snowplow robots is well known in size and structure. Therefore the scalability provided by a decentralized approach with agents exchanging parts of the map etc. is not necessary. Another drawback of the decentralized case would be the additional time needed for the sweeping process, since the task allocation has to be done with less information available.

The highest level of the proposed scheme (see Figure 5.1) is divided into two types of units. The first one, which we call the Command Center, is responsible for the central tasks. The second kind of unit (blocks denoted as Formation I to Formation M) represents the current constellation of vehicles where each unit corresponds to one formation with currently N_{I} to N_{M} vehicles. These units, which are independent from the Command Center most of the time, primarily are responsible for translating the assigned task into the appropriate formation motion. Note that the number of formations as well as the number of vehicles in a formation most likely changes during the sweeping process.

The core of the Command Center is the Task Allocation module that utilizes two data structures: MAP and GRAPH. The GRAPH data structure is prepared off-line from the MAP by assigning nodes to all intersection and edges for each road in between. Furthermore we assign two numbers to the edges of the graph: The first is an integer value equal to the number of plow robots needed for the sweeping of the corresponding road and the second number is equal to the road's length, which is assumed to be proportional to the necessary cleaning time. These values are used by the Task Allocation module in order to generate a reasonable output. Additionally we assign a flag to each edge that marks the already cleaned roads. In contrast to the static MAP, the GRAPH structure can be adjusted by a human operator closing and reopening roads according to the current airport traffic. Also plows that detect an obstacle on a road are meant to update the GRAPH structure.

The current GRAPH structure is used as an input for the Task Allocation module. Depending on the number of plows and the size of the roads, the module then generates one or multiple tasks that consist of information about the edge to be cleaned and the corresponding starting point. Further, it contains a list of the vehicles assigned to this task. Once these snowplow robots are gathered at the given starting point, they will start to remove the snow from the road that corresponds to the defined edge. An execution step of the Task Allocation module is triggered by snowplows that just accomplished (or failed) their current task, so they are waiting for new instructions. Besides the GRAPH, the module maintains actual plans of all coalitions as well as a priority setting for each road, which depends on the airport traffic as well as on the snowing intensity. Details about the functionality of the Task Allocation module can be found in [SHS08b].

For the Formation units we choose a leader-following structure, as it results in a good compromise between decentralization and control over individual vehicles. The Leader robot is responsible for generating a reference trajectory for its formation at the beginning of each task. This is done with the information received from the Task Allocation module and the appropriate coordinates from the MAP structure. When the trajectory generation is complete, the trajectory will be transmitted to the remaining robots within the group. The individual control inputs in order to follow the reference trajectory while maintaining the formation are calculated separately by each follower, where the Leader robot provides a signal for synchronization. As a result the Formation units are independent from the Command Center most of the time. Note that the Leader robot is just one designated robot in the formation, usually the one in front of the unit. Besides the snow shoveling the Leader acts as a connection between the robots and the Command Center. It informs the Command Center about detected obstacles, finished or aborted tasks as well as the need for additional robots to compensate for failures. Further note that the Leader robot is not meant to be the reference point in terms of the formation movement, in which it acts like the other robots of the group.

As will be seen in the following sections we divided the motion coordination problem into the subproblems of motion planning and trajectory tracking.

5.3 Motion planning

To be able to move the snowplow robots towards the points of interest deduced from the Task Allocation's output and the MAP, we have to construct a path that connects the points in the correct order. The resulting path has to be feasible for the formation of vehicles, which should track it during the shoveling task. Additionally it should be placed in a way that the sweeping plows create a clean track as wide as possible. The first state of the path is obtained by connecting the points of interest by line segments in the correct order. An example is shown in Figure 5.2, where the solid gray path refers to the line segment path connecting the points P_1 with P_2 and P_2 with P_3. Since this path cannot be followed by a snowplow, which is usually a vehicle of the car-like type, the sharp edges need to be replaced by curves. The dashed gray path in the figure characterizes the circular arc which connects the intersection points of lines a_1 and b_1 with the solid gray lines. Here, a_1 and b_1 intersect in the arc's center S_1 which lies on the bisecting line of the sharp edge at P_2. The intersection with the solid gray line segments is chosen at equal distance from P_2, which has to be large enough to guarantee a radius greater than or equal to the minimum turn radius of the vehicles.

Intuitively the dashed gray path seems to be traceable by a car-like vehicle with constant speed. But due to the discontinuous curvature in the connection points, its steering angle would need to change in an instant of time from 0 to a certain value or *vice versa*. One possible solution would be to stop the plow in the connection points and adjust the steering angle at zero speed, but that would make the approach very inefficient. The more elegant solution is to replace the sharp edges of the solid gray path by segments with continuous curvature (CC) like is has been described, e.g., in [FS04]. In Figure 5.2 such a CC-turn is depicted by the dashed black path. It consists of a clothoid arc reaching from the initial point of the turn (intersection with b_1) to the intersection with b_2, which is also chosen with respect to traceability. The arc's curvature varies from 0 to the constant curvature k of the second segment, which is a circular arc (from intersection with b_2 to intersection with a_2) with its center of curvature in S_2. The third and final segment (a_2 to a_1) is a clothoid arc of inverted sharpness compared to the first one whose curvature changes from k to 0. Due to the continuous curvature profile of the CC-turn the whole path can be followed without stopping. The final desired path with CC-turns can be parameterized by its arc length s resulting in two functions $x_c(s)$ and $y_c(s)$. To transform the path into a trajectory for the snowplow formations, a suitable speed profile $v_c(s)$ is assigned along the path. Note that the resulting trajectory is C^2-continuous with respect to x and y, what is essential for exact tracking as it will be described in Section 5.4.

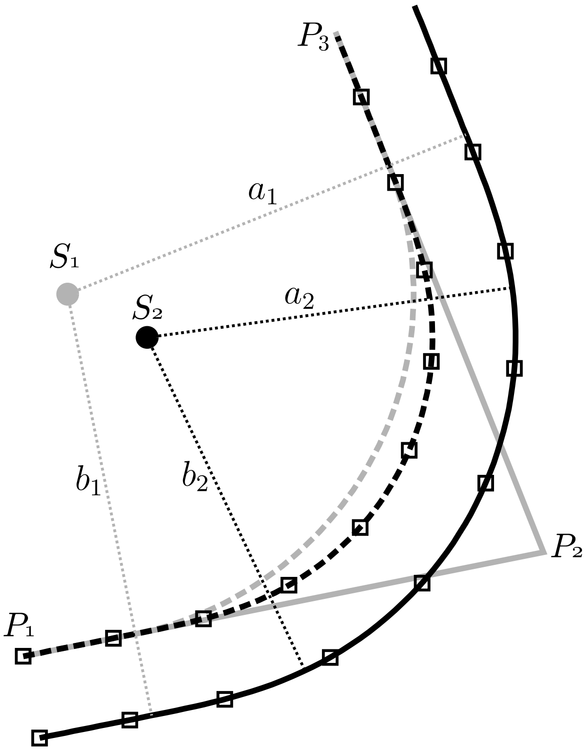


Figure 5.2: Different paths for connecting the points P_1, P_2 and P_3: Line segments (solid gray), line and circular arc-segments (dashed gray), line and clothoid arc segments (dashed black). The solid black path is shifted relatively to the dashed black reference path and might act as planned path for a vehicle of the formation.

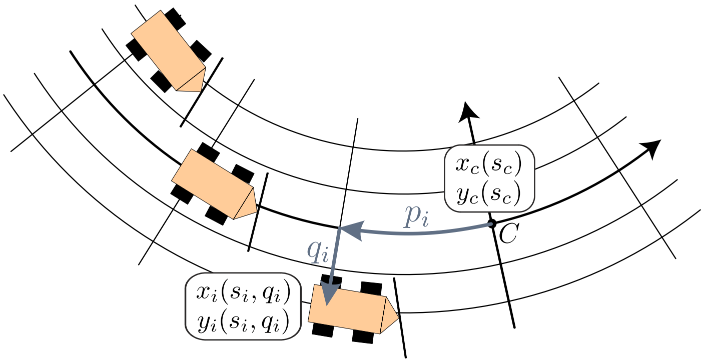


Figure 5.3: The formation of snowplows is maintained in curvilinear coordinates.

As it was mentioned in Section 4.1, methods for formation driving are often based on maintaining a certain distance and angle to a moving reference point which could be a leading vehicle, the barycenter of the group or even a predefined point. Since car-like vehicles are nonholonomic and therefore cannot accelerate in an arbitrary direction nor can they turn on the spot, this is not very applicable in the Cartesian space. However, the main problem is that the desired relative angle between the leader and a following vehicle changes, if they should drive through a curve one after the other on the same path. A solution to this problem is to maintain the distance to the reference point in curvilinear coordinates as in the motion planning approach that was described in Chapter 4. Compared to this method, here the reference trajectory is given in terms of the Cartesian coordinates $x_c(s)$ and $y_c(s)$ combined with a speed profile $v_c(s)$, rather than in terms of speed and curvature. This allows us to apply a feedback controller for trajectory tracking as it will be described in Section 5.4.

Figure 5.3 shows three snowplows in formation where C denotes the reference point and p_i and q_i denote the longitudinal and lateral distance from the ith vehicle to C respectively. p_i as well as the traveled distances of the reference point s_c and of the vehicles s_i are always measured along the reference path and therefore

$$s_i(t) = s_c(t) + p_i(t).$$

It can be seen that the Euclidean distances between the plows slightly vary depending on the momentary curvature of the reference trajectory. Due to a limited turning radius this deformation effect is upper bounded and therefore unproblematic if the initial values for p_i are chosen such that the distance between the vehicles is big enough. $x_c(s_i)$ and $y_c(s_i)$ provide the position of the reference point at the traveled distance of vehicle i. Then, with respect to q_i the desired position for the ith snowplow can be computed as

$$\begin{aligned} x_i^d(s_i, q_i) &= x_c(s_i) - q_i \sin(\hat{\theta}_i), \\ y_i^d(s_i, q_i) &= y_c(s_i) + q_i \cos(\hat{\theta}_i), \end{aligned}$$

where $\hat{\theta}_i$ is equal to the angle between the x-axis and a tangential vector at $\big(x_c(s_i), y_c(s_i)\big)^T$ pointing towards the movement direction. Next to the dashed black path in Figure 5.2 the solid black curve shows the computed path for a vehicle with $q_i < 0$. The quadratic markers on both curves are placed on equal traveled distances measured along the reference path. It can be seen that a vehicle on the solid path has to move faster in order to keep up with one on the dashed reference path. Apart from that its steering angle has to be smaller. Note that for traceability the vehicle's steering angle constraint needs to be satisfied at every point of the desired trajectory. But since the airfield was designed for big airplanes, whose turning

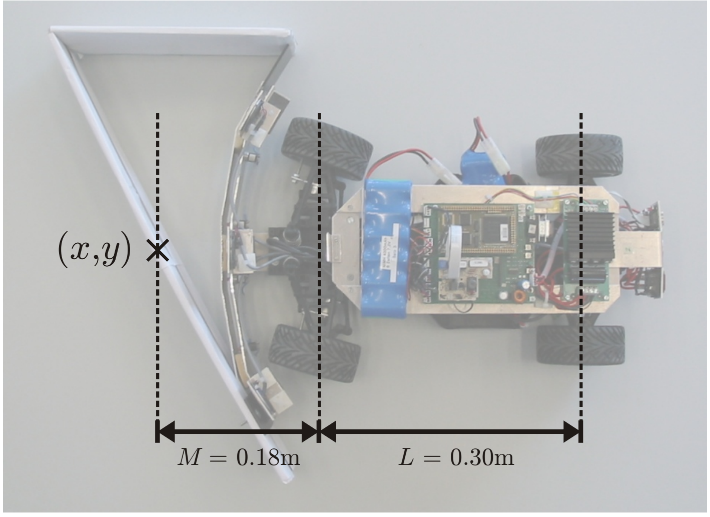


Figure 5.4: Topview of a MERLIN robot (cf. Appendix B) equipped with a shovel.

radius is usually more limited, we assume that the steering angle constraint of the snowplows is satisfied.

5.4 Adapted trajectory tracking control

To follow the desired trajectories a trajectory tracking controller developed for car-like robots was implemented on each snowplow. In [DOS98] a dynamic feedback controller is described that utilizes exact feedback linearization on the kinematic equations for the barycenter of the robot located at the center of its rear axle. The dynamic controller was intensively tested and showed its applicability except in sharp turns as can be seen in Figure 5.5(a). Note that it is not possible to influence the orientation of the vehicle when tracking a trajectory with a car-like robot. So the bigger the distance between shovel and barycenter of the plow, the more the cleaned area deviates from the desired path. To increase the cleaning performance we choose the shovel's mounting point as the new position that should track the desired trajectory. The steps of the controller's derivation will be described in the following.

The kinematic model for one vehicle with the new reference point can be written as

$$\begin{aligned}
\dot{x}(t) &= v(t)\cos\big(\theta(t)\big) - (L+M)\sin\big(\theta(t)\big)\dot{\theta}(t), && (5.1)\\
\dot{y}(t) &= v(t)\sin\big(\theta(t)\big) + (L+M)\cos\big(\theta(t)\big)\dot{\theta}(t), && (5.2)\\
\dot{\theta}(t) &= v(t)\frac{\tan\big(\phi(t)\big)}{L},\\
\dot{\phi}(t) &= \psi(t),
\end{aligned}$$

where x, y are the Cartesian coordinates of the shovel's mounting point (cf. Figure 5.4), θ denotes the orientation angle with respect to the x-axis, and ϕ denotes the steering angle. Further, L denotes the distance between front and rear axle and M corresponds to the distance between front axle and the shovel's mounting point. The two control inputs speed and angular velocity of the steering angle are referred to as v and ψ. Note that compared to the kinematic model of a usual car-like robot (3.8) we simply shifted the Cartesian coordinates $(x, y)^T$ to another location and added an integrator to the original control input ϕ.

In the following equations the dependency on time t is omitted due to readability. According to the tracking task we choose the system outputs as $(x, y)^T$. Now with $u = (v, \psi)^T$ we can write (5.1) and (5.2) as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\sin\theta(L+M)\dot{\theta} \\ \cos\theta(L+M)\dot{\theta} \end{pmatrix} + \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{pmatrix} u.$$

The input ψ does not appear and therefore the matrix weighting the inputs is singular. Like in [DOS98] we add an integrator with state ξ on the first input

$$v = \xi, \qquad \dot{\xi} = v', \tag{5.3}$$

with v' as auxiliary input. With (5.3) the second derivative of the output is computed as

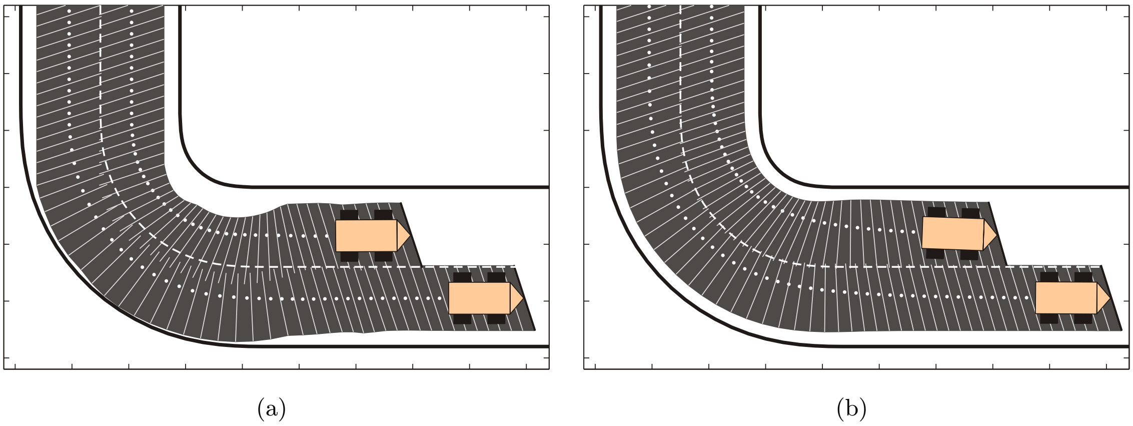
$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} -\xi^2 \frac{\tan\phi}{L^2}\big(L\sin\theta + (L+M)\cos\theta\tan\phi\big) \\ \xi^2 \frac{\tan\phi}{L^2}\big(L\cos\theta - (L+M)\sin\theta\tan\phi\big) \end{pmatrix} + \begin{pmatrix} P & R \\ Q & S \end{pmatrix} \begin{pmatrix} v' \\ \psi \end{pmatrix} \tag{5.4}$$

with

$$\begin{aligned} P &= \cos\theta - \left(1 + \frac{M}{L}\right)\sin\theta\tan\phi, \\ Q &= \sin\theta + \left(1 + \frac{M}{L}\right)\cos\theta\tan\phi, \\ R &= -\xi\left(1 + \frac{M}{L}\right)\frac{\sin\theta}{\cos^2\phi}, \\ S &= \xi\left(1 + \frac{M}{L}\right)\frac{\cos\theta}{\cos^2\phi}. \end{aligned}$$

Assuming $\xi \neq 0$, which makes the matrix weighting the inputs nonsingular, we set $(\ddot{x}, \ddot{y})^T = (r_1, r_2)^T$. Then solving equation (5.4) for the input vector $(v', \psi)^T$ yields the following nonlinear dynamic feedback controller

$$\begin{aligned} v &= \xi, \\ \psi &= \cos^2\phi\Big[4L^3(r_2\cos\theta - r_1\sin\theta) \\ &\quad - 4\big[\xi^2\big((L+M)^2/\cos^2\phi - M(2L+M)\big) \\ &\quad + L^2(L+M)(r_1\cos\theta + r_2\sin\theta)\big]\tan\phi\Big]/\big(4\xi L^2(L+M)\big), \\ \dot{\xi} &= r_1\cos\theta + r_2\sin\theta + \frac{(L+M)\xi^2\tan^2\phi}{L^2}. \end{aligned} \tag{5.5}$$



(a)

(b)

Figure 5.5: Comparison of barycenter tracking (a) and shovel center tracking (b) in simulation. The dots denote the path of the car's barycenters and the dashed line between the cars denotes the reference path. Further, the solid thin lines denote the shovels' position at certain time instances.

The linearization process transformed the original system into two double integrators

$$\begin{aligned}\ddot{x} &= r_1,\\ \ddot{y} &= r_2.\end{aligned}$$

Now the decoupled system can be stabilized with linear feedback of the form

$$\begin{aligned}r_1 &= \ddot{x}^d + K_{v,1}(\dot{x}^d - \dot{x}) + K_{p,1}(x^d - x),\\ r_2 &= \ddot{y}^d + K_{v,2}(\dot{y}^d - \dot{y}) + K_{p,2}(y^d - y),\end{aligned}$$

with the feedback gains such that the polynomials

$$\lambda^2 + K_{v,j}\lambda + K_{p,j}, \qquad j = 1, 2$$

are Hurwitz. As can be seen in Figure 5.5(b), the resulting controller handles sharp turns in a much more efficient way than the original one. Note that for those controllers to work properly an accurate positioning system is required that provides feedback of the position, and velocity errors. But since the site to clean is not changing, a reliable positioning system (e.g., DGPS, landmarks) can be installed on site for feedback purposes.

The task allocation algorithm might command vehicles to move into the direction they just came from. This can be due to a blocked road or simply because it is the most efficient route towards the upcoming task. Because of the limited steering angle of car-like robots, turning the vehicle around is not a trivial task. If there is enough space at the vehicles current position we let them turn around before heading towards the next task. If there is too few space available, the vehicles should move backwards towards the previous intersection they passed. This will be no problem, because the snow has been removed recently. Once the plows reach the intersection they can reorient in order to proceed forward towards their currently assigned task. An example for such a maneuver is shown in the simulation snapshots in Section 5.5. Note that for the backward motion the vehicles need to switch to a different controller, because otherwise they would turn around and switch to a forward motion as a result of minimal disturbances. The backward motion controller can be derived in the same way as the one for forward motion by simply negating the factor $(L + M)$ in the kinematic equations (5.1) and (5.2). As a result, the reference point for the tracking is located behind the plow.

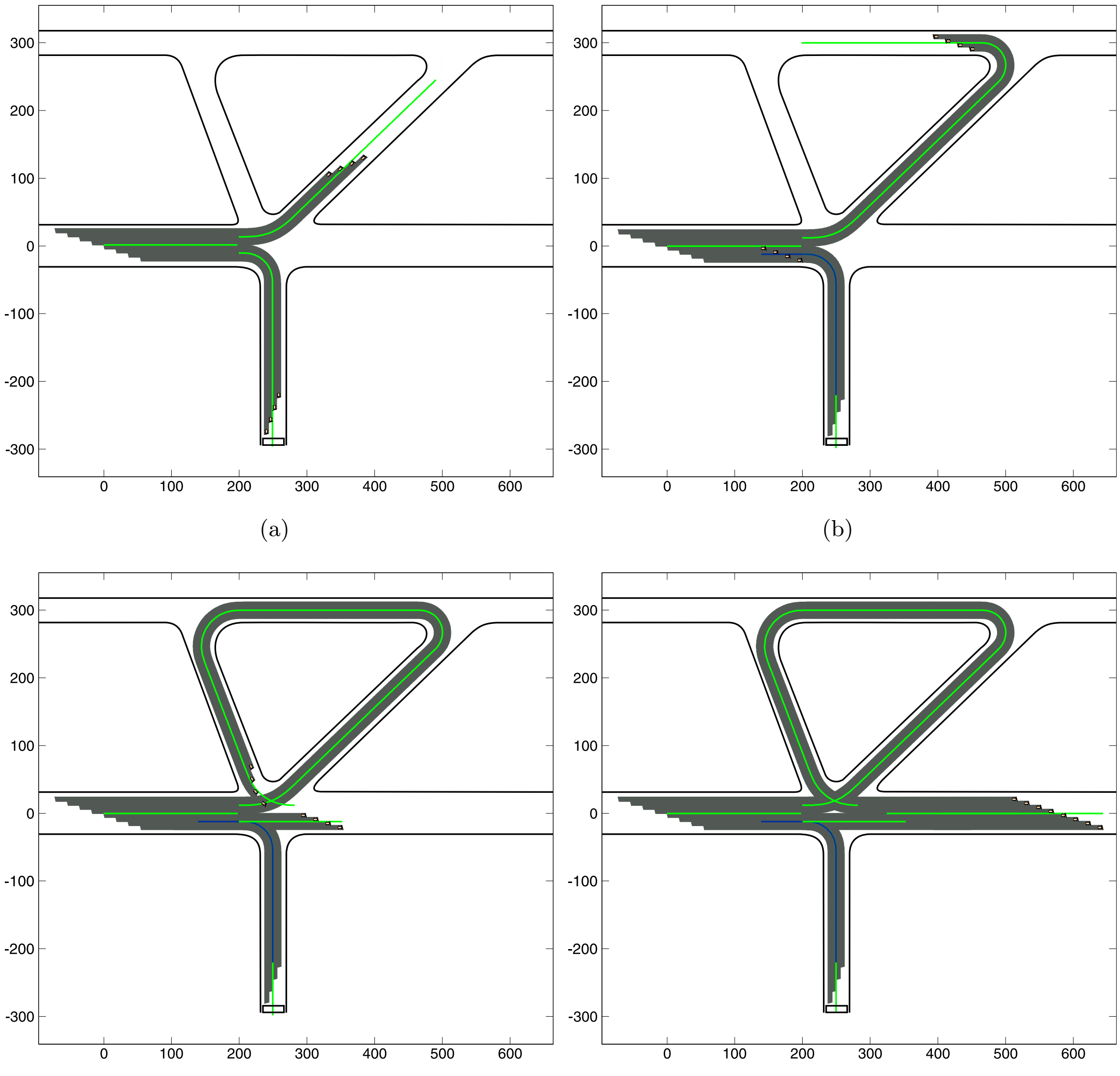
5.5 Implementation and simulation results

In this section we provide some remarks concerning the implementation of the multi-snowplow motion coordination. Furthermore we present the results from the simulation of a motion coordination scenario that covers all crucial situations that might occur during the snow shoveling process. This includes dividing and extending formations as well as reacting on unforeseen obstacles.

For the snow removal the formation should be arranged in such a way that the snow is transported to the side of the roads. Since the Task Allocation module decides about the compositions of the formations, the vehicles should be applicable in every possible setup. As a natural consequence the plows have to be homogeneous, i.e., size, speed and steering bounds, shovel orientation, etc. should be the same for all of them. This induces a formation structure in which the vehicles are arranged diagonal to the roads direction. Since the shovel's mounting point is not in the vehicle's center, the shovel covers the path to a lesser extent during turns than on straight trajectory segments. Therefore we arrange the vehicles in such a way that their shovels overlap a little even during maximal turning. This also leaves some tolerance for temporary displacements due to uncertainties in the localization.

In some cases it can be useful to adjust the shape of a formation while its vehicles are in motion. This can be done, e.g., in order to adapt to changes in the environment like a road getting narrower, to avoid an obstacle, or to adjust the formation structure after switching to a new group. Such deformations can be achieved by appropriate and smooth enough modifications of the formation parameters p_i and q_i as it was described in Chapter 4.

Figure 5.6 shows four snapshots of the implemented snow shoveling simulation. The initial formation whose reference point starts at position $(0, 0)$ consists of 8 plows. The desired speed of the reference point is 10m/s throughout all trajectory segments and for the feedback gains of the trajectory tracking controller we chose $K_{p,j} = 6.25$ and $K_{v,j} = 4.75$ with $j \in$



(a)

(b)

(c)

(d)

Figure 5.6: Simulation snapshots from the introduced motion coordination approach in an artificial scenario that covers all crucial situations that might occur during the snow shoveling process. Units are in meters.

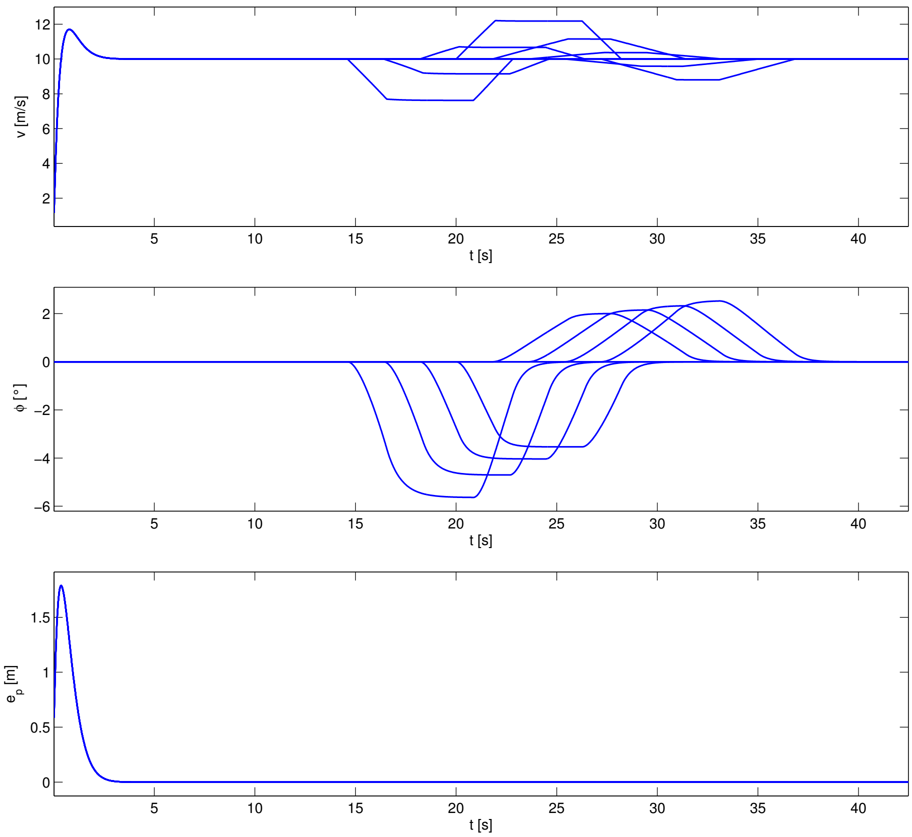


Figure 5.7: Profiles of speed v, steering angle ϕ and position error e_p from the first part of the snow shoveling simulation.

$\{1, 2\}$. On the first intersection the group is divided into two formations in order to sweep smaller roads as it has been commanded by the Task Allocation module. On the motion coordination level this is achieved by letting snowplows, that reach the transition point, switch into another group by changing their q-parameter to an appropriate value and follow the new reference trajectory respectively. In the transition from one into two formations depicted in Figure 5.6(a) q_i, $i \in \{1, \ldots, 8\}$ is changed from $(q_1, ..., q_8) = (-21, -15, -9, -3, 3, 9, 15, 21)$ to $(q_1, ..., q_4) = (q_5, ..., q_8) = (-9, -3, 3, 9)$, where we numbered the vehicles from the front to the rear of the initial formation. It can be seen that the transition to the subsequent reference trajectory happens smoothly.

After some time the formation heading south faces an obstacle blocking the road. At this point the plow that detected the obstacle informs the Command Center. As a result of the Task Allocation module's next iteration the group in the south receives the command to return to the northern intersection and wait for four more vehicles before start sweeping the road heading eastwards. For turning the formation around there is not enough space available. So the formation moves backwards until it reaches the last intersection where an appropriate reorientation is always possible (cf. Figure 5.6(b)). The snapshot in Figure 5.6(c) shows the formation waiting for additional plows at the road to the right of the intersection. At the moment the group that cleaned the road in the north arrives, the two formations unite and start sweeping the east heading road (cf. Figure 5.6(d)). Further simulated scenarios like, e.g., for the environment of Frankfurt International airport can be found in [HSS07a, SHS08a].

Figure 5.7 shows the profiles of speed v, steering angle ϕ and position error $e_p = |(x - x_d, y - y_d)|$ from the first part of the snow shoveling simulation. The depicted profiles end shortly after the point when all vehicles completed their first curve (just before snapshot 5.6(a) was taken). The remaining speed and steering changes during the other curves are principally recurrences with different amplitude and prolongation.

Looking at the position error profile it can be seen that the error initially grows before it converges asymptotically towards zero. This results from the dynamic nature of the controller and the fact that the vehicles start with zero speed. The speed profile depicts that the vehicles' speeds overshoot the desired speed of 10m/s at the beginning, in order to catch up with the constantly moving reference point. To avoid this, one could play with the values for $K_{p,j}$ and $K_{v,j}$ or one could define an adequate desired speed function $v_c(s)$ that also considers the acceleration time of the plows. As can be also seen from the error profile, switching of formations as well as driving through curves, which happens to all vehicles within the first 40s, does not significantly influence the position error. As expected, the changes of speed and steering of vehicles of the same formation, happen during intervals of equal length for the same curve. Note that the speed and steering profiles are continuous, which would not be the case, if the barycenter controller from [DOS98] were applied. This is because for exact tracking it requires a reference trajectory that is at least C^3-continuous with respect to x and y.

5.6 Hardware experiments

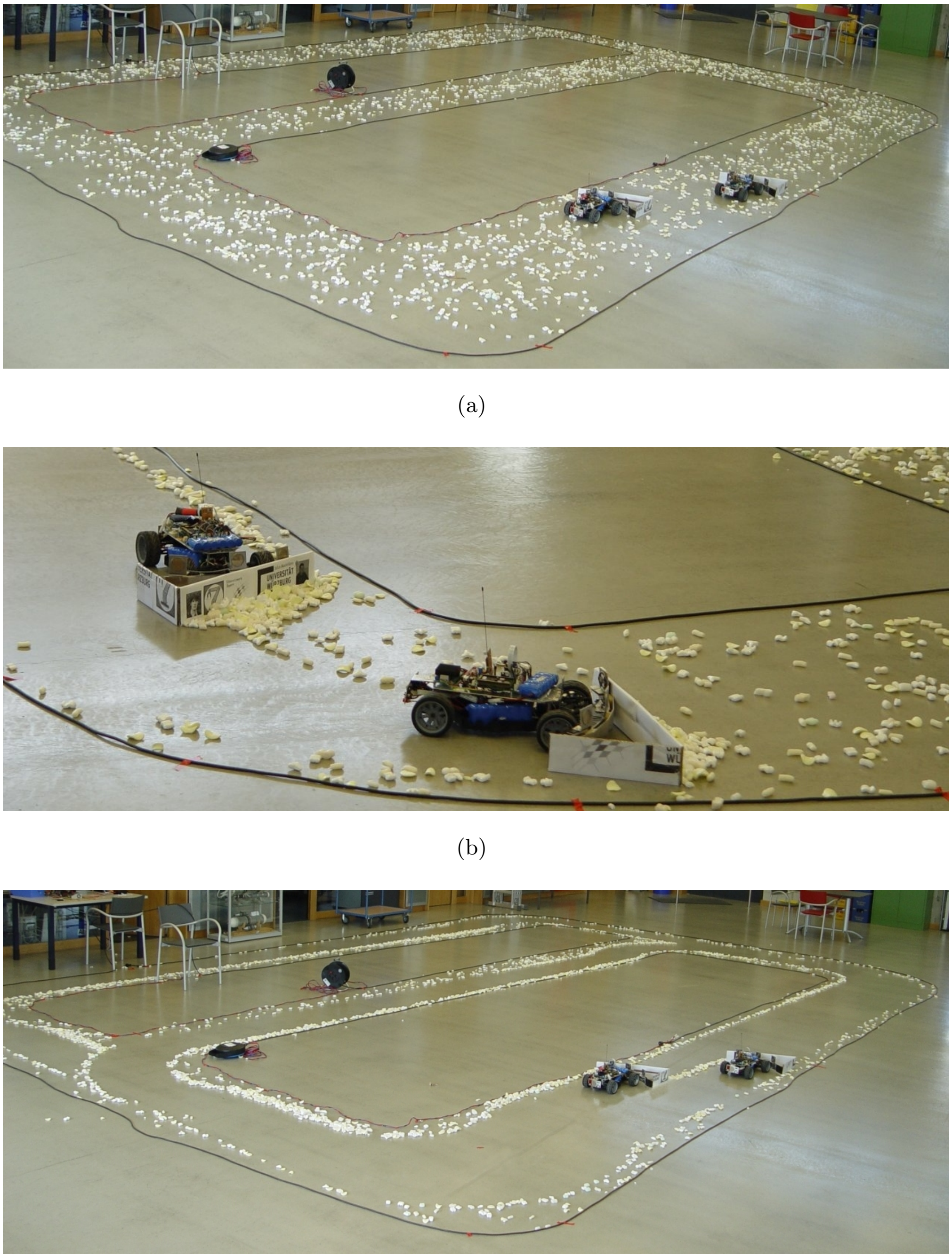
This section describes the snow shoveling hardware experiments that were carried out using a testbed with two car-like mobile robots of the Indoor MERLIN type (cf. Appendix B). The aim of the hardware experiments is to verify the results of the motion coordination simulations and especially the feedback controller (5.5) in a real environment. The utilized MERLIN robots are equipped with various sensors including wheel encoders and a gyroscope. Further, each vehicle contains a wireless communication device to enable inter-robot data-exchange. For the experiments the snow was made of small pieces of polystyrene and for the shovels we used straight bars, which were mounted transversely to the bumper of the mobile robots. The distances between front and rear axle L and between front axle and center of the shovel M were determined as $L = 0.3$m and $M = 0.18$m. The snowshovels' breadthwise coverage when moving straight is approximately 0.3m.

The designed scenario consists of a larger road for two vehicles and two smaller roads that can be swept by only one plow. Since we have just three roads and two intersections, the task allocation for this small scenario is optimally solved in one step. Initially the vehicles relative formation parameters are chosen to be $p_1 = 0$m, $p_2 = -1.5$m, $q_1 = 0.25$m and $q_2 = -0.25$m. We did several experimental runs with different controller gains in the scenario where the reference path is composed of line and circle segments as well as in the scenario where the reference path is composed of line segments and CC-turns. Figure 5.8 shows three snapshots from the experiments.

At the beginning of a run, the two snowplows start as one formation to remove the snow from the bigger road. After shoveling the first turn together the formation is divided into two formations with only one plow in order to clean the smaller roads separately. At the point of separation, we set $p_1 = p_2 = q_1 = q_2 = 0$ and the vehicles start tracking individual trajectories. Before the snowplows reunite, the vehicle on the center road has to wait for the other one to pass by. Otherwise the robot coming from the outer slope would leave an amount of snow in the turnoff. Note that the remaining snow on the roads after the shoveling process results from the simple design of the shovels. When the vehicles unite again to one formation their p and q-parameters are reset to its initial value and the plows converge to their initial arrangement during the final turn. After this, they proceed towards the starting point on the big road.

The localization of the robots in the experiment relies on dead reckoning which is based on information obtained from wheel encoders and gyroscope. As it was mentioned before, an accurate positioning system should be utilized in a real application in order to obtain external feedback for the robots' absolute position. But note that a global positioning system will not significantly change the results of the experiment, since the principle responses of the motion coordination implementation as well as of the tracking controller will be the same. The only difference lies in the fact that a global feedback prevents the accumulation of the global position error, which is crucial for the safety and robustness in real world scenarios.

The speed and the steering profile as well as the vehicles individual position error of two experimental runs are depicted in Figure 5.9. The solid curves denoted by cc_1 and cc_2 refer to the run with clothoid segments while the dashed curves denoted by circ_1 and circ_2 correspond to the run with circular segments during turns. In these runs we set the feedback gains of the trajectory tracking controller to $K_{p,j} = 9.0$ and $K_{v,j} = 6.0$ with $j \in \{1, 2\}$ such that the poles of the closed-loop system are located at -3. The mean speed along the reference trajectory



(a)

(b)

(c)

Figure 5.8: Photographic snapshots from the snow shoveling laboratory hardware experiments. The initial experimental setup is depicted in (a), while (b) shows a closeup of the two robots cleaning the first curve together. Image (c) was taken after a successful run of the experiment.

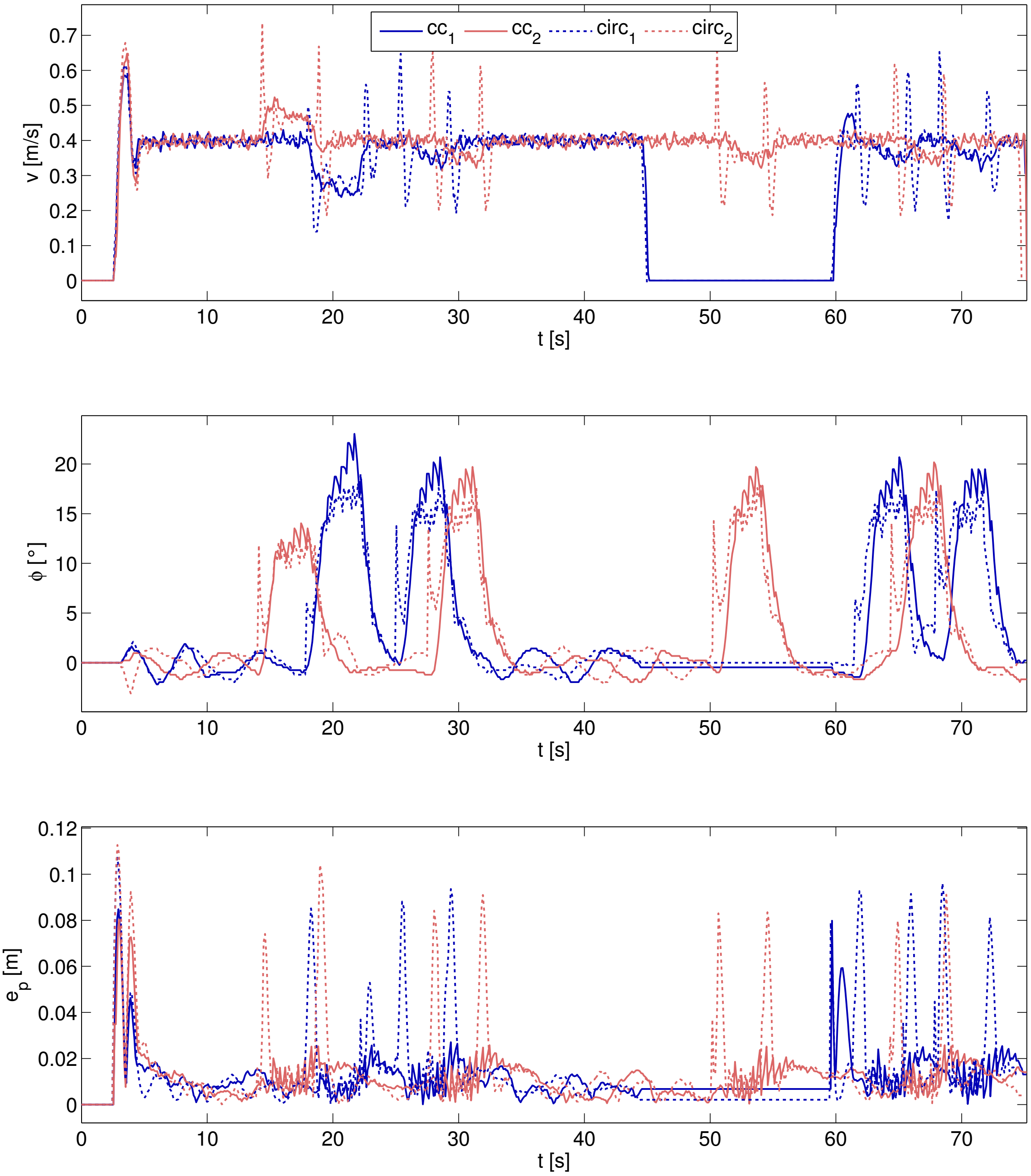


Figure 5.9: Plots from the snow shoveling hardware experiment. The figure contains plots of the individual vehicles' speed v, steering angle ϕ, and position error e_p from two experimental runs. The solid curves denoted by cc_1 and cc_2 refer to the run with clothoid segments during turns while the dashed curves denoted by circ_1 and circ_2 correspond to the run with circular segments.

was set to 0.4m/s. As a compromise between a reasonable resolution for the encoder-based speed measurement and a preferably small discretization stepsize, the sampling interval was set to 105ms. The speed values, which are also utilized by the dead reckoning, are quite accurate. In contrast to this the steering values are only a linear approximation from the PWM signal currently applied to the steering servo. The accuracy of the steering values is further decreased by a tolerance of about 4° in the wheel suspension.

In the position error plot in Figure 5.9 it can be observed that there is a relatively large position error at the beginning of both simulation runs as well as in the profiles of robot 1 at around $t = 60$s, which is directly after the waiting. This comes from the fact that the reference speed is already 0.4m/s from the beginning and in contrast to the kinematic model, the real robot cannot accelerate arbitrarily fast due to its inertia. A reduction of this error without considering a dynamic model could be achieved by adapting the speed profile of the reference trajectory. The chosen gains are quite aggressive for bigger errors as they occur when starting from standstill, like it happens, e.g., at the beginning of the speed profile. But once this initial position error is overcome, the controller keeps the error below 0.027m in the case of a smooth enough reference trajectory. The increase as well as the oscillation of the position error during turns results from the discretization of the continuous model. In the continuous-time simulation presented in Section 5.5 this phenomenon could not be observed.

Due to the step in the derivatives when switching from a line segment to a circular segment we observe short peaks at these connection points in the dashed speed profiles as well as in the dashed profiles of the position error. Further, there is an initial overshoot in the steering angle at the beginning of each turn. These phenomena can be reduced by choosing less aggressive controller gains at the expense of a slower decrease of the tracking error. Moreover, as can be seen in the steering profile in Figure 5.9, the steering angle oscillates with an amplitude of approximately 2° on straight path segments. This results from the tolerance in the wheel suspension of the robots such that in the worst case the controller needs to command a change greater than 4° to achieve a change of steering at all.

5.7 Summary

In this chapter we discussed the application of autonomous multi-snowplow formations for the snow shoveling on airfields. For this we introduced a centrally supervised system architecture with a leader-follower hierarchy on the motion coordination level. The architecture was designed by consideration of the specific properties of the task. To solve the motion coordination part we introduced a trajectory planner that transforms the output of the Task Allocation module into a trajectory for the reference point of the corresponding Formation unit. On the lowest level each vehicle computes its own desired trajectory based on the reference trajectory and applies a feedback controller that was adapted for the needs of the snow shoveling task. Furthermore, the framework is able to react on unforeseen changes in the environment such as narrow road segments, obstacles and blocked roads. The individual approaches as well as the complete framework have been tested in various simulations. Additionally the robustness of the motion coordination solutions was verified in hardware experiments.

6 Rendezvous for groups of car-like mobile robots

This chapter deals with the rendezvous for groups of car-like mobile robots. The rendezvous problem for multi-vehicle systems requires that all nodes are driven to a common configuration by means of local control strategies. Before we survey related work on rendezvous and the interconnected consensus problem, we describe how to model the interaction topology in multi-agent systems by a graph. Then, we formulate the problem statement and consider the rendezvous task in groups of car-like robots with free steering angle. In the main part of this chapter we solve the rendezvous problem for the bounded steering case by applying discontinuous feedback control in combination with a specific switching logic. After presenting results from simulations and hardware experiments, the method is extended for faster convergence, which is verified by a statistical analysis.

6.1 Graph-based representation of the interaction topology in multi-agent systems

In multi-agent systems, the underlying interaction topology can be represented by a *graph* $G = (\mathcal{V}, E)$. Given a group of N agents, each element of the *set of vertices* $\mathcal{V} = \{1, \ldots, N\}$ represents one agent of the system. The *set of directed edges* E, often referred to as arcs, consists of ordered pairs of vertices $(i, j) \in \mathcal{V} \times \mathcal{V}$ which indicate an interaction link from agent i to agent j. Such a link enables agent i to obtain knowledge of the current state of agent j. This might be achieved by active sensing or via local communication as depicted in Figure 1.2.

For agent i the *set of neighbors* $\mathcal{N}_i$ consists of all agents $j \in \mathcal{V}, j \neq i$, that are connected with agent i via an interaction link, i.e., $(i, j) \in E$ and $|\mathcal{N}_i|$ denotes the cardinality of the set $\mathcal{N}_i$. If $(i, j) \in E \Rightarrow (j, i) \in E$ holds for all $(i, j) \in E$ we refer to the graph G as *undirected.* If this is not the case G is a *directed* graph also referred to as *digraph.* A sequence of vertices is called a *path*, if for each of its vertices there exists an edge to the next vertex in the sequence. An undirected (directed) graph is *connected* (*strongly connected*) if for all pairs of vertices

$(i,j) \in \mathcal{V} \times \mathcal{V}$ there exists a path from i to j. If the underlying undirected graph of a digraph G is connected, we call G *weakly connected.* Furthermore, a digraph G is called *balanced*, if the number of incoming edges of each vertex of G is equal to its number of outgoing edges.

In a *proximity graph* the existence of an edge between two distinct agents depends on the geometric distance of agents in the configuration space. Formally, if for the utilized distance function $d(i,j), i \neq j$, it holds that $d(i,j) \leq \hat{d}$ for a predefined constant $\hat{d} > 0$, if and only if $(i,j) \in E$. Note that a proximity graph is always undirected, since distance functions are symmetric by definition.

6.2 Problem statement

Consider a system consisting of N homogeneous car-like robots. The motion of the ith robot in the plane is described by the following kinematic equations:

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i, \\ \dot{y}_i &= v_i \sin\theta_i, \\ \dot{\theta}_i &= v_i \frac{\tan\phi_i}{L}, \end{aligned} \tag{6.1}$$

where $v_i \in \mathbb{R}$ is the velocity input and ϕ_i is the steering input, which is limited to $(-\pi/2, \pi/2)$ in the free steering case. Like in model (3.8) in Section 3.1, L denotes the distance between front and rear axle of a vehicle. The vector $p_i = (x_i, y_i)^T$ represents the position in the plane, and θ_i is the orientation angle with respect to the x-axis. We can rewrite (6.1) for the complete system as

$$\begin{aligned} \dot{p} &= f_1(p,\theta,v) = \begin{pmatrix} v_1 \cos\theta_1 \\ v_1 \sin\theta_1 \\ \vdots \\ v_N \cos\theta_N \\ v_N \sin\theta_N \end{pmatrix}, \\ \dot{\theta} &= f_2(p,\theta,v,\phi) = \begin{pmatrix} v_1 \tan\phi_1/L \\ \vdots \\ v_N \tan\phi_N/L \end{pmatrix}, \end{aligned} \tag{6.2}$$

with $p = (x_1, y_1, \ldots, x_N, y_N)^T$, $\theta = (\theta_1, \ldots, \theta_N)^T$, $v = (v_1, \ldots, v_N)^T$, and $\phi = (\phi_1, \ldots, \phi_N)^T$. Further, we assume that each vehicle is labeled with a unique ID from 1 to N and we denote by ID_i the ID of the ith robot.

The problem is now to find local continuous-time control laws that drive the robots' position vectors p_i to a common configuration. During the execution of the control law, the robots have access to the state variables of their neighbors in the interaction graph, which is assumed to be undirected, connected, and static. Further, we assume that the vehicles have no spatial extent and therefore do not collide with each other.

6.3 Related work

The rendezvous problem in networked multi-agent systems can be seen as a specific application of the consensus problem. Due to the close relation we review the key results and related work from consensus theory before we survey the work that explicitly deals with the rendezvous of robotic agents.

6.3.1 Consensus in networked multi-agent systems

Reaching a consensus in networks of agents or dynamic systems means to reach an agreement regarding a certain quantity of interest. A *consensus protocol* also referred to as *consensus algorithm* is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network [OSFM07]. Since the consensus protocol is implemented locally on each agent, the consensus problem is related to the field of distributed computation (cf. [Lyn96]). Also in statistics, consensus problems have a long history (cf. [DeG74]), while from a system theory point of view they are closely related to the concept of connective stability (cf. [Šil78, Chapter 4]). Later, the problem has been approached in the context of distributed computation over networks like in the work of Borkar and Varaiya on asymptotic agreement in distributed estimation [BV82] and in the work of Tsitsiklis et. al. on asynchronous distributed computation [Tsi84, TBA86]. During the last years this issue has received a lot of attention in the cooperative control community. For a better understanding of the consensus problem in networked multi-agent systems we review recent key results before we summarize the most important works concerning this issue.

Let $x_i(t),\ i \in \{1, \ldots, N\}$ be the information state of agent i at time t, which represents the information the agents should agree on. This could be, e.g., its position, heading, speed, an environmental measurement, or a decision variable. Reaching a consensus means that under a certain control law (consensus protocol) the agents information state asymptotically converges to a space for which

$$x_1 = x_2 = \ldots = x_N,$$

which means $||x_i - x_j|| \rightarrow 0$ as $t \rightarrow \infty, \forall i \neq j$. As it was described in Section 6.1 the interaction topology of the group of agents can be projected on a graph $G = (\mathcal{V}, E)$. If we assume the information state of all agents follows simple integrator dynamics in continuous time, i.e.,

$$\dot{x}_i(t) = u_i(t), \tag{6.3}$$

a consensus protocol can be expressed as

$$\dot{u}_i(t) = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(x_j(t) - x_i(t)), \tag{6.4}$$

where a_{ij} is a positive, possibly time-varying weighting factor for the edge $(i, j) \in E$ and

$$\mathcal{N}_i(t) = \{j \in \mathcal{V} : a_{ij} \neq 0\}$$

denotes the set of neighbors of agent i, which includes all agents whose information is available for agent i at time t (cf., e.g., [RB05, OSFM07]). In matrix form the closed-loop system of (6.3) and the linear consensus protocol (6.4) can be written as

$$\dot{x}(t) = -Lx(t),$$

where $x = (x_1, \ldots, x_N)^T$ and $L = (l_{ij})_{i,j \in \{1,\ldots,N\}}$ is the *graph Laplacian* defined by

$$l_{ij} = \begin{cases} \sum_{k=1,k\neq i}^{N} a_{ik} & j = i \\ -a_{ij} & j \neq i. \end{cases}$$

The corresponding formulation for agents with discrete-time models can be found, e.g., in [RB05, OSFM07].

The following theorem about asymptotic stability and the group decision of (6.4) has been stated in [OSFM07].

Theorem 6.3.1. *Consider a network of N integrator-agents with topology G applying consensus protocol* (6.4)*. Suppose G is a strongly connected digraph. Then a consensus is asymptotically reached for all initial states and the group decision is*

$$\lim_{t\to\infty} x_i(t) = \sum_{i=1}^{N} w_i x_i(0) \quad \forall i \in \{1, \ldots, N\},$$

with $\sum_{i=1}^{N} w_i = 1$. Further, if the digraph is balanced, an average-consensus is asymptotically reached, i.e.,

$$\lim_{t\to\infty} x_i(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(0) \quad \forall i \in \{1, \ldots, N\}.$$

Proof. See the proof of Theorem 1 in [OSFM07]. □

If the communication graph is only weakly connected or not connected at all, it is most likely that a consensus will not be reached by application of (6.4).

The speed of reaching a consensus in a system with integrator dynamics depends on the topology of the network. Considering the application of a consensus algorithm in balanced directed networks that include undirected networks as a special case, it can be easily shown that $\alpha = (\sum_{i=1}^{N} x_i(t))/N$ is an invariant quantity. Thus, we can define the *disagreement vector* $\delta(t) = x(t) - \alpha(1, \ldots, 1)^T$ that provides a measure for the disagreement within the network [OSM04]. With respect to the dynamics of the disagreement vector, the following theorem provides an upper bound on the convergence speed for linear continuous-time consensus in balanced networks.

Theorem 6.3.2. *Let $\lambda_2(L_s)$ be the second smallest Eigenvalue of the matrix $L_s = (L + L^T)/2$. For a strongly connected and balanced directed network, a continuous-time consensus is globally exponentially reached with a speed that is faster or equal to $\lambda_2 = \lambda_2(L_s)$, i.e., $||\delta(t)|| \leq ||\delta(0)|| \exp(-\lambda_2 t)$.*

Proof. See the proof of Theorem 8 in [OSM04]. □

Note that the value of λ_2, which is also referred to as algebraic connectivity of the graph G, is relatively large for dense graphs and relatively small for sparse graphs. According to Theorem 6.3.2 that means a network with many interconnections solves a consensus problem faster than a connected but sparse network. In this context Olfati-Saber demonstrated in [OS05] that the algebraic connectivity of a regular network can be considerably increased via random rewiring that turns local links into nonlocal ones. The results of [CFSZ08] show significant improvement of the convergence rate under weighted digraphs, when instead of being fixed, the communication graph is chosen randomly at each time step over a family of graphs with the constraint that the number of incoming arcs in each vertex is constant.

Let us suppose that agent i receives the information state sent by its neighbor j after a constant time-delay τ_{ij}, which can be seen as a network with a uniform one-hop communication time-delay. In [OSM04] the authors proposed the following consensus protocol that solves the average-consensus problem for such systems on undirected graphs.

Theorem 6.3.3. *Consider a network of integrator agents with equal communication time-delay $\tau > 0$ in all links. Assume the network topology G is static, undirected, and connected. Then, the consensus protocol*

$$u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij} \big(x_j(t - \tau_{ij}) - x_i(t - \tau_{ij}) \big)$$

with $\tau_{ij} = \tau$ globally asymptotically solves the average-consensus problem if and only if $0 \leq \tau < \pi/2 \cdot \lambda_N$*, where $\lambda_N = \lambda_{\max}(L)$ is the largest Eigenvalue of the graph Laplacian L of graph G.*

Proof. See the proof of Theorem 10 in [OSM04]. □

In many applications the communication topology changes over time, e.g., due to link changes, packet loss, formation reconfiguration, etc. This gives rise to the following theorem, which states that a consensus is asymptotically reached, when the communication topology switches between balanced and strongly connected digraphs $G_1, \ldots, G_m$.

Theorem 6.3.4. *Consider a network of agents applying the consensus protocol*

$$\dot{x} = -L(G_{s(t)})x,$$

where $s(t) : \mathbb{R} \to \{1, \ldots, m\}$ is an arbitrary switching signal. Suppose every graph $G_1, \ldots, G_m$ is a balanced digraph that is strongly connected and let

$$\lambda_2^* = \min_{k \in \{1,\ldots,m\}} \lambda_2(G_k).$$

Then, the agents asymptotically reach an average-consensus for all initial states with a speed faster or equal to λ_2^*.

Proof. See the proof of Theorem 9 in [OSM04]. □

Another result that does not require the individual undirected graphs to be connected has been presented in [JLM03, Theorem 2] for the discrete-time domain. The authors show that it is sufficient that there exists an infinite sequence of successive, nonempty, and bounded time-intervals during which the union of the individual communication graphs is connected. This result was further generalized to directed graphs in [Mor05] and [RB05].

Recent surveys on the subject have been published in [OSFM07, RBA05]. As can be seen, many publications about rendezvous and consensus in networked multi-agent systems follow the direction of the results mentioned above. [Ren07] introduces extended consensus algorithms for systems modeled by second-order dynamics (double-integrator). Further, the article points out that many existing leader-follower, behavioral, and virtual structure/virtual leader formation control approaches (cf. Section 4.1) can be mapped on consensus-based strategies. In [RMC07] the authors present a generalized consensus protocol for l-integrator systems ($l \in \mathbb{N}$), and [Ren08] analyzes consensus algorithms in networks of double-integrator agents with bounded control input as well as without relative velocity measurements.

[XW07] presents necessary and sufficient conditions to solve a consensus problem for multi-agent systems in high-dimensional state space. Consensus in a network of discrete-time integrator agents under quantized communication has been discussed in [Spe06, Chapter 4], while quantized consensus on connected graphs in discrete-time for which the information state of each node is always an integer-value has been studied in [KBS07]. Consensus protocols for reaching an agreement in finite time in networks of continuous-time integrator agents were introduced in [WX07]. Moreover the authors consider switching communication topologies and establish an upper bound for the overall convergence time. Linear consensus strategies in randomly changing communication networks have been studied, e.g., in [PS07] and [FZ08], while reaching consensus via model predictive control strategies has been addressed in [BGP06, JSJJ08, KJ08]. Consensus protocols especially designed for the application in wireless sensor networks recently have been analyzed in [KM08] and [BMM08].

A novel asynchronous consensus control strategy for continuous-time multi-agent systems with switching topology and time-varying delays was proposed in [XW08]. Due to its asynchronism, the approach overcomes the difficulties caused by unreliable communication channels that usually cannot be avoided in current state-of-the-art wireless networks.

6.3.2 Rendezvous for groups of mobile robots

The rendezvous problem can be seen as a special case of the consensus problem, as the goal is to develop local control strategies that drive each robots' positions to a common value in the configuration space. Even though in some works the point of rendezvous is predefined [BF08] or has to be within a finite set of candidate rendezvous points [FMC08], we focus on decentralized control strategies where this is not the case.

Local control strategies for the rendezvous in groups of mobile autonomous agents have been introduced in [LBF04]. Graph-based nonlinear feedback control laws for a group of integrator agents based on weighted graph Laplacians have been proposed in [JE07]. Further, the authors introduce a control law that solves the rendezvous problem while ensuring connectedness, meaning the agents move in a way such that established communication links in the proximity graph will never break down.

The rendezvous problem for groups of differential drive robots was investigated in [DK07]. The authors use Lyapunov's direct method and LaSalle's invariance principle that were extended to nonsmooth systems in [SP94] in order to prove asymptotic convergence of the proposed distributed and discontinuous control law to a common configuration in the state space. Further, the control law ensures connectedness of the existing edges in the communication graph. In special cases, rendezvous is not achieved by the proposed control law as it is shown in Chapter C of the appendix.

In [RCB$^+$08] the results from various rendezvous and axial alignment hardware experiments are presented. With four differential drive robots the authors verify the theoretical results for the single integrator consensus protocol (6.4) applied in scenarios with fixed and switching interaction topologies. They also emphasize the importance of a high update rate for the robot positioning, which was achieved by a ceiling-mounted camera.

Cyclic pursuit, which is a special case of rendezvous control law in which agent i pursues agent $(i+1) \operatorname{modulo} N$, has been the subject of research in recent publications. [MBF04] for example presents stability results for cyclic pursuit of integrator agents as well as for unicycle vehicles. In [SG06] the authors determine the point of convergence and define a set of reachable rendezvous points depending on the individual controller gains of the agents with integrator dynamics. Cyclic pursuit for agents with nonlinear equations of motion has been studied in [SG07].

An alternative definition of rendezvous control laws can be found in [MBCF07]. At successive communication rounds, the agents exchange their position with their neighbors. Rendezvous without connectivity constraint is achieved by letting the agents move between communication rounds such that they reach the point that is the average of the positions received from their neighbors. Under a second law originally introduced in [AOSY99], the agents move towards the circumcenter of their neighbors positions and its own, thus ensuring connectivity with each of their neighbors. The circumcenter of a set of points is defined as the center of the smallest circle that encloses all of the points. This law was further investigated in [LMA03, LMA04, CMB06, CP07] for the synchronous and the asynchronous case, where decisions do not depend on a common clock. The main contribution of [MBCF07] in terms of the rendezvous problem is the determination of upper and lower bounds for the time complexity of the two laws. A generalized analyzis of these “stop-and-go” strategies can be found in [LMA07a] and [LMA07b] as well as in [BCM08].

There exist several publications on the rendezvous problem that are less related to our work. For example they deal with rendezvous of spacecraft [WH07], which usually involves docking with previous attitude alignment of the spacecraft. [BF08] investigates minimum-time rendezvous of Dubins vehicles based on path planning. A Dubins vehicle can be considered as a simplified model of a mobile robot with bounded curvature and constant speed [Dub57]. Rendezvous of Dubins cars with minimalist sensing and control was studied in [YLL08]. With Lyapunov tools the authors showed that the vehicles can achieve guaranteed rendezvous by exclusive usage of a three level quantized control and a simple sensor that only reports the presence of another agent within some sector of its windshield. The performance of discrete-time integrator rendezvous under random time-varying communication topologies with respect to average relative agents' distance, input energy consumption and the number of communication messages has been discussed in [SZ06b, SZ06a]. In [NB06] the distributed computation of minimal enclosing shapes for achieving minimum time rendezvous of the robotic agents was investigated. In [SBF07] the analogy between Euclidean curve shortening and the rendezvous problem for mobile autonomous robots is shown on a group with a polygonal interaction topology.

The main area of application for mobile robot rendezvous is field and outdoor robotics. Although mobile outdoor robots are typically of the car-like type, nobody tackled the rendezvous problem for a group of car-like vehicles. In the following sections we will consider the rendezvous problem for these kind of vehicles in order to fill the gap. We will introduce feedback control systems for the free steering case as well as for the more complicated case when the vehicles' steering is bounded. Besides mathematical analysis of the system's convergence we also verify the developed algorithms in various simulation and hardware experiments.

6.4 Rendezvous with Free Steering Angle

In this section we introduce a control law for system (6.1) which drives the robots' position vectors p_i to a common configuration. Note that this leaves out the orientation which means that we do not claim to achieve an agreement on the full state. During the rendezvous process each vehicle takes into account only their own states and those of their neighbors in the underlying interaction graph. Thus the control law can be considered as decentralized.

For the ith robot with kinematics given by (6.1), we define the following discontinuous feedback control law:

$$v_i = -c_1 \operatorname{sgn}(\hat{v}_i) \cdot \sqrt{\left(\frac{1}{2|\mathcal{N}_i|}\hat{x}_i\right)^2 + \left(\frac{1}{2|\mathcal{N}_i|}\hat{y}_i\right)^2}, \tag{6.5}$$

$$\phi_i = \arctan\left(c_2 \omega_i \frac{L}{v_i}\right), \tag{6.6}$$

where

$$\hat{x}_i = \sum_{j\in\mathcal{N}_i}(x_i - x_j), \qquad \hat{y}_i = \sum_{j\in\mathcal{N}_i}(y_i - y_j),$$

$$\hat{v}_i = \cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i,$$

$$\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{otherwise} \end{cases},$$

$$\omega_i = -\frac{\pi}{2} + \left(\operatorname{atan2}\left(-\hat{y}_i, -\hat{x}_i\right) - \theta_i + \frac{\pi}{2}\right) \textbf{ mod } \pi.$$

c_1 and c_2 are positive constants and the function $\operatorname{atan2}(\cdot, \cdot)$ denotes the arctangent of two variables as it was defined in (3.12). It can be seen that for $v_i = 0$ the value of ϕ_i is not defined. But since in this case the system state of robot i does not change, we assume ϕ_i remains unchanged whenever $v_i = 0$.

In the following theorem we prove the partial asymptotic stability of the closed-loop system defined by equations (6.1), (6.5), and (6.6) under some sufficient conditions.

Theorem 6.4.1. *Consider a connected system comprised of N robots that evolve according to* (6.1) *and the control law defined by* (6.5)-(6.6)*. We assume that the constants c_1 and c_2 are chosen according to the following condition:*

$$c_2 > \frac{2}{\pi} c_1. \tag{6.7}$$

Further we assume that $v_i \neq 0$ for vehicles located at unstable weak equilibrium configurations. Then the closed-loop system is asymptotically stable to an equilibrium characterized by

$$p_i - p_j = 0, \quad \forall i, j \tag{6.8}$$

uniformly in θ.

Proof. Consider the Lyapunov function candidate

$$V(p) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left((x_i - x_j)^2 + (y_i - y_j)^2 \right). \tag{6.9}$$

Computing the time-derivative of (6.9) yields

$$\begin{aligned}
\dot{V}(p) &= \nabla V^T f_1(p, \theta) \\
&= \begin{bmatrix} 2\sum_{j \in \mathcal{N}_1} (x_1 - x_j) \\ 2\sum_{j \in \mathcal{N}_1} (y_1 - y_j) \\ \vdots \\ 2\sum_{j \in \mathcal{N}_N} (x_n - x_j) \\ 2\sum_{j \in \mathcal{N}_N} (y_n - y_j) \end{bmatrix}^T \begin{bmatrix} v_1 \cos\theta_1 \\ v_1 \sin\theta_1 \\ \vdots \\ v_N \cos\theta_N \\ v_N \sin\theta_N \end{bmatrix} \\
&= 2 \sum_{i=1}^{N} \left(v_i \left(\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i \right) \right) \\
&= 2 \sum_{i=1}^{N} \left(v_i \hat{v}_i \right).
\end{aligned} \tag{6.10}$$

By substituting the control law (6.5) for v_i in equation (6.10) and considering that $\operatorname{sgn}(z) z = |z|$ we get

$$\dot{V}(p) = -2c_1 \sum_{i=1}^{N} \left(|\hat{v}_i| \cdot \sqrt{\left(\frac{1}{2|\mathcal{N}_i|} \hat{x}_i \right)^2 + \left(\frac{1}{2|\mathcal{N}_i|} \hat{y}_i \right)^2} \right) \leq 0. \tag{6.11}$$

Since $\dot{V}(p)$ is continuous, the set-valued derivative $\dot{\overline{V}}_{K[f_1]}(p,\theta)$ is the singleton $\{\dot{V}(p)\}$ for all possible system configurations. This yields $V\big(p(t)\big) \leq V\big(p(0)\big)$ for all $t \geq 0$, and due to Theorem 2.4.3 the closed-loop system is Lyapunov stable to an equilibrium characterized by $p_i - p_j = 0$ for all i,j uniformly in θ.

Now we define the following sublevel set of relative vehicle positions

$$S_k = \{p : V(p) \leq k\}.$$

For any $k > 0$ the set S_k is closed by the continuity of V. Furthermore it is bounded with respect to the robots' relative positions because for every pair $(i,j) \in E$ we have $|p_i - p_j| \leq \sqrt{k}$ from the definition of V and since the interaction graph is connected this yields

$$|p_i - p_j| \leq (n-1)\sqrt{k} \quad \forall i,j. \tag{6.12}$$

Therefore S_k is compact and we can apply Theorem 2.4.4. It states that all semi-trajectories $p(t)$ starting in S_k will converge to the largest invariant set within

$$\begin{aligned} S &= \{p \in S_k : 0 = \dot{V}(p)\} \\ &= \left\{ p \in S_k : \sum_{i=1}^{N} |\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i| = 0 \right\}. \end{aligned} \tag{6.13}$$

From the set predicate of S, which also can be written as

$$\forall i \in \{1,\dots,N\} : \hat{x}_i = \hat{y}_i = 0 \vee \theta_i = k\pi - \text{atan2}(\hat{x}_i, \hat{y}_i), \qquad k \in \mathbb{Z}\backslash\{0\},$$

it is easy to see that the set S contains the desired equilibrium characterized by (6.8) as well as all configurations for which the direction vector $[\cos\theta_i, \sin\theta_i]^T$ of each vehicle i is oriented perpendicular to the vector defined by $[\hat{x}_i, \hat{y}_i]^T$. We will show now that the derivative of the inner product of the two vectors is not equal to zero for at least one car if the system is not at the desired equilibrium. From this it follows that the largest invariant set in S contains only the desired configurations where $p_i - p_j = 0 \;\; \forall i,j$. The derivative of the inner product yields

$$\dot{\hat{v}}_i = \sin\theta_i \left(\sum_{j\in\mathcal{N}_i} (\dot{y}_i - \dot{y}_j) - \dot{\theta}_i \hat{x}_i \right) + \cos\theta_i \left(\sum_{j\in\mathcal{N}_i} (\dot{x}_i - \dot{x}_j) + \dot{\theta}_i \hat{y}_i \right).$$

W.l.o.g. we set $i \in \arg\max_j |v_j|$ and assume $\theta_i = 0$. As a result the derivative of $\hat{v}_i$ becomes

$$\dot{\hat{v}}_i = \underbrace{\sum_{j\in\mathcal{N}_i} (\dot{x}_i - \dot{x}_j)}_{=:a} + \underbrace{\dot{\theta}_i \hat{y}_i}_{=:b}. \tag{6.14}$$

Note that for $V > 0$ the elements of S correspond to points on the discontinuous surface of the closed-loop system. From the control law (6.5) it can be seen that in terms of Filippov solutions for the velocity of vehicle i it holds

$$v_i \in \left[-\frac{c_1}{2|\mathcal{N}_i|}|\hat{y}_i|, \; \frac{c_1}{2|\mathcal{N}_i|}|\hat{y}_i| \right]. \tag{6.15}$$

It is easy to see that since $v_i = 0$ is included in the set, these configurations are weak equilibria of the system. Looking at the neighborhood of these points it is obvious that these equilibria are also unstable and any small perturbation would drive the system out. Remember that by assumption we have $|v_i| > 0$. As such, for the summands of (6.14) we obtain

$$\begin{aligned} |a| &= \left|\sum_{j\in\mathcal{N}_i}(\dot{x}_i - \dot{x}_j)\right| = \left|\sum_{j\in\mathcal{N}_i}(\underbrace{\cos\theta_i}_{=1} v_i - \cos\theta_j v_j)\right| \\ &\le 2|\mathcal{N}_i|\cdot|v_i| \overset{(6.15)}{\le} c_1|\hat{y}_i| \\ &\overset{(6.7)}{<} \frac{\pi}{2}c_2|\hat{y}_i| = c_2|\omega_i|\cdot|\hat{y}_i| = |c_2\omega_i\hat{y}_i| = |\dot{\theta}_i\hat{y}_i| = |b|. \end{aligned} \tag{6.16}$$

For the rearrangement we used the fact that according to equation (3.12) $\operatorname{atan2}(y, 0) = (\pi/2)\operatorname{sgn}(y)$ and therefore it holds that $|\omega_i| = \pi/2$ as well as $\operatorname{sgn}(\hat{y}_i)\omega_i = -|\omega_i|$ from the definition of ω_i. Now from $|a| < |b|$ we conclude that $\dot{v}_i = a + b \neq 0$. This means that the set S without the desired equilibrium (6.8) is non-invariant under the closed-loop system defined by (6.1), (6.5), and (6.6) and the partial trajectory $p(t)$ converges to the set characterized by (6.8). □

Note that Theorem 6.4.1 does not claim anything about the absolute position to which all the agents converge; however, it says that the agents will converge to some common point. Theoretically, also the solutions for multiple runs with the same initial conditions are not necessarily the same, since it is not clear if vehicles will move forward or backward when the system starts from a point on the discontinuous surface. Further note that each local part of the control law can be interpreted as a point stabilizing controller with a moving reference point. If we look at a point of time, the rendezvous control law of vehicle i is equivalent to the position stabilizing controller in Subsection 3.2.1 with the desired position

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \frac{1}{2|\mathcal{N}_i|}\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} = \frac{1}{2}\begin{pmatrix} x_i + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} x_j \\ y_i + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} y_j \end{pmatrix}.$$

6.5 Rendezvous with Bounded Steering Angle

In this section the control law (6.5)-(6.6) is enhanced for systems of car-like robots with limited steering angle. The kinematics of vehicle i are as in (6.1) except that for the steering input we have

$$\phi_i \in [-\phi_{\max}, \phi_{\max}], \tag{6.17}$$

where $0 < \phi_{\max} < \pi/2$.

6.5.1 The closed-loop feedback control law

For each robot i we define the following discontinuous feedback control law:

$$v_i = \begin{cases} -c_1\left(\hat{v}_i + \operatorname{sgn}(\hat{v}_i)\cdot\hat{b}_i\right) & \hat{e}_i > \frac{L}{\tan\phi_{\max}} \\ -c_1\hat{v}_i & \text{otherwise} \end{cases}, \tag{6.18}$$

$$\phi_i = \sigma_{\phi_{\max}}\left(\arctan\left(c_2\omega_i\frac{L}{v_i}\right)\right), \tag{6.19}$$

where $\hat{x}_i, \hat{y}_i, \hat{v}_i, \omega_i, c_1, c_2$ and $\operatorname{sgn}(\cdot)$ are defined as in Section 6.4, and

$$\hat{e}_i = \sqrt{\left(\frac{1}{2|\mathcal{N}_i|}\hat{x}_i\right)^2 + \left(\frac{1}{2|\mathcal{N}_i|}\hat{y}_i\right)^2}, \tag{6.20}$$

$$\hat{b}_i = \hat{e}_i - \frac{L}{\tan\phi_{\max}},$$

$$\sigma_b(x) = \begin{cases} b & x > b \\ -b & x < -b \\ x & \text{otherwise} \end{cases}.$$

Like in the free steering case we assume that the value of ϕ_i remains unchanged whenever $v_i = 0$. Stability and convergence properties of the control law defined by (6.18) and (6.19), which we also refer to as the closed-loop controller, are given in the two following lemmas.

Lemma 6.5.1. *Consider a connected system of N robots that evolve according to* (6.1) *with constraint* (6.17) *and the control law* (6.18)-(6.19). *Then the closed-loop system is Lyapunov stable to an equilibrium characterized by $p_i - p_j = 0$ for all i, j uniformly in θ.*

Proof. To show stability in the sense of Lyapunov we use equation (6.9) as a candidate Lyapunov function like in Section 6.4 and compute

$$\dot{V}(p) = 2\sum_{i=1}^{N}(v_i\hat{v}_i).$$

Considering again $\operatorname{sgn}(z)z = |z|$ and substituting v_i by control law (6.18), yields

$$\begin{aligned}\dot{V}(p) &= \begin{cases} -2c_1\sum_{i=1}^{N}\left(\hat{v}_i^2 + |\hat{v}_i|\hat{b}_i\right) & \hat{e}_i > \frac{L}{\tan\phi_{\max}} \\ -2c_1\sum_{i=1}^{N}\hat{v}_i^2 & \text{otherwise} \end{cases} \\ &\leq 0.\end{aligned} \tag{6.21}$$

As a result $V\big(p(t)\big) \leq V\big(p(0)\big)$ for all $t \geq 0$, and due to Theorem 2.4.3 the closed-loop system is Lyapunov stable to an equilibrium characterized by $p_i - p_j = 0$ for all i, j uniformly in θ. □

Lemma 6.5.2. *Consider a connected system of N car-like robots that evolve according to* (6.1) *with constraint* (6.17). *We assume that the constants c_1 and c_2 are chosen to satisfy*

(6.7). *Further we assume that* $v_i \neq 0$ *for vehicles located at unstable weak equilibrium configurations. Then applying the control law* (6.18)-(6.19) *for all vehicles drives the system's position variables towards the set*

$$\begin{aligned} S_{\text{inv}} &= \left\{ p : \dot{V}(p) = 0 \wedge \hat{e}_j \leq L/\tan\phi_{\max}\ \forall j \in \{1, \ldots, N\} \right\} \\ &= \left\{ p : \sum_{i=1}^{N} |\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i| = 0 \wedge \hat{e}_j \leq L/\tan\phi_{\max}\ \forall j \in \{1, \ldots, N\} \right\}. \end{aligned}$$

Proof. Like in the proof of Theorem 6.4.1 we define the sublevel set

$$S_k = \{p : V(p) \leq k\},$$

which for any $k > 0$ is closed by the continuity of V, and also bounded with respect to the robots relative positions. Further we have $\dot{V}(p) \leq 0$ for all $p \in S_k$ (cf. Lemma 6.5.1). Now we can apply the modified version of LaSalle's invariance principle stated in Theorem 2.4.4. As a result, all semi-trajectories $p(t)$ starting in S_k will converge to the largest invariant set within

$$\begin{aligned} S &= \{p \in S_k : \dot{V}(p) = 0\} \\ &= \left\{ p \in S_k : \sum_{i=1}^{N} |\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i| = 0 \right\}. \end{aligned}$$

To classify the non-invariant subsets of S, we let the current configuration be an element of the set S where $V > 0$. Further we assume that $\hat{e}_j > L/\tan\phi_{\max}$ for at least one j, and the constants c_1 and c_2 are chosen appropriately by considering (6.7). Now, w.l.o.g. we set the orientation $\theta_i = 0$ where $i \in \arg\max_j |v_j|$ is chosen to be among the fastest vehicles of the group. Since the set S is the same as in the free steering case the derivative in time of $\hat{v}_i$ is equal to (6.14). Again the elements of S correspond to points on the discontinuous surface for $V > 0$.

Considering that $v_i \neq 0$ at unstable weak equilibria as well as the definition of v_i in (6.18) it can be seen that in terms of Filippov solutions the velocity is an element of a bounded set, specifically

$$v_i \in \left[-c_1 \left(\frac{|\hat{y}_i|}{2|\mathcal{N}_i|} - \frac{L}{\tan\phi_{\max}} \right),\ c_1 \left(\frac{|\hat{y}_i|}{2|\mathcal{N}_i|} - \frac{L}{\tan\phi_{\max}} \right) \right] \backslash \{0\}.$$

Now to calculate estimates for the terms a and b of (6.14) we have to distinguish between two cases: The first case occurs when the saturation function $\sigma_{\phi_{\max}}(\cdot)$ has no effect on vehicle i, that is, when $|\arctan(c_2\omega_i L/v_i)| \leq \phi_{\max}$. As a result $\dot{\theta}_i = c_2\omega_i$ and (6.16) holds like in the free steering case. The second case contains all configurations in which the steering angle is at its limit and therefore $\dot{\theta}_i = v_i \operatorname{sgn}(-\hat{y}_i) \tan\phi_{\max}/L$. As such

$$|a| \overset{(6.16)}{\leq} 2|\mathcal{N}_i| \cdot |v_i| < \left| v_i |\hat{y}_i| \frac{\tan\phi_{\max}}{L} \right| = |\dot{\theta}_i \hat{y}_i| = |b|. \tag{6.22}$$

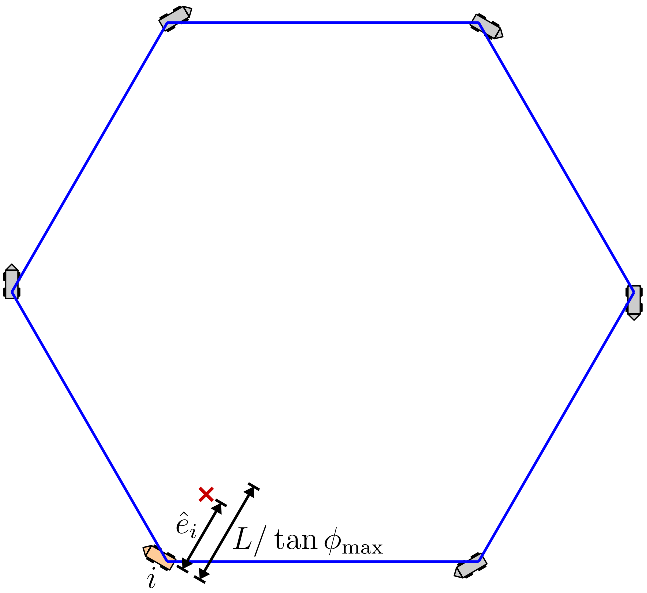


Figure 6.1: Example for an equilibrium configuration for the closed-loop controller defined by equations (6.18)-(6.19).

For the less-than relation it was considered that from $\hat{e}_i > L/\tan\phi_{\max}$ follows

$$2|\mathcal{N}_i| < |\hat{y}_i|\frac{\tan\phi_{\max}}{L}$$

when $\theta_i = 0$.

So as long as there is one vehicle i with $\hat{e}_i > L/\tan\phi_{\max}$ the system will break out of the set S. On the other hand we get $v_i = 0$ for all i if $\hat{e}_i \leq L/\tan\phi_{\max}$ for all vehicles, i.e., this part of S is invariant under the control law given by equations (6.18) and (6.19). As a result we can define the invariant set $S_{\text{inv}} \subseteq S$ as

$$S_{\text{inv}} = \left\{ p \in S_k : \sum_{i=1}^{N} |\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i| = 0 \wedge \hat{e}_j \leq L/\tan\phi_{\max} \; \forall j \in \{1,\dots,N\} \right\}.$$

□

As can be seen from Lemmas 6.5.1 and 6.5.2, the bounded steering angle of the vehicles limits the convergence properties of the closed-loop control system. We cannot even conclude the convergence to a certain region. As an example consider a group of vehicles arranged equidistantly on a circle with radius much bigger than $L/\tan\phi_{\max}$. Each vehicle is connected to the two closest robots and oriented perpendicular to the line through its position coordinates and the center of the circle (cf. Figure 6.1). It is easy to see that $\hat{e}_i < L/\tan\phi_{\max}$ as well as $\hat{v}_i = 0$ for all $i \in \{1,\dots,N\}$ and therefore the configuration is contained in the set S_{inv}.

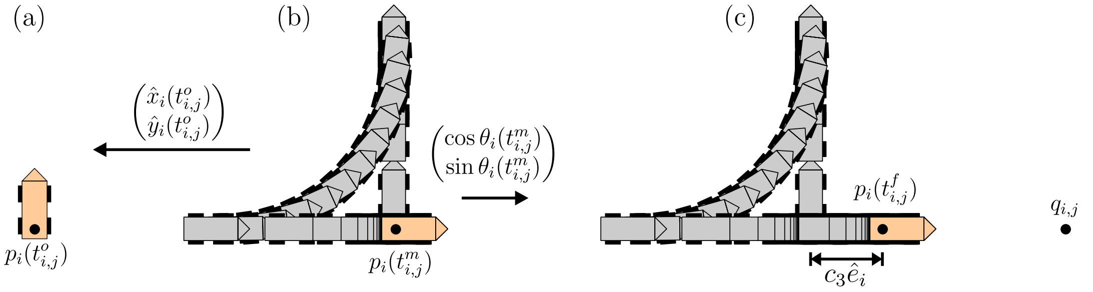

(a)
(b)
(c)
$\begin{pmatrix}\hat{x}_i(t^o_{i,j})\\ \hat{y}_i(t^o_{i,j})\end{pmatrix}$
$\begin{pmatrix}\cos\theta_i(t^m_{i,j})\\ \sin\theta_i(t^m_{i,j})\end{pmatrix}$
$p_i(t^o_{i,j})$
$p_i(t^m_{i,j})$
$p_i(t^f_{i,j})$
$q_{i,j}$
$c_3\hat{e}_i$

Figure 6.2: Illustration of a maneuver that satisfies (M1)-(M4). The figure shows a snapshot from the beginning of the maneuver (a), at time $t^m_{i,j}$ (b), and from the end of the maneuver (c). The point $q_{i,j}$ denotes the position for vehicle i that minimizes the V-function during the maneuver.

6.5.2 Introducing orientation adjustment maneuvers

Since we want to achieve convergence to a common position, we somehow have to complement the closed-loop controller such that the system is moved away from the undesired configurations of the invariant set. To achieve this, we add a switching logic to the controller that forces certain vehicles to leave the closed-loop system temporarily (cf. Figure 6.3). Such a robot then performs a two step maneuver with the following four properties:

(M1) $t^f_{i,j} - t^o_{i,j} < t_m < \infty$,

(M2) $p_i(t^m_{i,j}) = p_i(t^o_{i,j})$,

(M3) $\big(\cos\theta_i(t), \sin\theta_i(t)\big)^T = -1 \cdot \frac{(\hat{x}_i(t^o_{i,j}),\hat{y}_i(t^o_{i,j}))^T}{\left|\left|(\hat{x}_i(t^o_{i,j}),\hat{y}_i(t^o_{i,j}))^T\right|\right|}, \quad \forall t \in [t^m_{i,j}, t^f_{i,j}]$,

(M4) $p_i(t^f_{i,j}) = p_i(t^o_{i,j}) - \frac{c_3}{2|\mathcal{N}_i|}\begin{pmatrix}\hat{x}_i(t^o_{i,j})\\ \hat{y}_i(t^o_{i,j})\end{pmatrix}, \quad c_3 \in (0,1]$.

Here, $t^o_{i,j}$ and $t^f_{i,j}$ denote the starting and the ending time of the jth maneuver of the ith vehicle and $t^m_{i,j}$ is a point in time with $t^o_{i,j} < t^m_{i,j} < t^f_{i,j}$. (M1) states that the maneuver should be carried out within finite time and t_m is meant to be a global upper bound for all possible maneuvers. It is easy to see that such a t_m always exists for a properly defined maneuver and it depends on the maneuver's definition and on the initial position of the vehicles. Condition (M2) ensures that at time $t^m_{i,j}$ during the maneuver the corresponding vehicle's coordinates $p_i(t^m_{i,j})$ match the maneuver's initial position. From that point in time until the end of the maneuver the vehicle has to be oriented towards the target point $p_i(t^f_{i,j})$ as required by condition (M3). Finally, (M4) fixes the target point, where the constant $c_3 \in (0,1]$ has to

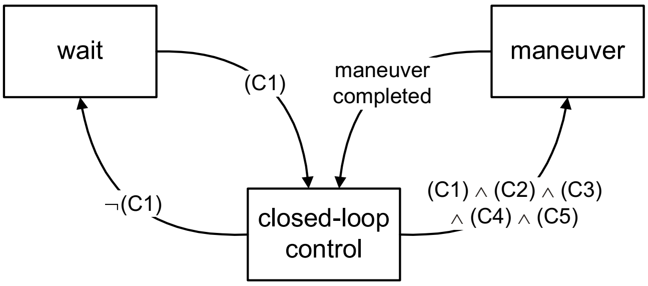


Figure 6.3: Automaton that describes the switching between controllers for each robot of the group. If conditions (C1)-(C5) are satisfied, the corresponding robot will switch to the open loop maneuver controller in order to adjust its orientation. For the neighbors of the maneuvering robot condition (C1) becomes untrue such that they will switch in order to wait for the maneuver to be completed.

be chosen globally. An example of such a maneuver is illustrated in Figure 6.2. During the maneuver the control input v_j of all neighbors of the maneuvering vehicle i is set to zero in order to prevent changes of $\hat{x}_i$ and $\hat{y}_i$ (waiting case in Figure 6.3).

As illustrated in Figure 6.3 robot i will start maneuvering if all of the following conditions are met:

(C1) None of the neighbors is currently maneuvering,

(C2) $|v_i| < \varepsilon_1$,

(C3) $\forall j \in \mathcal{N}_i : \ |v_j| < \varepsilon_1$,

(C4) $(\hat{x}_i \neq 0 \vee \hat{y}_i \neq 0) \quad \wedge \quad \Big((\cos\theta_i, \sin\theta_i) \cdot \frac{(\hat{x}_i,\hat{y}_i)^T}{||(\hat{x}_i,\hat{y}_i)^T||} < \varepsilon_2\Big)$,

(C5) $(\hat{e}_i > \max_{j\in\mathcal{N}_i} \hat{e}_j) \quad \vee \quad (\forall j \in \arg\max_{j\in\mathcal{N}_i} \hat{e}_j : \mathrm{ID}_i < \mathrm{ID}_j \wedge \hat{e}_i = \hat{e}_j)$.

Here, ε_1 and ε_2 are positive constants which should be chosen close to 0. These bounds are necessary to start maneuvers in finite time, since because of the definition of the closed-loop controller, zero is only reached as $t \to \infty$. Condition (C1) ensures that neighboring vehicles in the interaction graph will not maneuver at the same time. Note that multiple vehicles might still maneuver at the same time as long as the interaction graph is not complete. (C2)-(C4) are true at the same time if and only if robot i including its neighbors are close to an undesired equilibrium configuration of the closed-loop system described by (6.1), (6.18) and (6.19). In condition (C4) we use the normalized vector for the inner product rather than the original one. This is to avoid results close to zero when the vectors are not orthogonally aligned, which would happen when vehicles are located close to each other. Further the case where $\hat{x}_i = \hat{y}_i = 0$ is excluded, that is, when robot i is located at its momentary desired position. Since all vehicles are close to an equilibrium of the closed-loop system when (C2)-(C4) are true for $\varepsilon_1, \varepsilon_2$ small enough, a decision has to be made about which robot will do a maneuver. This is resolved by condition (C5), which ensures that vehicle i only maneuvers if the factor $\hat{e}_i$, which is proportional to the speed after the maneuver, is greater than the respective value of its neighbors. If it happens that neighboring robots' values of $\hat{e}_i$ are equal, we simply chose to favor the one with a lower ID. This implies of course that neighboring vehicles exchange the value of v_i and $\hat{e}_i$ in addition to their position coordinates.

Before stating the convergence results, we introduce a corollary and a lemma that will be used in the proof of the convergence theorems.

Corollary 6.5.3. *Consider the position vector p of the current configuration of a connected system of N vehicles, such that there exist i, j with $i \neq j$ and $p_i \neq p_j$. If p is an element of the set*

$$S_{\text{inv}} = \left\{ p : \sum_{i=1}^{N} |\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i| = 0 \wedge \hat{e}_j \leq L/\tan\phi_{\max} \ \forall j \in \{1, \ldots, N\} \right\}$$

or "close" to the set S_{inv}, then conditions (C1)-(C5) are satisfied for at least one vehicle.

Here, being "close" to the set S_{inv} means, the position vector nearly satisfies the set constraints of S_{inv}, where the tolerance depends on the choice of the constants $\varepsilon_i,\ i = 1, 2$, in conditions (C2)-(C4). Since the closed-loop controller drives the system asymptotically towards the set S_{inv}, the corollary states that if the vehicles do not head for a common position, a maneuver will be initiated within finite time. This can be easily verified by examining the conditions (C1)-(C5) for values from the set S_{inv} and its neighborhood.

Lemma 6.5.4. *Consider a connected system of N vehicles and let V^* be a constant with $V^* > 0$. Then, for all position vectors p that satisfy $V(p) \geq V^*$, there exists a positive constant $\mu = \mu(V^*, N)$ such that*

$$\exists i \in \{1, \ldots, N\} : \hat{e}_i \geq \mu. \tag{6.23}$$

Proof. From

$$\hat{e}_i = \frac{1}{2|\mathcal{N}_i|}\sqrt{\hat{x}_i^2 + \hat{y}_i^2}$$

it follows that the statement of Lemma 6.5.4 is true, if there exists a positive constant that bounds $\max_i \hat{x}_i$ from below for any p with $V(p) \geq V^*$ and any valid interaction topology.

Further, as a result of $V(p) \geq V^* > 0$ and the definition of V in (6.9) there have to exist two connected vehicles, which w.l.o.g. we set to be vehicle 1 and 2, with

$$|p_1 - p_2| \geq \nu := \sqrt{\frac{2V^*}{N(N-1)}}.$$

Also w.l.o.g. we can rotate the whole system in the plane such that

$$x_1 - x_2 \geq \nu. \tag{6.24}$$

Considering that

$$\hat{x} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_N \end{pmatrix} = L \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = Lx$$

where L corresponds to the graph Laplacian of the underlying interaction graph, we can now calculate

$$\min_x ||Lx||_\infty \qquad \text{s.t.} \qquad x_1 - x_2 \geq \nu. \tag{6.25}$$

Since every norm is positive definite and convex, there has to exist a global minimum for (6.25). This minimum has to be strictly positive, because the Laplacian of a connected graph always has a simple Eigenvalue at 0 with the one-dimensional Eigenspace $\text{span}\big((1,\dots,1)^T\big)$ (cf., e.g., [OSM04]) which, due to (6.24), does not contain the vector x. Calculating (6.25) for the finite number of valid graph Laplacians and dividing the smallest resulting value by $2|\mathcal{N}_i|$ will provide a $\mu > 0$ that satisfies (6.23) for all p with $V(p) \geq V^*$. □

With Corollary 6.5.3 and Lemma 6.5.4 we can now state the main result of this section that is concerned with the convergence of the overall system.

Theorem 6.5.5. *Consider a connected system of N car-like robots that evolve according to* (6.1) *with constraint* (6.17)*. Then, by applying the control law* (6.18)-(6.19) *combined with the orientation-adjustment-maneuver extension as depicted in Figure 6.3, the position coordinates p_i of all vehicles converge towards a common value as $t \to \infty$.*

Proof. For every maneuver j we have $p_i(t^o_{i,j}) = p_i(t^m_{i,j})$ by definition. Thus, we can assume for all maneuvers that the position p_i of each robot stays constant during the time interval $[t^o_{i,j}, t^m_{i,j}]$. Now let $N \geq 2$ be the number of vehicles. We know from Lemma 6.5.2 that the closed-loop controller drives the system's position variables towards the set S_{inv}. It remains to show convergence for the case when the closed-loop controller's limit is not a common position. From Lemma 6.5.1 and the assumption that the position of each vehicle i stays constant during the time interval $[t^o_{i,j}, t^m_{i,j}]$, it results that the Lyapunov function $V\big(p(t)\big)$ defined in (6.9) is a monotonically nonincreasing function of t that is upper bounded by $V\big(p(0)\big)$ and lower bounded by zero. As a result there must exist a finite limit $V^* = \lim_{t\to\infty} V\big(p(t)\big)$. In the remainder of the proof we will show that this limit is equal to zero, which corresponds to the configurations where all vehicles are located at a common position. Now let us assume $V^* > 0$. Then due to Lemma 6.5.4 there always has to exist a vehicle i with $\hat{e}_i \geq \mu$ for all configurations p with $V^* \leq V(p) \leq V\big(p(0)\big)$, where μ is a finite positive constant that depends on V^* and the number of vehicles N.

From equation (6.21) it follows that

$$-2c_1(\hat{v}_i^2 + |\hat{v}_i|\hat{b}_i) \leq -2c_1\hat{v}_i^2 \leq 0,$$

and therefore no vehicle will ever contribute to an increase of the V-function by means of the closed-loop controller. Thus it suffices to concentrate on a vehicle i with $i \in \arg\max_j \hat{e}_j$ that satisfies maneuver condition (C5) and therefore cannot be in the waiting state. It follows from Corollary 6.5.3 that such a vehicle i will start an orientation adjustment maneuver within finite time, if the vehicle and its neighbors are not heading towards a common position.

We show now that in case vehicle i performs an orientation adjustment maneuver, the nonpositive difference between the V-values before and after the jth maneuver of vehicle i

$$\Delta V_{i,j} = V\big(p(t_{i,j}^f)\big) - V\big(p(t_{i,j}^o)\big) \tag{6.26}$$

is nonzero and upper bounded. Since the neighboring vehicles have to wait during the maneuver, the contribution of vehicle i to the Lyapunov function (6.9) can be written as

$$V_i(x_i, y_i) = \frac{1}{2}\sum_{j\in\mathcal{N}_i} \left(\big(x_i - x_j(t_{i,j}^o)\big)^2 + \big(y_i - y_j(t_{i,j}^o)\big)^2\right). \tag{6.27}$$

It is easy to see that the unique global minimum of (6.27) is located at

$$q_{i,j} := \underset{(x_i,y_i)^T\in\mathbb{R}^2}{\arg\min}\; V_i(x_i, y_i) = \frac{1}{|\mathcal{N}_i|}\left(\sum_{j\in\mathcal{N}_i} x_j(t_{i,j}^o), \sum_{j\in\mathcal{N}_i} y_j(t_{i,j}^o)\right)^T.$$

Condition (M3) and (M4) from the maneuver definition ensure that vehicle i decreases its distance to the point $q_{i,j}$ by $c_3\hat{e}_i \geq c_3\mu$ during the time interval $[t_{i,j}^m, t_{i,j}^f]$. As a result the value of $V_i(x_i, y_i)$ and therefore also the value of the Lyapunov function (6.9) decreases. An estimate for $\Delta V_{i,j}$ is given by

$$\begin{aligned}
\Delta V_{i,j} &= V\big(p(t_{i,j}^f)\big) - V\big(p(t_{i,j}^o)\big)\\
&\leq V\big(p(t_{i,j}^f)\big) - V\big(p(t_{i,j}^m)\big) \leq V_i\big(x_i(t_{i,j}^f), y_i(t_{i,j}^f)\big) - V_i\big(x_i(t_{i,j}^m), y_i(t_{i,j}^m)\big)\\
&= V_i\big(x_i(t_{i,j}^f) - c_3\hat{x}_i(t_{i,j}^f)/(2|\mathcal{N}_i|), y_i(t_{i,j}^f) - c_3\hat{y}_i(t_{i,j}^f)/(2|\mathcal{N}_i|)\big) - V_i\big(x_i(t_{i,j}^m), y_i(t_{i,j}^m)\big)\\
&= -\frac{4c_3 - c_3^2}{8|\mathcal{N}_i|}\left(\left(\sum_{j\in\mathcal{N}_i} x_j - |\mathcal{N}_i|x_i\right)^2 + \left(\sum_{j\in\mathcal{N}_i} y_j - |\mathcal{N}_i|y_i\right)^2\right)\\
&= -\frac{4c_3 - c_3^2}{8|\mathcal{N}_i|}\left(\hat{x}_i^2 + \hat{y}_i^2\right) \overset{(6.20)}{=} -\frac{4c_3 - c_3^2}{2}|\mathcal{N}_i|\hat{e}_i^2 \leq \Delta V_\mu := -\frac{4c_3 - c_3^2}{2}\mu^2.
\end{aligned} \tag{6.28}$$

So as long as the vehicles do not tend towards a common position by means of the closed-loop controller, there will always be a vehicle i with $\hat{e}_i \geq \mu$ that performs an orientation adjustment maneuver within finite time. Let t_k, $k \in \mathbb{N}^+$ denote the ending times of those maneuvers. Now since $V^* > 0$ is assumed to be the limit of the positive semidefinite function $V\big(p(t_k)\big)$, the inequality

$$V\big(p(t_k)\big) \geq V^* \quad \forall k \in \mathbb{N} \tag{6.29}$$

should hold. But since we have

$$V\big(p(t_k)\big) \leq V\big(p(t_0)\big) - k\Delta V_\mu$$

as long as (6.29) holds, it follows that (6.29) is not satisfied for k chosen sufficiently large. Because of this contradiction we can conclude that V^* has to be zero and the vehicles' positions converge towards a common value as $t \to \infty$. □

In Theorem 6.5.5 we assumed that the position of maneuvering vehicles does not change in the first part of each maneuver. This is crucial for the proof of convergence for the designed system, because if we look at any configuration of the set S_{inv} with not all vehicles at a common position, it is easy to see that due to the steering constraint (6.17) the V-function would increase for any kind of motion. Note that Theorem 6.5.5 also holds without the assumption that $v_i \neq 0$ at unstable weak equilibria as it was necessary for Theorem 6.4.1 and Lemma 6.5.2. This is because in such a situation one of the vehicles will satisfy conditions (C1)-(C5) and therefore starts an orientation adjustment maneuver that drives the system away from the undesired equilibrium while achieving a decrease of the V-function at the same time. Further note that for the proof of Theorem 6.5.5 we only considered the decrease of the V-function during recurring maneuvers of vehicles with $\hat{e}_i \geq \mu$. Nevertheless, the convergence time does not necessarily decrease with an increasing number of these maneuvers. In general, the decrease of the V-function is even much higher during intervals without any maneuvers as can be observed from the simulation results described in Sections 6.6 and 6.8.2. The problem is that it is more difficult to define a lower bound for the amount of decrease than in case of a maneuver.

To get an estimate for the decrease we now consider the case where a vehicle i with $i \in \arg\max_j \hat{e}_j$, $\hat{e}_i \leq L/\tan\phi_{\max}$ and condition (C5) satisfied does not perform an orientation adjustment maneuver within a nonempty time interval t_n. As a result this robot does not satisfy at least one of the conditions (C1)-(C4), since otherwise it would start an orientation adjustment maneuver. Condition (C1) must be satisfied, or else one vehicle j with $\hat{e}_j \geq \hat{e}_i$ would perform a maneuver, which is excluded by assumption. If (C2) is not satisfied, we have $|v_i| \geq |c_1 \hat{v}_i| \geq \varepsilon_1$ by definition of the control law (6.18) and therefore $|\hat{v}_i| \geq \varepsilon_1 / c_1$. Now it follows from equation (6.21) that

$$\dot{V} \leq -2c_1 \hat{v}_i^2 \leq -\frac{2\varepsilon_1^2}{c_1}. \tag{6.30}$$

In the same way we obtain

$$\dot{V} \leq -2c_1 \hat{v}_j^2 \leq -\frac{2\varepsilon_1^2}{c_1}, \tag{6.31}$$

if (C3) is not satisfied. For the case that (C4) is not satisfied we can estimate the absolute value of the ith robot's speed as

$$|v_i| \geq |c_1 \hat{v}_i| = |c_1(\cos\theta_i \cdot \hat{x}_i + \sin\theta_i \cdot \hat{y}_i)| \geq c_1 \varepsilon_2 \underbrace{|(\hat{x}_i, \hat{y}_i)|}_{2\hat{e}_i |\mathcal{N}_i|} \geq 2c_1 \varepsilon_2 \mu,$$

and therefore $|\hat{v}_i| \geq 2\varepsilon_2 \mu$ as well as

$$\dot{V} \leq -2c_1 \hat{v}_i^2 \leq -2c_1 \varepsilon_2^2 \mu^2. \tag{6.32}$$

Thus, from (6.30)–(6.32) a lower bound for the amount of decrease of the V-function can be derived as

$$V\big(p(t+t_n)\big) - V\big(p(t)\big) \leq \max\left\{-\frac{2\varepsilon_1^2}{c_1} t_n,\ -2c_1 \varepsilon_2^2 \mu^2 t_n\right\} < 0.$$

Compared to this, the case when $\hat{e}_i > L/\tan\phi_{\max}$ for vehicle i is more problematic. This is because if condition (C4) is satisfied, the value for $\hat{v}_i$ can be arbitrary close to zero while no maneuver is initiated and therefore the value for $\dot{V}$ can be arbitrary small. We know from Lemma 6.5.2 that the closed-loop controller drives the system towards the set S_{inv} and therefore it will happen within finite time that $\hat{e}_i \leq L/\tan\phi_{\max}$. During this process it might happen that vehicle i performs an orientation adjustment maneuver depending on the value of $\hat{v}_i$ when $\hat{e}_i \approx L/\tan\phi_{\max}$. This causes no problem as the maneuver's time is upper bounded by t_m and therefore it reduces the value of $\hat{e}_i$ to $(1 - c_3)\hat{e}_i$ within finite time. Nevertheless it turned out to be difficult to derive a lower bound for the amount of decrease of V in this case.

6.5.3 Distributed termination of the rendezvous process

In a real-world implementation vehicles cannot overlap without causing a collision. Thus, in practice it is desired to gather the vehicles within a certain area rather than in one point. In other words, we want the rendezvous process to stop, when the distance between any two vehicles falls below a certain bound, that is, when

$$\max_{i,j} |p_i - p_j| = \max_{i,j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq d_{\max} \tag{6.33}$$

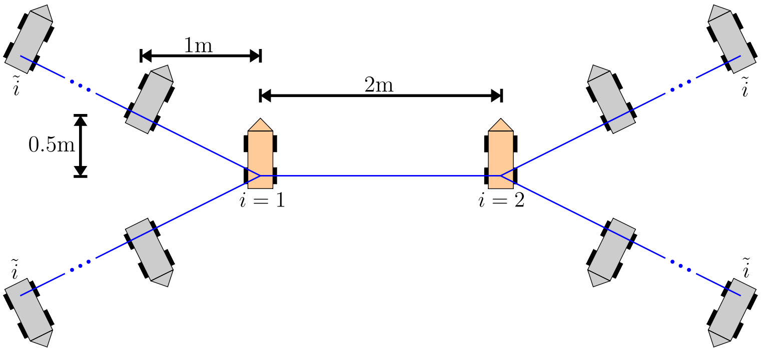


Figure 6.4: System configuration that demonstrates the impossibility of a local stopping criterion. The robots denoted by $\tilde{i}$ are the only ones with $\hat{e}_{\tilde{i}} > 0$, but for $d_{\max} = n \cdot 1.2$m they satisfy (6.34) and (6.36) for arbitrary $N \geq 5$. But they could be located many nodes away from the vehicles with $i \in \{1, 2\}$ and $\hat{e}_i = 0$ that do not satisfy the local stopping criteria.

for some $d_{\max} > 0$. This condition automatically holds if for every vehicle $i \in \{1, \ldots, N\}$ the local condition

$$\max_{j \in \mathcal{N}_i} |p_i - p_j| \leq \frac{d_{\max}}{N-1}, \tag{6.34}$$

is satisfied. To derive another local condition that implies (6.33) consider a group of N robots with a connected interaction graph. It can be easily verified that the lowest value for the V-function, such that $\max_{i,j} |p_i - p_j| = d_{\max}$, is achieved when all vehicles are arranged equidistantly on a line of length $d_{\max}$ and share an interaction link with their nearest neighbors. Since the outer robots have only one neighbor in this configuration, we have

$$V = \frac{d_{\max}^2}{N-1} \tag{6.35}$$

for this configuration. Note that if we also consider the interaction structure to be known, the denominator could be replaced by the number of edges of the longest path in the interaction graph that does not contain a cycle. By taking into account additional details of the interaction graph it might be possible to derive an even higher value for the righthandside of (6.35). Based on the definition of V in (6.9), a second local condition that implies (6.33) is given by

$$V_{i,\text{local}} := \sum_{j \in \mathcal{N}_i} \left((x_i - x_j)^2 + (y_i - y_j)^2\right) \leq \frac{2}{N} \cdot \frac{d_{\max}^2}{N-1}, \quad \forall i \in \{1, \ldots, N\}. \tag{6.36}$$

Unfortunately in the proposed system, condition (6.33) cannot be validated in a decentralized fashion. This becomes clear from the situation depicted in Figure 6.4, where the two robots closest to the center do not satisfy (6.34) nor (6.36). For these vehicles we also have

$\hat{e}_1 = \hat{e}_2 = 0$, which implies that they stand still at the moment. In contrast, the robots in the corners denoted by $\tilde{i}$ are the only ones with $\hat{e}_{\tilde{i}} > 0$, but they satisfy both local stopping criteria. Nevertheless they do not have any information about the vehicles with $i \in \{1, 2\}$. The dots in the interaction links indicate that there could be any finite number of equally spaced vehicles with the same orientation, which also would have $\hat{e}_i = 0$. So the information whether to continue or to stop the rendezvous system can be located arbitrarily far away in the interaction graph.

To efficiently stop the execution of the rendezvous controller when (6.33) is satisfied, we suggest the following broadcasting strategy. The idea of this strategy is that vehicles which do not satisfy (6.36) will broadcast this information from time to time over the network. Assuming that a connected communication topology is available, this can be simply achieved by a communication packet, in the following referred to as SP, which contains the vehicle ID and an incremental packet number. After the SP was sent to all neighbors of the broadcasting robot, the receiving nodes should forward the data to all of their neighbors excluding the sending unit. If a robot receives the same SP for the second time it will be simply discarded. For simplicity we assume that there exists a maximum broadcasting time t_b that depends on the longest path in the communication graph of the system. This means that a broadcasted message will be received by all vehicles of the system within the time interval t_b after its initiation. To integrate the SPs into the controller's logic we let successive broadcasts of the same vehicle happen after a period of t_B, which should correspond to the time needed for doing the first part of a maneuver, i.e., $t_B := t^m_{i,j} - t^o_{i,j}$. Now we disallow doing maneuvers for all vehicles that satisfy (6.36) and did not receive a SP within a period of t_B. Note that since the vehicles are homogeneous and we assume that every robot applies the same maneuver controls, the difference $t^m_{i,j} - t^o_{i,j}$ is the same for all valid choices of i and j. Further, any vehicle that did not receive a SP for a finite amount of time $t^*_B > t_B$ should stop all control actions immediately.

The described broadcasting extension gives rise to the following theorem.

Theorem 6.5.6. *Consider a connected system of N car-like vehicles with bounded steering angle. By applying the control system depicted in Figure 6.3 combined with the broadcasting extension described above, the vehicles will stop maneuvering after finite time and tend to a configuration that satisfies* (6.33) *for a chosen $d_{\max} > 0$.*

Proof. If all vehicles initially satisfy (6.36) we are done. If not, after a period t_b every vehicle will have the permission to move and to do maneuvers as long as there exists a vehicle that does not satisfy (6.36). Now, from Theorem 6.5.5 we can conclude that $V\big(p(t_k)\big)$ is decreasing until there exist no more vehicles satisfying (6.36). Once this is the case for a k^*, we know from (6.35) that $V\big(p(t_k)\big)$ has to be less or equal than $d_{\max}^2/(N-1)$ for all $k > k^*$, which implies that (6.33) holds.

It remains to show that the vehicles will stop maneuvering after finite time. This is clear if there will be no more SP after a period of $t_b + t_B^*$ after the last SP has been distributed. Indeed it is possible that (6.36) becomes untrue for vehicles that previously satisfied the criterion by application of the closed-loop controller, since the time-derivative of $V_{i,\text{local}}$

$$\dot{V}_{i,\text{local}} = 2\sum_{j\in\mathcal{N}_i}\Big(v_i\big((x_i-x_j)\cos\theta_i + (y_i-y_j)\sin\theta_i\big) + v_j\big((x_j-x_i)\cos\theta_j + (y_j-y_i)\sin\theta_j\big)\Big)$$

can take positive as well as negative values.

Let us assume that there exists always a vehicle that does not satisfy (6.36) before all vehicles come to a stop after a period of $t_b + t_B^*$. As a result $V\big(p(t_k)\big)$ continues decreasing such that after finite time it will be less or equal than $d_{\max}^2/\big(N(N-1)\big)$. But since V is non-increasing and

$$V = \frac{1}{2}\sum_{i=1}^{N} V_{i,\text{local}}, \tag{6.37}$$

this stands in contradiction with the existence of a vehicle that satisfies (6.36). □

The local stopping criterion (6.34) is closer related to (6.33) than (6.36). Nevertheless it is less applicable as a stopping criterion since we have no results about the development of the summed distances. Indeed there exist situations where the summed squared distances as they occur in (6.36) and in the V-function decrease, while the summed non-squared distances increase.

To guarantee that there will be no more maneuvers after finite time it is necessary to stop the control actions after finite time t_B^* after the last received SP. If we would not do this one can find a configuration for which all vehicles satisfy (6.36) and one vehicle converges asymptotically to a point where (6.36) becomes untrue by application of the closed-loop controller. As a result it will send another SP after an arbitrary long time interval.

6.6 Simulation results

In this section we present selected simulation results of the rendezvous control systems introduced in the previous sections. With the first randomly generated scenario we demonstrate the different behavior of the controllers for the free and for the bounded steering case. The second scenario shows how the bounded steering rendezvous system acts in a constructed setup in which most of the vehicles initially are located at their desired reference point. For the simulation runs we set the kinematic constant denoting the distance between front and rear axle of the vehicles to $L = 0.3$m.

6.6.1 Scenario 1

The first scenario contains ten car-like robots, whose initial position and orientation was chosen randomly, where the position variables were limited to a rectangle of 20m $\times$ 20m. The structure of the interaction graph was chosen to be the minimal fully connected proximity graph of the initial system configuration. We present two simulation runs of this scenario, one for the free steering case (Figure 6.5) and one for the bounded steering case (Figure 6.7) with a maximum steering angle $\phi_{\max} = 10°$. The corresponding speed and steering profiles as well as the development of the V-function and of $V_{i,\text{local}}$ can be found in Figure 6.6 for the free steering case and in Figure 6.8 for the bounded steering case respectively. The interaction graph is depicted in the first of the simulation snapshots.

For the free steering case we applied the control law defined by equations (6.5) and (6.6), where the constants of the controller were chosen as $c_1 = 6$ and $c_2 = 12$, which clearly satisfy the constraint (6.7). For the bounded steering case on the other hand we applied the control system depicted in Figure 6.3 combined with the broadcasting extension described in Subsection 6.5.3. There we chose the controllers constants as $c_1 = 3$ and $c_2 = 6$ in order to achieve approximately the same initial speed for the vehicles. Note that if we increase (decrease) the values of c_1 and c_2 by the same percentage, the resulting vehicle paths will be the same with an increased (decreased) speed profile. Therefore, the time until the local stopping criterion (6.36) is satisfied decreases (increases) for all vehicles. The constant that characterizes the amount of movement in the final phase of the maneuvers in the bounded steering case was set to $c_3 = 0.2$.

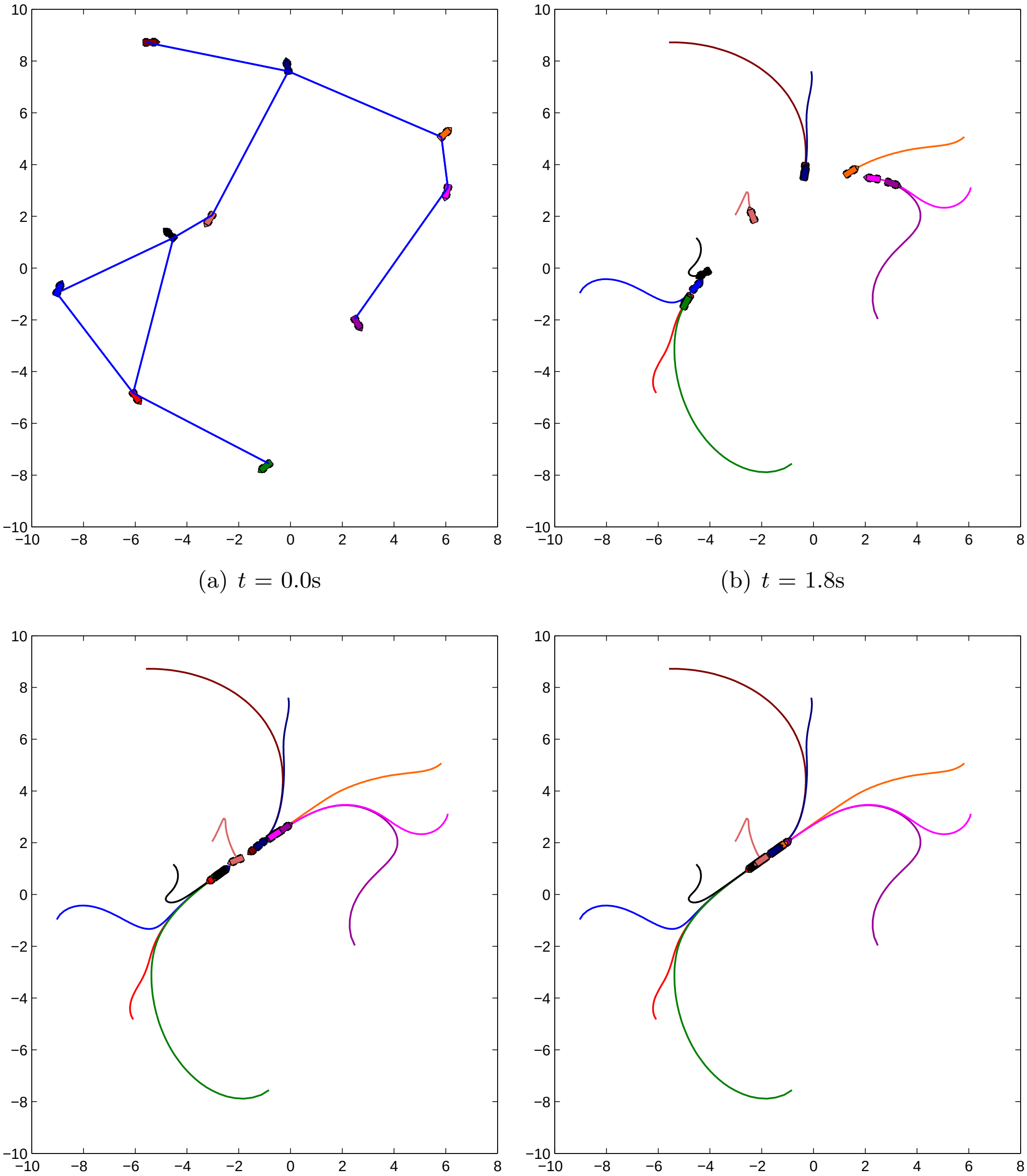
In this scenario we set $d_{\max} = 4$m. For more illustrative figures we let the system continue with the control actions for some time even after (6.36) has been satisfied for all vehicles. The global stopping criterion (6.33) is reached, once the value of the V-function has fallen below 1.78. In the free steering case this happens after 6.79s and in the bounded steering case after 16.22s. The local stopping criterion $V_{i,\text{local}} \leq 0.36$ is reached for all vehicles after 9.28s in the free steering case and after 19.97s in the bounded steering case as can be observed in the plots of $V_{i,\text{local}}$ in Figures 6.6 and 6.8. Since in this scenario the longest path in the interaction graph without cycles contains seven edges, the benefit of the relaxed stopping criterion mentioned above is rather small. The resulting constraint $V_{i,\text{local}} \leq 0.46$ becomes true for all i after 8.74s with free steering and after 19.47s in the bounded steering case.

In the speed profile in Figure 6.6 we do not observe a curve passing through zero. That means there is no vehicle changing its movement direction during the simulation run, which is caused by the free steering angle. Note that this does not mean that a change of the movement direction will never happen with the free steering controller. If the moving reference point of a vehicle's controller, given by $\big(x_d(t), y_d(t)\big)^T = 1/(2|\mathcal{N}_i|)\big(\hat{x}_i(t), \hat{y}_i(t)\big)^T$, moves faster than the vehicle will follow with its orientation, the sign of the robot's speed will change. In certain cases this will also happen when the moving reference point passes through the vehicle's position coordinates. As can be seen in the speed profile in Figure 6.8, in the bounded steering case it is more likely that vehicles change their movement direction because of the limited steering angle.

It can be observed that in both cases (free and bounded steering) the vehicles head towards a common position and the non-increasing V-function tends towards zero even if $V_{i,\text{local}}$ partly increases for some of the vehicles. In the bounded steering case the closed-loop controller reaches one of the undesired equilibria at $t \approx 6.9$s and two of the vehicles start an orientation adjustment maneuver with a slight offset. The time interval of the maneuvers is depicted in the V-plot in Figure 6.8, where the inner vertical dash denotes $t_{i,1}^m$, the point in time when the corresponding vehicle's coordinates match the maneuver's initial position. In the corresponding speed and steering profiles one can observe that the inputs during the maneuver are piecewise continuous, which results from the simple design of the maneuver. The forward motions during the maneuver are achieved by utilization of the position stabilizing controller that was introduced in Subsection 3.2.2 and the backward motion that stabilizes the car onto the line through the points $p_i(t_{i,j}^m)$ and $p_i(t_{i,j}^f)$ are achieved by the controller described in [LKYC99, Proposition 1]. Since with these controllers the desired equilibrium is not reached within finite time, we switch to the subsequent one, once the position and orientation error have fallen below a certain value ($< 10^{-3}$). As it was shown in Subsection 6.5.2 there is always a certain decrease of the V-function in the final part of the maneuvers, but it is hard to see in the plot of V in Figure 6.8 due to its relative smallness. This decrease can be seen much clearer in the second simulation scenario.

6.6.2 Scenario 2

The second scenario is essentially the example depicted in Figure 6.4 that was used to demonstrate the impossibility of a local stopping criterion for the rendezvous system for vehicles with bounded steering angle. For this scenario we only present the results for the bounded steering case with a maximum steering angle $\phi_{\max} = 10°$. Six snapshots of the simulation run are depicted in Figure 6.9 and the corresponding speed and steering angle profiles as well as the development of the V-function and of $V_{i,\text{local}}$ can be found in Figure 6.10. For this scenario the distance of the stopping criterion was chosen to be $d_{\max} = 12m$ which is already satisfied in the global sense. The maneuver control inputs were calculated in the same way as



(a) $t = 0.0$s

(b) $t = 1.8$s

(c) $t = 6.6$s

(d) $t = 10.8$s

Figure 6.5: Rendezvous simulation scenario 1 snapshots: free steering. The figure consists of four snapshots from the simulation run, while (a) shows the initial configuration and the underlying interaction topology.

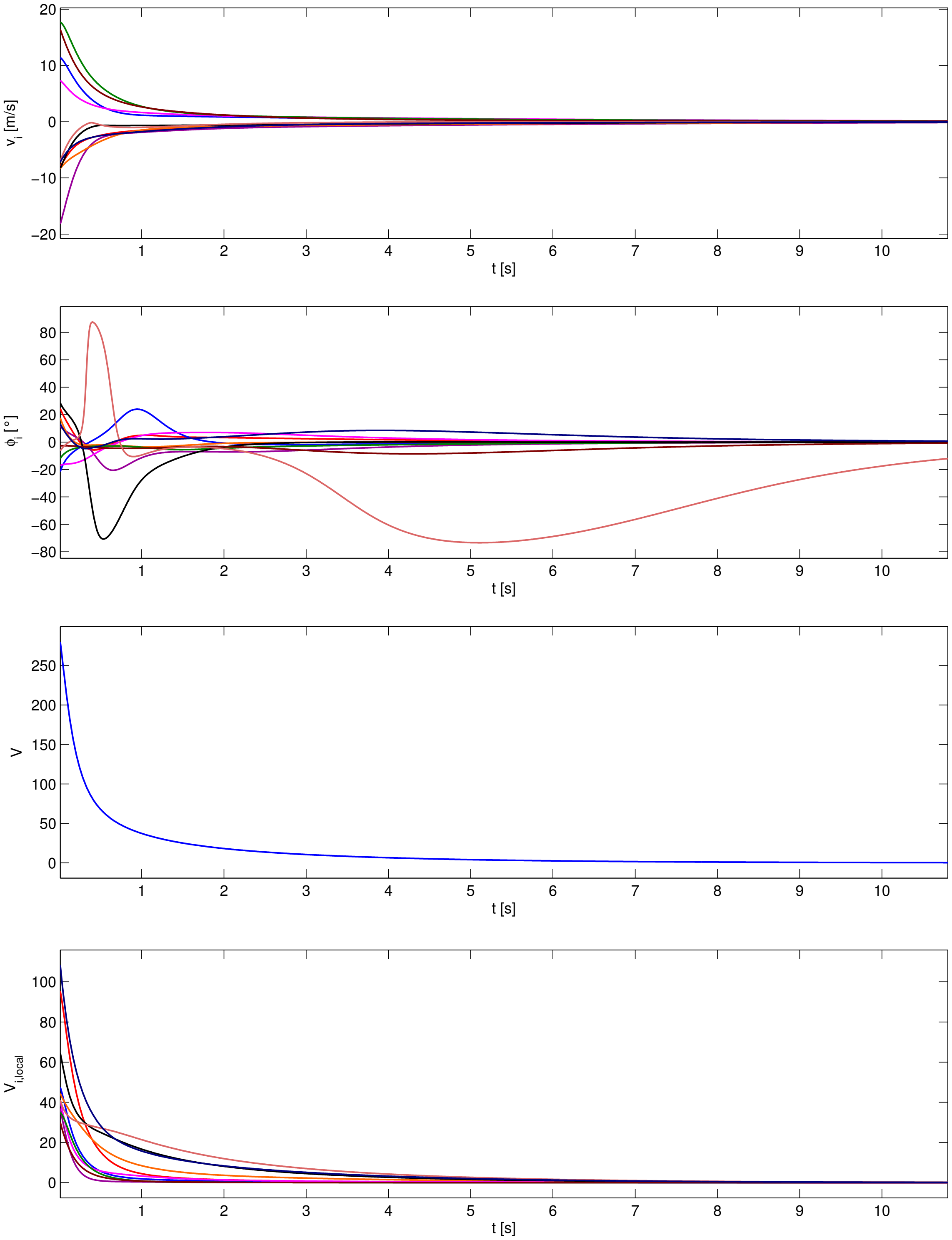
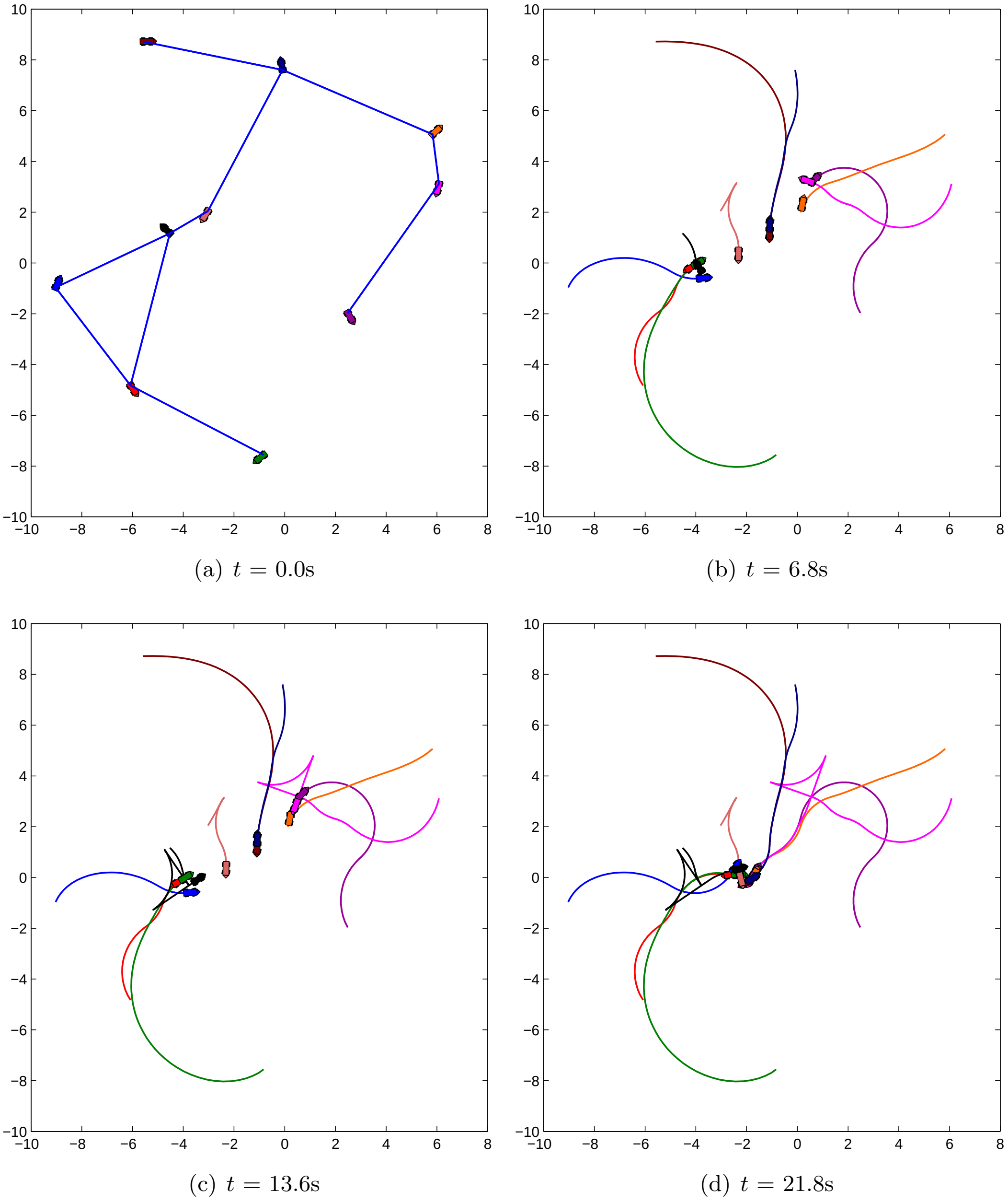


Figure 6.6: Rendezvous simulation scenario 1 plots: free steering. The figure depicts the corresponding speed and steering profiles as well as the plots of V and $V_{i,\text{local}}$.



(a) t = 0.0s

(b) t = 6.8s

(c) t = 13.6s

(d) t = 21.8s

Figure 6.7: Rendezvous simulation scenario 1 snapshots: bounded steering. The figure consists of four snapshots from the simulation run, while (a) shows the initial configuration and the underlying interaction topology.

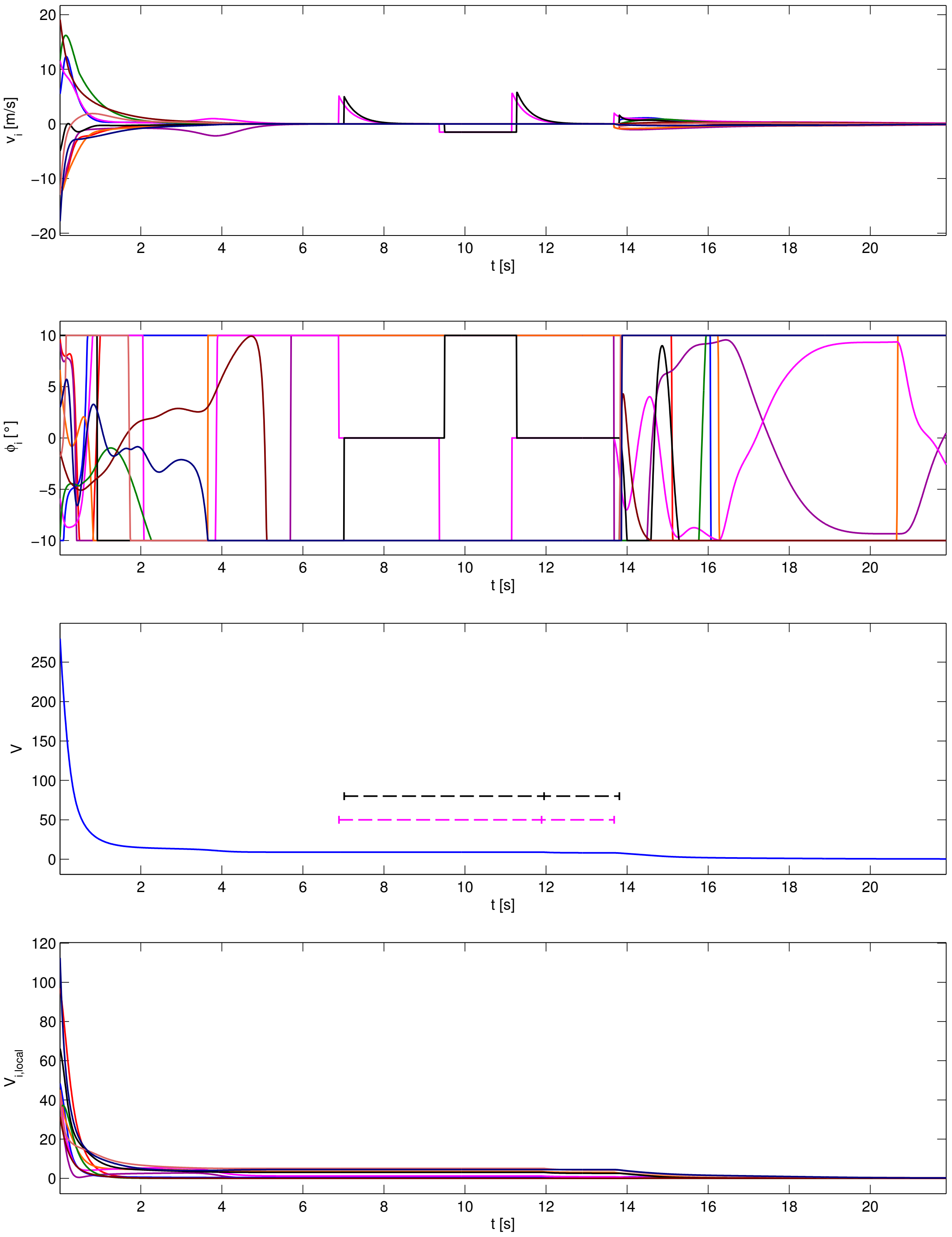
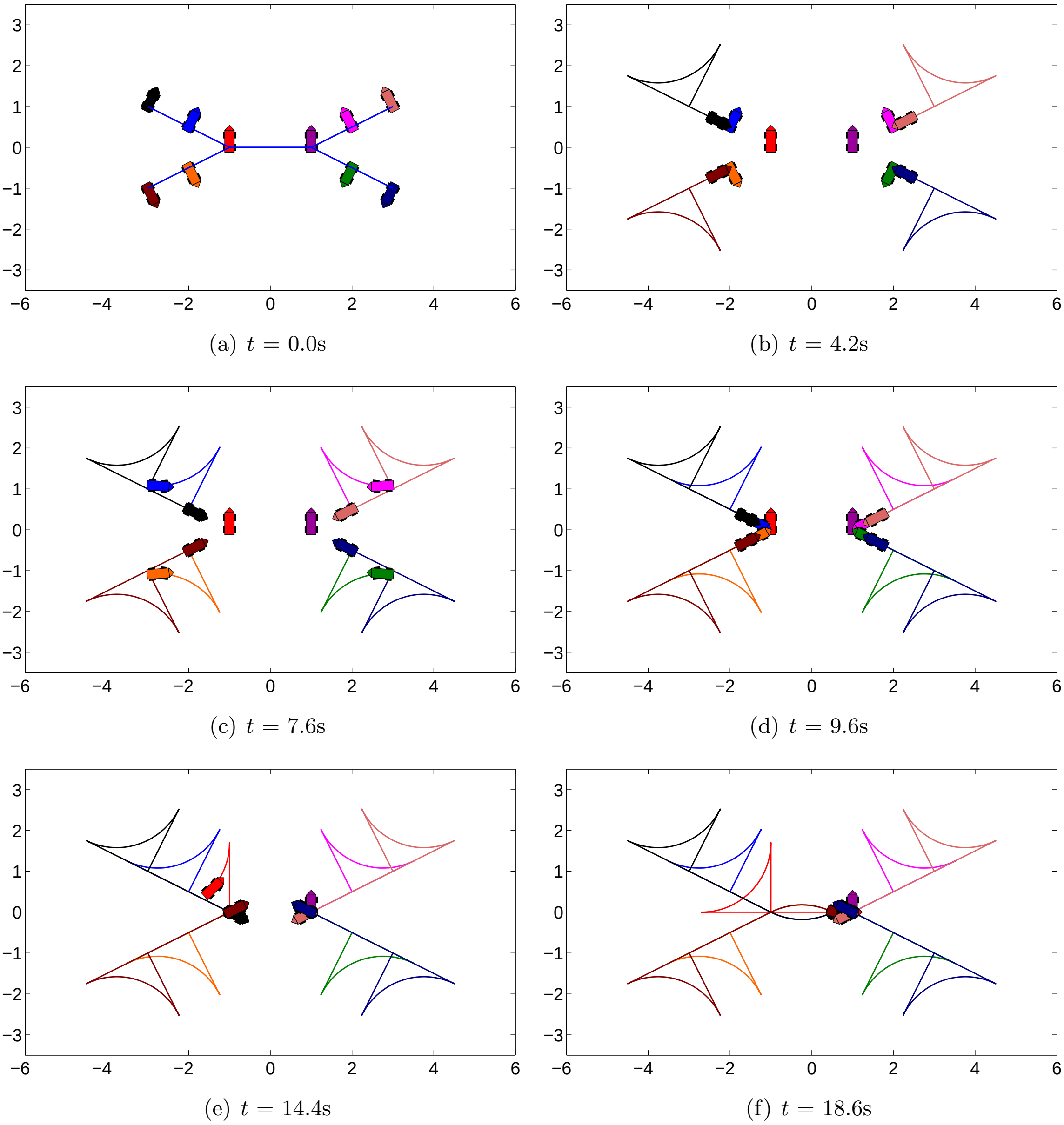


Figure 6.8: Rendezvous simulation scenario 1 plots: bounded steering. The figure depicts the corresponding speed and steering profiles as well as the plots of V with indicated maneuver intervals and $V_{i,\mathrm{local}}$.



(a) t = 0.0s

(b) t = 4.2s

(c) t = 7.6s

(d) t = 9.6s

(e) t = 14.4s

(f) t = 18.6s

Figure 6.9: Rendezvous simulation scenario 2 snapshots: bounded steering. The figure consists of six snapshots from the simulation run, while (a) shows the initial configuration and the underlying interaction topology.

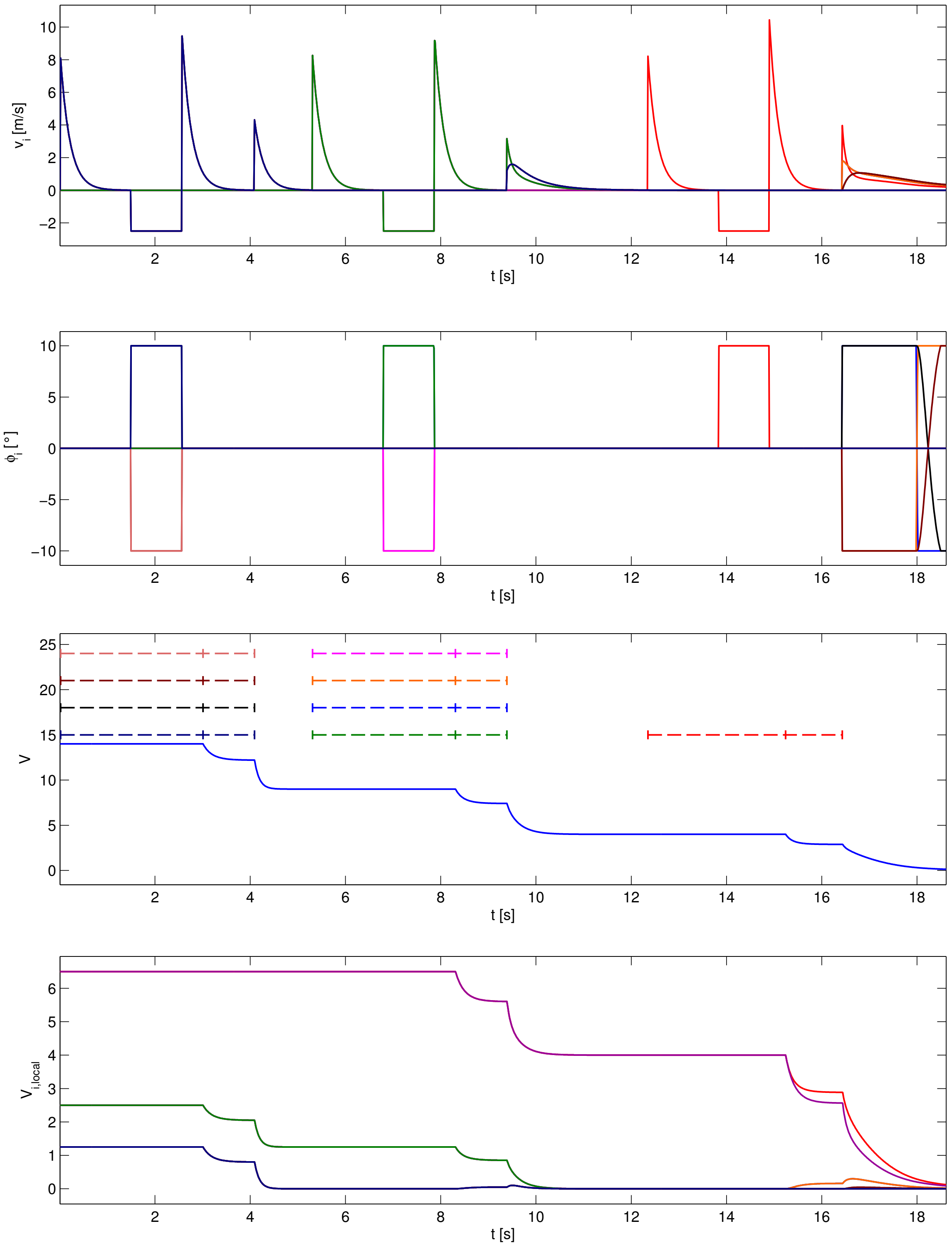


Figure 6.10: Rendezvous simulation scenario 2 plots: bounded steering. The figure depicts the corresponding speed and steering profiles as well as the plots of V with indicated maneuver intervals and $V_{i,\text{local}}$.

it was done for the bounded steering case in the first scenario, while this time the closed-loop controller's constants were set to $c_1 = 5$ and $c_2 = 10$.

At the beginning of the simulation run, the four outer vehicles do an orientation adjustment maneuver. All other vehicles remain motionless, since they are located at the current position of their reference point, which means we have $\hat{x}_i = \hat{y}_i = 0$ for those robots. After completing the maneuver these four robots converge towards the position of their only neighbor. Once they are close enough such that maneuver condition (C3) is satisfied from the perspective of their neighbor, each of them performs an orientation adjustment maneuver as can be observed in the snapshot in Figure 6.9(c). After this, the two vehicles on each of the four sides head towards one of the robots in the center with a certain offset as can be seen from the differences in the corresponding speed profile in Figure 6.10 at $t \approx 9.4$s. This offset comes from the fact that the two vehicles follow different reference points, resulting from different neighbors in the interaction graph. While the reference point for the vehicles of the second maneuver wave is located between them and the inmost vehicles, the initially outside located robots' reference points are located between themselves and their only neighbor. The two inmost vehicles receive the permission for a maneuver at the same time. Since they are connected and the distance to their reference point is the same, only the left robot performs a maneuver because of its lower ID. During the final part of this maneuver at $t \approx 15.44$s the local stopping criterion ($V_{i,\text{local}} \leq 3.2$) becomes true for all of the vehicles. In this scenario the relaxed stopping criterion has a much bigger impact than it was observed in the first scenario. With a longest cycle-free path of five edges, the relaxed criterion $V_{i,\text{local}} \leq 5.76$ is reached after approximately 8.64s. As can be seen from the plot of V in Figure 6.10 the global stopping criterion $V \leq 16$ is satisfied from the beginning with the chosen value for d_{max}.

6.7 Hardware experiments

We carried out a series of laboratory hardware experiments that demonstrate the applicability of the presented rendezvous control system on real robots of the car-like type. The distance between front and rear axle of the four utilized Indoor MERLINs (cf. Appendix B) was determined as $L = 0.278$m. As a result of the natural steering bound of car-like vehicles, we always utilized the rendezvous control system developed for the bounded steering case, where we set the corresponding constants to $c_1 = 0.4$, $c_2 = 0.8$ and $c_3 = 0.1$. It is well known that updating a robot's position via dead reckoning involves an accumulating position error. Nevertheless if the utilized sensors are well calibrated, the resulting accuracy is good enough for short-term hardware experiments like the ones described in the remainder of this section.

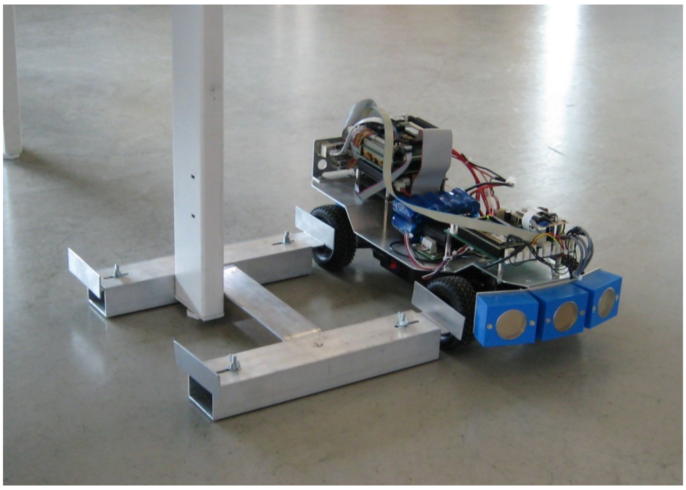


Figure 6.11: MERLIN robot with positioning device. After placing the robot, its dead reckoning module is reset and it is moved to to its desired initial configuration via remote control.

In principle the applied control inputs for a robot depend on the relative distance vectors between itself and its neighbors in the interaction graph. Thus, the location of the origin has no effect on the resulting system behavior. However, since the utilized vehicles are not able to determine the location of their neighbors it is essential for the experiments to match the vehicles global coordinate systems. In the first scenario presented in Subsection 6.7.1 we achieve this by resetting the robots dead reckoning module at the same location (cf. Figure 6.11), one after the other. After the reset, the robot is moved to its desired initial configuration via remote control. In the second scenario we simply set the initial configuration in the robots system and placed the vehicles in the environment respectively.

Once all vehicles are set up, we remotely start the rendezvous control system. Note that the photo snapshots as well as the corresponding plots of the experiments were taken exclusively during the active phase of the rendezvous control system. In our experiments the interaction between the vehicles is realized through broadcast communication of UDP-packets. The static communication structure is simply hard-coded. That means each robot knows the IDs of its neighbors and the rendezvous controller exclusively uses their incoming data. Compared to the simulations, we are facing delays of up to 50ms in the communication links of the hardware system. However, due to the design of the controller, the robots significantly reduce their movement speed the closer they get together, and therefore these delays are negligible in our experiments. To avoid inter vehicle collisions, which first and foremost interfere with the dead reckoning based positioning as a result of skidding and slipping, we added basic collision avoidance techniques to the robots' system. Sensor based approaches for collision avoidance might be more sophisticated but since the main focus of the experiments lies on the rendezvous control system, the vehicles just use the position data received from the others to decide if another robot blocks the current direction of motion. If this is the case, the vehicle that detected the potential collision simply stops moving. Due to the possibility of mutual blocking in large groups of robots, this method might prevent the rendezvous from happening. However, in our experiments with four robots we did not observe such a situation.

Since a robot's speed input given by the control law can be quite large depending on the choice of c_1 as well as position and distance of its neighbors, we limited the absolute speed of each robot to 0.6m/s. It is easy to see from equation (6.21) that such a limitation does not influence the convergence behavior of the system. Due to the need for real-time responses and the limited computational power on the robot's microcontroller, the control laws had to be discretized with a fixed time interval in comparison to the continuous treatment in the theoretical examinations and the adaptively refined discrete treatment in the simulations. After a few tests we set the fixed time interval to 105ms, in order to cope with the resolution of the wheel encoders ($\approx$ 900ticks/m $\hat{=}$ 1.11mm) and the valid timer settings of the microcontroller.

Note that besides the scenarios described in what follows, we carried out additional experiments with different steering bounds and communication structures, where all runs provided the expected outcome.

6.7.1 Scenario 1

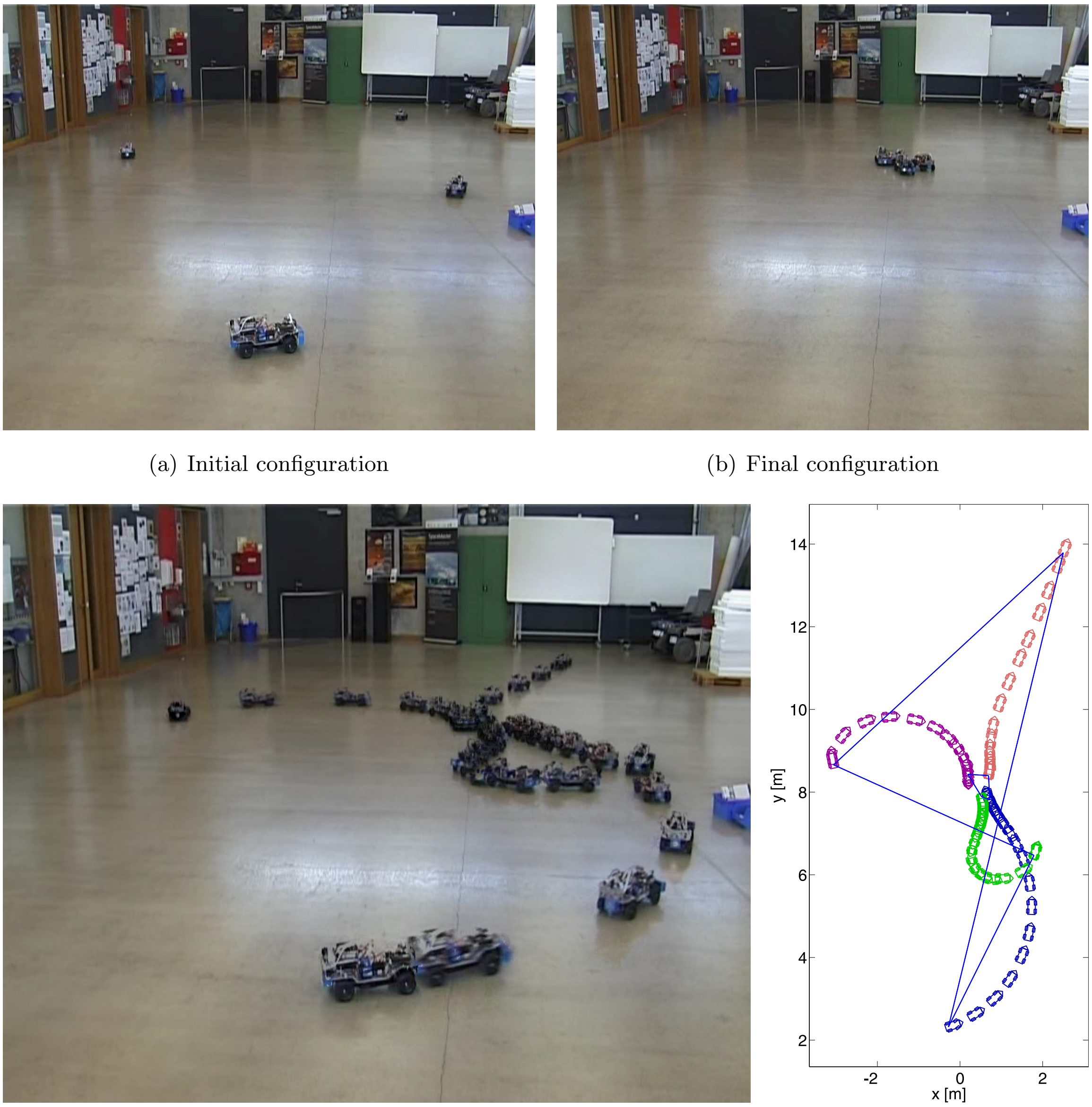
The first scenario shows the behavior of the rendezvous control system on a group of four randomly placed vehicles with their maximum absolute steering angle limited to 20°. The system's configuration before and after the application of the rendezvous control are depicted in Figure 6.12(a) and Figure 6.12(b), respectively. Figure 6.12(c) shows photographic snapshots of the experiment that were taken every two seconds and merged into one picture. Furthermore, with a frequency of 1Hz Figure 6.12(d) depicts the dead reckoning data of the robots which refers to the positions and orientations where the robots "think" they are located. The lines between the robots in the initial as well as in the final configuration denote the communication links, which form a ring structure in this scenario. The corresponding speed and steering profiles as well as the development of the V-function and $V_{i,\text{local}}$ are depicted in Figure 6.13. Like in the snow shoveling hardware experiments in Section 5.6 the speed values were measured by the robots wheel encoders and therefore they are quite accurate while the steering values are just a linear approximation from the PWM signal currently applied to the steering servo. Also, the steering values' accuracy is further decreased by a tolerance of about 4° in the wheel suspension. Fortunately the dead reckoning does not rely on the steering angle values, since changes of the orientation are directly obtained from a gyroscope, whose measurements are quite accurate in comparison.

As expected from the theoretical results and the simulations, the vehicles converge towards a common position. The V-function tends towards zero and the rendezvous control system stops after finite time. In the snapshots in Figure 6.12 it can be observed that the vehicle starting at $(x_o, y_o) = (1.79, 6.47)$ initially drives away from the final rendezvous point, which results from the fact that this robot has no information from the one starting on the top right of the configuration space. The slight overshoots ($|v_i| > 0.6\text{m/s}$) in the speed profile result from the behavior of the integrated linear speed controller, which was tuned to react quite aggressively, in order to quickly reach the values demanded by the rendezvous control system. By comparing the final vehicle positions of the photographic snapshot in Figure 6.12(b) with those of the dead reckoning data in Figure 6.12(d) it can be seen that they do not exactly concur with each other, especially for the vehicle that started in the upper right corner of the configuration space. This emphasizes the well known fact that positioning in longterm vehicle experiments should not just rely on dead reckoning.

6.7.2 Scenario 2

The second scenario was designed to compare the system's behavior for different bounds of the steering angle. Since it turned out to be difficult to observe orientation adjustment maneuvers in a random scenario with only four vehicles, for this scenario the robots have been arranged in a square with respect to their positions. Each robot is connected to the two closest neighbors such that the communication graph depicted in the positions diagram (Figure 6.14(d)) forms a square as well. In the following the results from two runs are described that provide the same initial configuration except for different steering bounds. For the run labeled scenario 2a, the steering bound $\phi_{\max}$ was set to 20° and for scenario 2b we set $\phi_{\max} = 10°$. Photographic snapshots as well as the positioning data obtained from the dead reckoning can be found in Figure 6.14 for scenario 2a and in Figure 6.16 for scenario 2b, respectively. Furthermore the corresponding speed and steering profiles as well as the development of the V-function and $V_{i,\text{local}}$ are depicted in Figure 6.15 for scenario 2a and in Figure 6.17 for scenario 2b.

In scenario 2a, the steering bound was chosen big enough such that there is no need for orientation adjustment maneuvers and the vehicles converge towards a common position by means of the closed-loop controller. Initially all four vehicles stand perpendicular to their momentary desired point and $\hat{e}_i > L/\tan\phi_{\max}$ for all of them. As a result all vehicles start driving forward, but since their initial orientations are not related in a symmetric fashion, the



(a) Initial configuration

(b) Final configuration

(c) Merged snapshots (0.5Hz)

(d) Dead reckoning (1Hz)

Figure 6.12: Rendezvous hardware experiment scenario 1 snapshots.

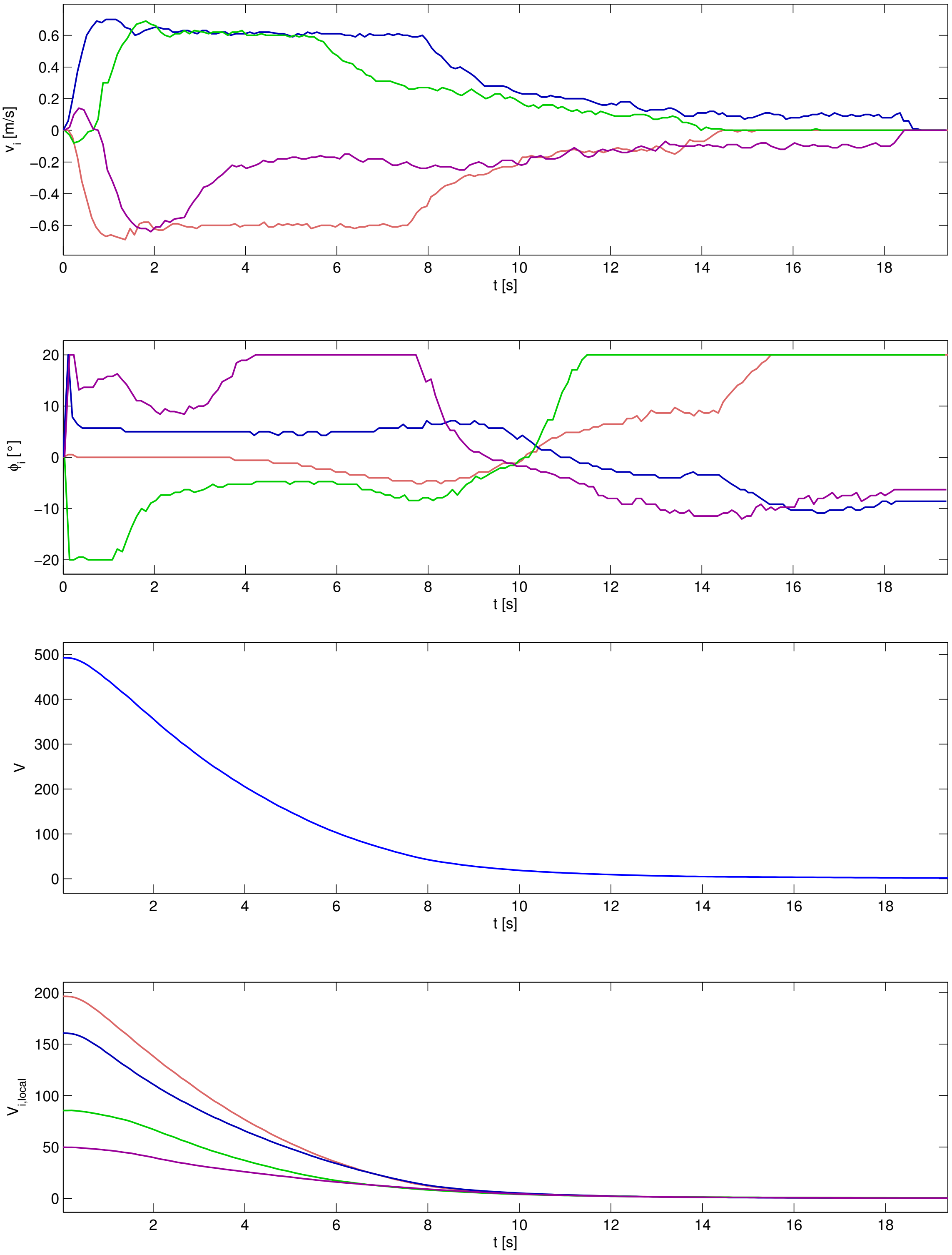
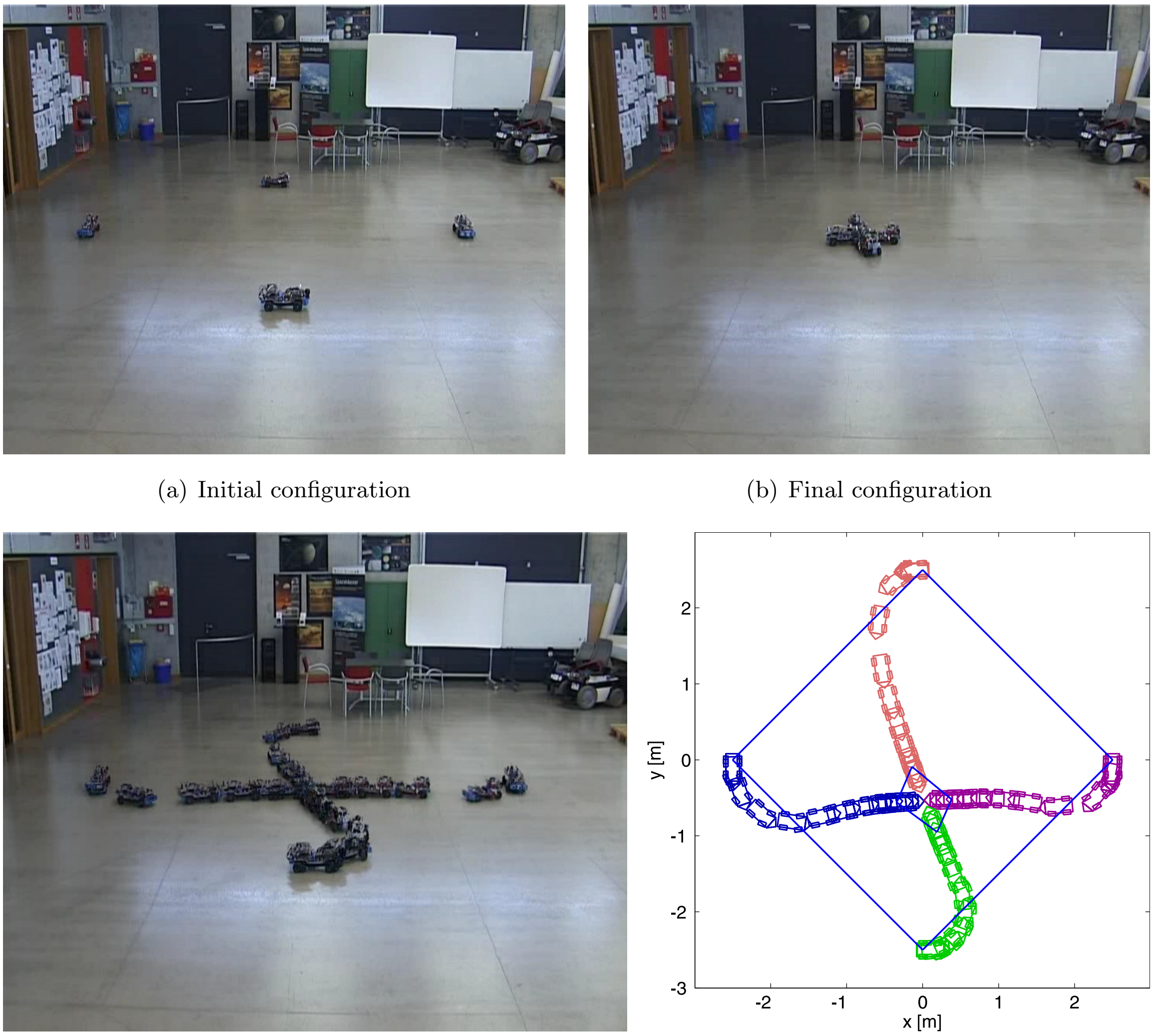


Figure 6.13: Rendezvous hardware experiment scenario 1 plots. The figure depicts the corresponding speed and steering profiles as well as the plots of V and $V_{i,\text{local}}$.



(a) Initial configuration

(b) Final configuration

(c) Merged snapshots (0.5Hz)

(d) Dead reckoning (1Hz)

Figure 6.14: Rendezvous hardware experiment scenario 2a snapshots.

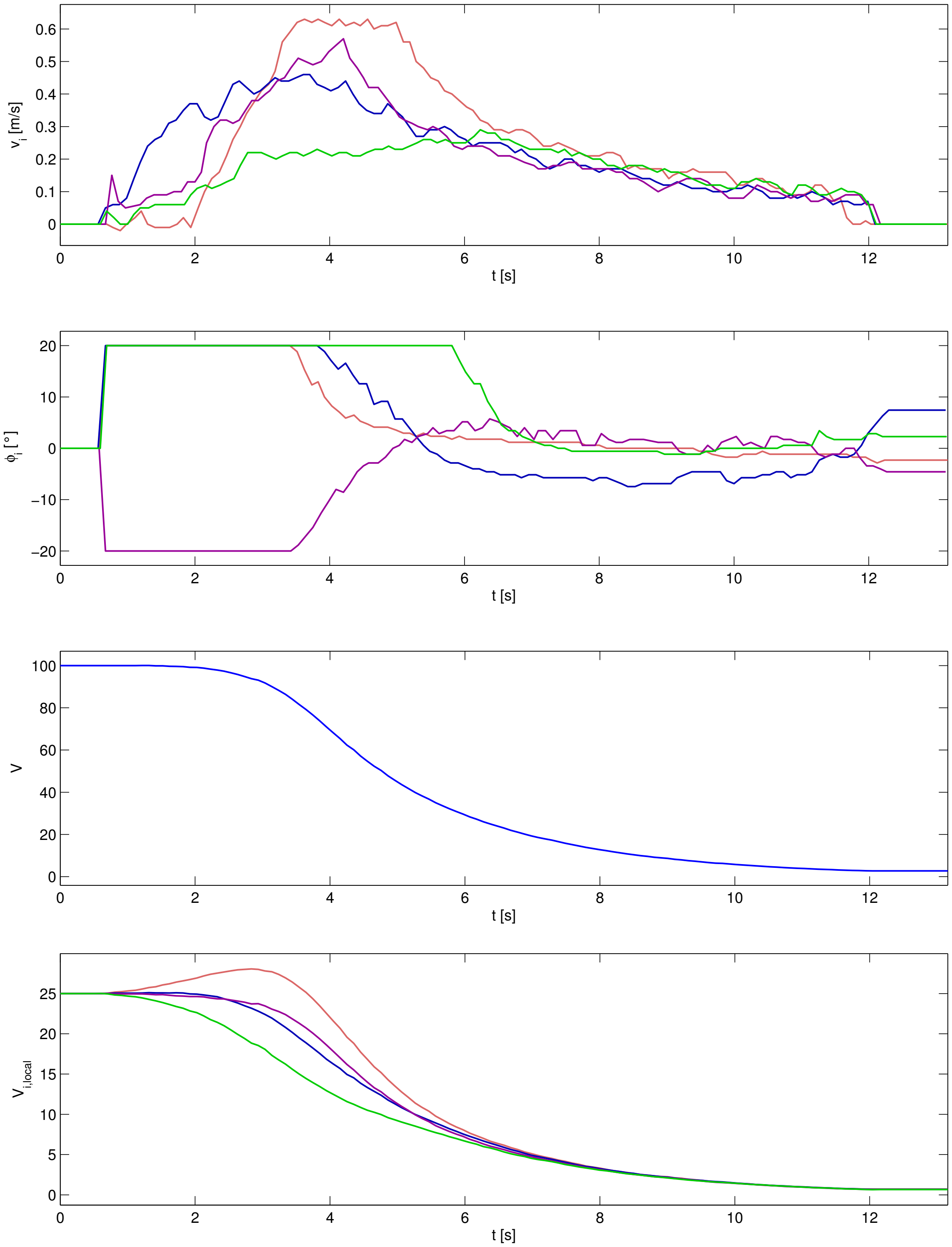
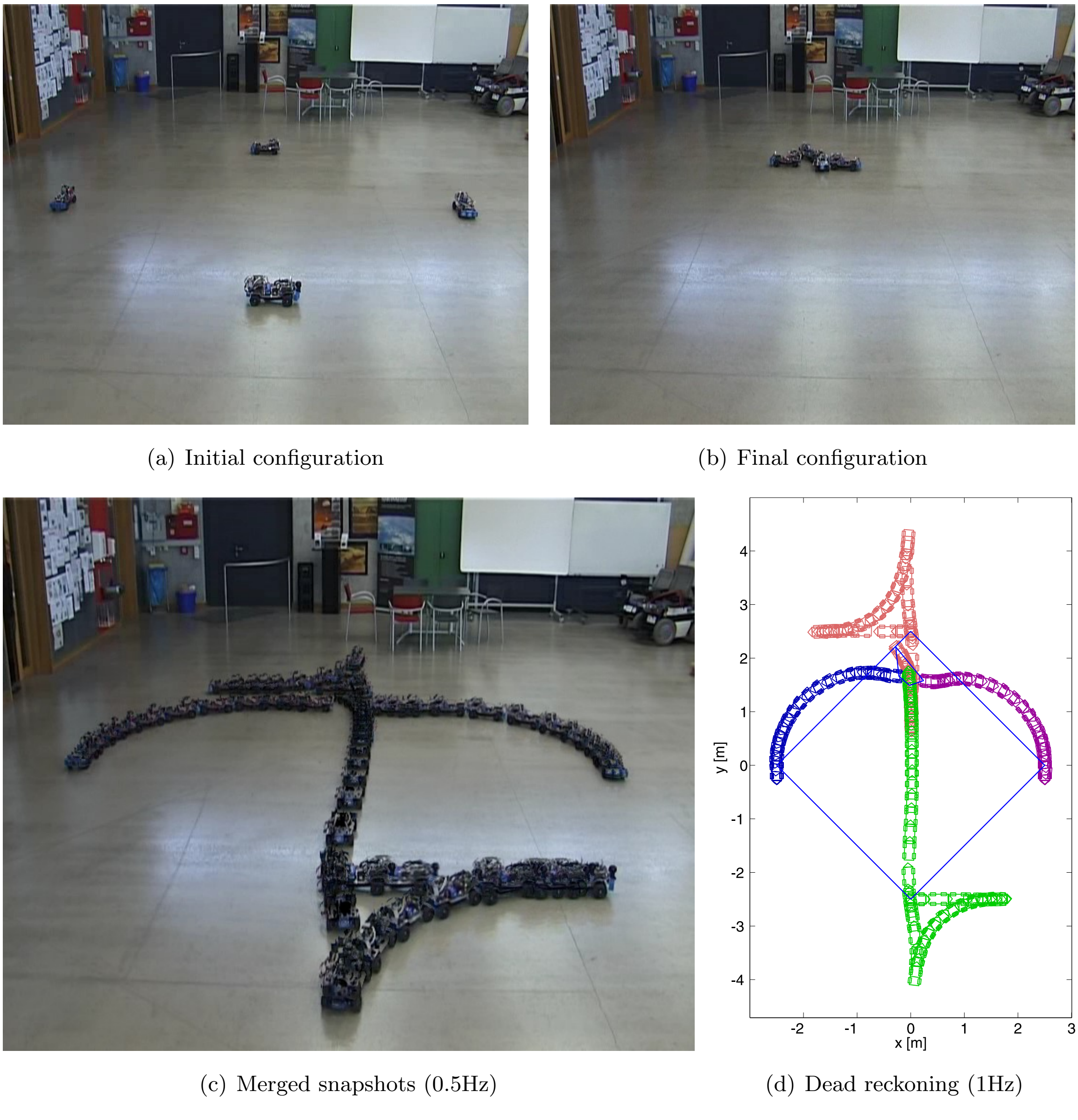


Figure 6.15: Rendezvous hardware experiment scenario 2a plots. The figure depicts the corresponding speed and steering profiles as well as the plots of V and $V_{i,\text{local}}$.



(a) Initial configuration

(b) Final configuration

(c) Merged snapshots (0.5Hz)

(d) Dead reckoning (1Hz)

Figure 6.16: Rendezvous hardware experiment scenario 2b snapshots.

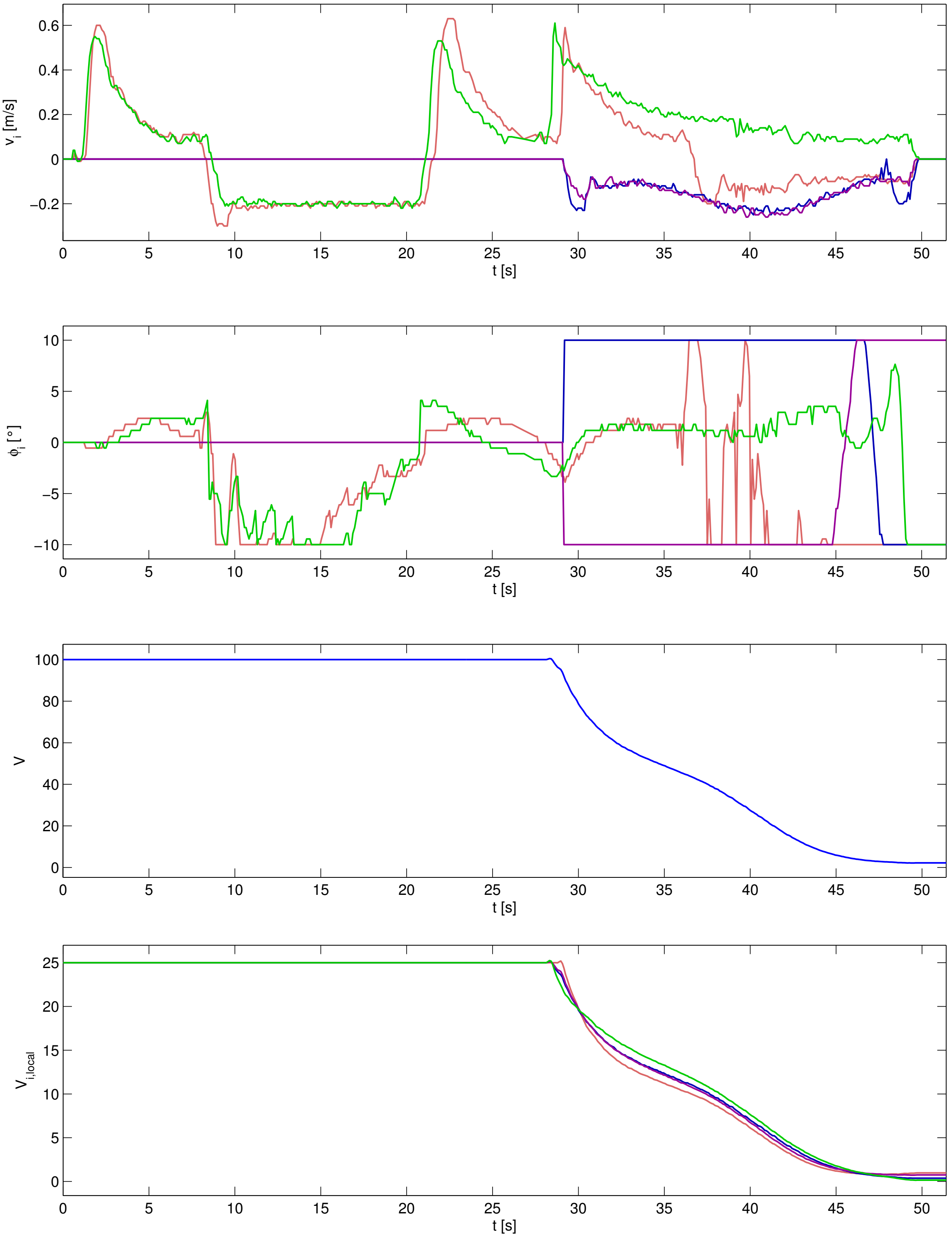


Figure 6.17: Rendezvous hardware experiment scenario 2b plots. The figure depicts the corresponding speed and steering profiles as well as the plots of V and $V_{i,\mathrm{local}}$.

rendezvous point is not in the center as one might expect, but shifted below. It should also be noted that even though $V_{i,\text{local}}$ initially increases for one robot, the value of the V-function is non-increasing all of the time.

In scenario 2b, the set steering bound necessitates orientation adjustment maneuvers at the beginning of the experiment. Initially $\hat{e}_i$ is the same for all vehicles, such that robot 1, the vehicle with the lowest ID and the opposing one (robot 2) start doing orientation adjustment maneuvers. In a simulation environment the other two vehicles would start maneuvers once the maneuvers of robot 1 and 2 have been completed. This is usually not the case in real environments due to existing uncertainties in the robot hardware like, e.g., from the tolerance in the wheel suspensions or from the inertia of the motors. As a result the maneuvers are not as accurate as in the simulation such that the vehicles need slightly different time intervals for the same maneuver as can be observed from the corresponding figures. Also note that if the vehicle starting from the bottom had a higher ID, it would have not fulfilled condition (C5) and robot 1 would have been the only maneuvering vehicle.

The robot that started from the bottom is the first that finishes the maneuver. Thus, it already moved a bit towards its desired point, when the opposing robot finishes its maneuver. As a result the desired point for the two robots located at the left and the right side is shifted from the global coordinate system's origin towards the top, which is also emphasized by the vehicles backward motion after their neighbors maneuvers have been completed. After the maneuvers of robot 1 and 2 a small increase of the V-function can be observed. This is due to the fact that because of existing uncertainties, the positions $p_i(t^o_{i,1})$ and $p_i(t^m_{i,1})$ for $i = 1, 2$ during the maneuvers are not exactly identical. Further it can be observed that the vehicle, which started from the top, moves backwards and deviates to the left at the end of the experimental run. Actually the vehicle detected a potential collision with the vehicle coming from the south and stopped moving. The opposing robot itself did not predict a collision and therefore pushed against its opponent until the local stopping criterion had been satisfied for all four vehicles.

6.8 Extension to concurrent maneuvers among neighboring vehicles

A major drawback of the rendezvous approach for the bounded steering case introduced in Section 6.5 lies in the design of the maneuver methodology. During a vehicle's orientation adjustment maneuver, all of its neighbors have to remain motionless. Especially for a system with dense interaction topology, the number of neighbors of a robot can be very high. In the worst case, in which the interaction graph is complete, it is not possible to do orientation adjustment maneuvers for more than one vehicle at a time.

Since the neighbors of a maneuvering vehicle are also close to an equilibrium position of the closed-loop controller, this leads to the assumption that the convergence time of the rendezvous in dense networks can be improved with a different methodology for the maneuver part of the approach that somehow allows neighbors to perform concurrent orientation adjustment maneuvers. In the remainder of this section we introduce such a modification and compare its performance to that of the rendezvous system for the bounded steering case under various aspects.

6.8.1 The extended approach

The main reason for the prohibition of concurrent maneuvers comes from the fact that the current desired position coordinates of a robot depend on the position of its neighbors. By keeping the neighbors' positions constant during the orientation adjustment maneuver we ensure that the desired position of the maneuvering robot is not moving, thus the robot will be oriented towards this point at time $t^m_{i,j}$ and also at the end of its maneuver.

To enable concurrent maneuvers among neighboring robots we modify the switching methodology introduced in Figure 6.3 and complement the set of maneuver conditions (C1)-(C5) stated in Subsection 6.5.2 with

(C6) $\exists j^* \in \arg\max_{j \in \mathcal{N}_i} \hat{e}_j, \exists k : t^o_{j^*,k} < t < t^o_{j^*,k} + K \cdot (t^m_{j^*,k} - t^o_{j^*,k})$

with constant $K \in [0, 1]$. With the extended switching methodology (Figure 6.18) it is now possible for waiting vehicles to start an orientation adjustment maneuver if they fulfill conditions (C2)-(C4) as well as condition (C6), which is satisfied if there exists a neighboring vehicle j^* with $\hat{e}_{j^*} = \max_{j \in \mathcal{N}_i} \hat{e}_j$ that is currently maneuvering and the start of the maneuver is no longer ago than $K \cdot (t^m_{j^*,k} - t^o_{j^*,k})$. Because of potential neighbors currently in the closed-loop control state it is possible for waiting robots that conditions (C2)-(C4) become true later during the maneuver of vehicle j^*. The variable K defines for how long it is allowed for neighbors of robot j^* to initiate a maneuver. For $K = 0$ the extended approach degrades to the one described in Section 6.5 since in this case, condition (C6) cannot be satisfied. Higher values for K can result in longer waiting times for robot j^* after the maneuver.

If all vehicles carry out a maneuver as defined by (M1)-(M4) in Subsection 6.5.2 we cannot guarantee that a vehicle is oriented towards its desired position when switching back to the closed-loop controller since neighboring robots might have maneuvered in the meantime and thus possibly caused a shifting of this desired point. To avoid a movement in the wrong

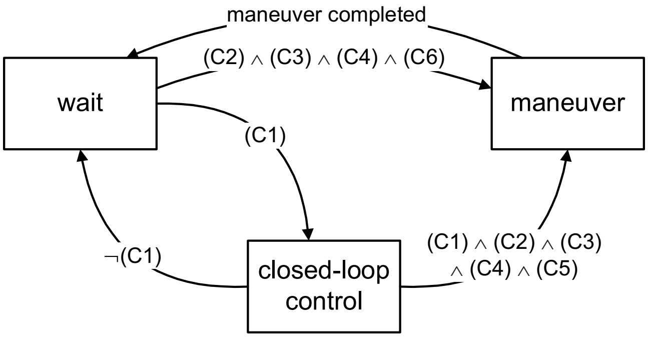


Figure 6.18: Automaton that describes the switching between controllers for the concurrent maneuver approach. Each robot in the waiting state that satisfies conditions (C2)-(C4) and (C6) will switch to the maneuver state while one or more of its neighbors are currently maneuvering. Robots that completed their maneuver switch to the waiting state until all maneuvering neighbors also finished with the orientation adjustment.

direction we require $c_3 = 0$ in maneuver property (M4) for all vehicles i that reached the maneuver state through the waiting state. Thus it is $p(t^f_{i,j}) = p(t^m_{i,j})$ for the final maneuver position of all vehicles that currently have a neighbor $j^* \in \mathcal{N}_i$ that satisfies maneuver condition (C5). In consequence these vehicles i usually complete their maneuvers earlier than their neighbor j^* with $c_3 > 0$ whose current desired point would be shifted if the robots i return to the closed-loop controller immediately after their maneuver. To keep the desired point of robot j^* constant we introduced a second modification in the switching scheme, which forces all vehicles that completed a maneuver to switch to the waiting state until every neighbor completed its maneuver. Accordingly we still can guarantee a lower-bounded decrease of the V-function during maneuvers for $V > 0$. This gives rise to the following Theorem:

Theorem 6.8.1. *Consider a connected system of N car-like robots that evolve according to* (6.1) *with constraint* (6.17)*. Then, by applying the control law* (6.18)-(6.19) *combined with the orientation-adjustment-maneuver extension as depicted in Figure 6.18, the position coordinates p_i of all vehicles converge towards a common value as $t \to \infty$.*

Proof. The proof follows the same arguments as in the case of exclusive maneuvers among neighboring vehicles (cf. proof of Theorem 6.5.5). □

Due to the complicated system behavior we are not able to make an analytical statement about the performance of the concurrent maneuver approach nor of the exclusive maneuver approach. Nevertheless, we carried out a series of simulations in order to compare the performance of the two approaches.

6.8.2 Performance evaluation of the two rendezvous systems

To compare the rendezvous performance of the exclusive maneuver approach (S_{em}) with the system that allows multiple concurrent maneuvers among neighbors (S_{mm}), a set of 1000 random scenarios for a group of ten robots with $L = 0.3$m was generated. The vehicles initial positions were randomly set within an area of $20\mathrm{m} \times 20\mathrm{m}$. The parameters for the closed-loop controller were fixed to $c_1 = 5$ and $c_2 = 10$. Orientation adjustment maneuvers were carried out like it was illustrated in the example depicted in Figure 6.2 while the amount of time needed for the maneuver was set to be equal to its path length in meters. This simulates an average speed of 1m/s during maneuvers. The constant in maneuver property (M4) was set to $c_3 = 0.2$ and the constant in condition (C6) of the extended approach S_{mm} was set to $K = 1/8$. When utilizing the local stopping criterion defined by (6.36), systems with a higher number of interaction links are disadvantaged since $V_{i,\mathrm{local}}$ depends on the number of neighbors. Thus, in order to achieve fair conditions, a simulation run was stopped once the distance between every pair of vehicles was less or equal to 0.5m no matter if they share an edge in the interaction graph or not. As the main performance criterion we evaluated the elapsed time from a scenario's start until the stopping criterion was fulfilled. Further the total traveled distance, and the total number of maneuvers (maneuver phases) each summed over all vehicles have been considered.

In the first series we kept the robots' steering bound constant at $\phi_{\mathrm{max}} = 10°$ and varied the average number of interaction links per vehicle from the minimal connected graph to the complete graph. The minimal interaction graph simply consists of nine undirected edges that connect robot 1 with robot 2, robot 2 with robot 3, etc. Thus every vehicle has two neighbors except for robot 1 and robot 10 with only one neighbor, which results in an average of 1.9 interaction links per vehicle. A denser interaction graph was achieved by randomly adding new interaction links up to the complete graph with an average of nine interaction links per vehicle. Altogether 30000 simulation runs were carried out for this series.

In the second series the interaction structure was chosen to be the smallest proximity graph that is still connected at the initial configuration. The resulting mean number of interaction links per vehicle averaged over all 1000 scenarios was 3.4. This time we varied the vehicles steering bound such that each of the generated scenarios was evaluated with 21 different values for ϕ_{max} ranging from 5° to 25° resulting in a total of 50000 simulation runs.

The results of the simulations are depicted in the plots in Figs. 6.19, 6.20 and 6.21. Figure 6.19 shows time, total traveled distance as well as the total number of maneuvers (maneuver phases) averaged over all 1000 scenarios and plotted against the average number of interaction links and the steering bound ϕ_{max} respectively. In Figs. 6.19(c) and 6.19(f) the solid curves illustrate the total number of maneuvers while the dotted curves represent the total number of maneuver phases. A new maneuver phase is initiated when a robot starts a maneuver and in that moment there exists no other robot in the group that started a maneuver no longer ago than $K \cdot (t^{m}_{j^*,k} - t^{o}_{j^*,k})$.

Since the depicted values are averaged over all scenarios, these plots are not suitable for the comparison of the two rendezvous systems. Their main intension is to depict the effect of different values for the varying parameter (av. comm. links, $\phi_{\max}$). The comparison of S_{em} and S_{mm} was done on a per scenario basis and the results for time, traveled distance and number of maneuvers are depicted with respect to the average number of interaction links per vehicle in Figure 6.20 and with respect to the maximum allowed steering in Figure 6.21. In both figures, plots (a)-(c) illustrate the number of scenarios in which each of the approaches dominated (performed better) as well as the ones with identical outcome. On the contrary, plots (d)-(f) compare the two approaches on a percentage basis. For each simulation run it was calculated by how many percent the dominating approach outperforms the other one, where we used a negative percentage when S_{em} performed better and a positive value when it was the other way round. Now the bars in the plots correspond to the average of these percent values, which denote the mean performance improvement of one system over the other. The negative and the positive standard deviation is represented as intervals placed on top of the bar plot.

Figure 6.19(a) shows that the time needed by S_{mm} averaged over all scenarios hyperbolically decreases with an increasing number of interaction links in the system. Compared to this S_{em} decreases in a similar way until the average number of interaction links per vehicle is approximately equal to three, which is one third of the average interaction links per node in the complete graph. The reason for the decrease in time can be found in the fact that the distribution of information is faster in systems with a denser interaction graph. For more than three average interaction links per vehicle the average time needed for the rendezvous slightly increases with additional links. This is due to the higher number of neighbors per vehicle in systems with a denser interaction graph. When a robot does an orientation adjustment maneuver, all of its neighbors switch to the waiting state. As a consequence, there are more vehicles that cannot decrease V by means of the closed-loop control nor by performing a maneuver themselves. This also can be observed in Figure 6.19(c), where the number of total maneuvers decreases while the number of maneuver phases increases with a higher number of interaction links in the system, which means there are less concurrent maneuvers during the rendezvous process. Note that for the complete graph the number of maneuver phases

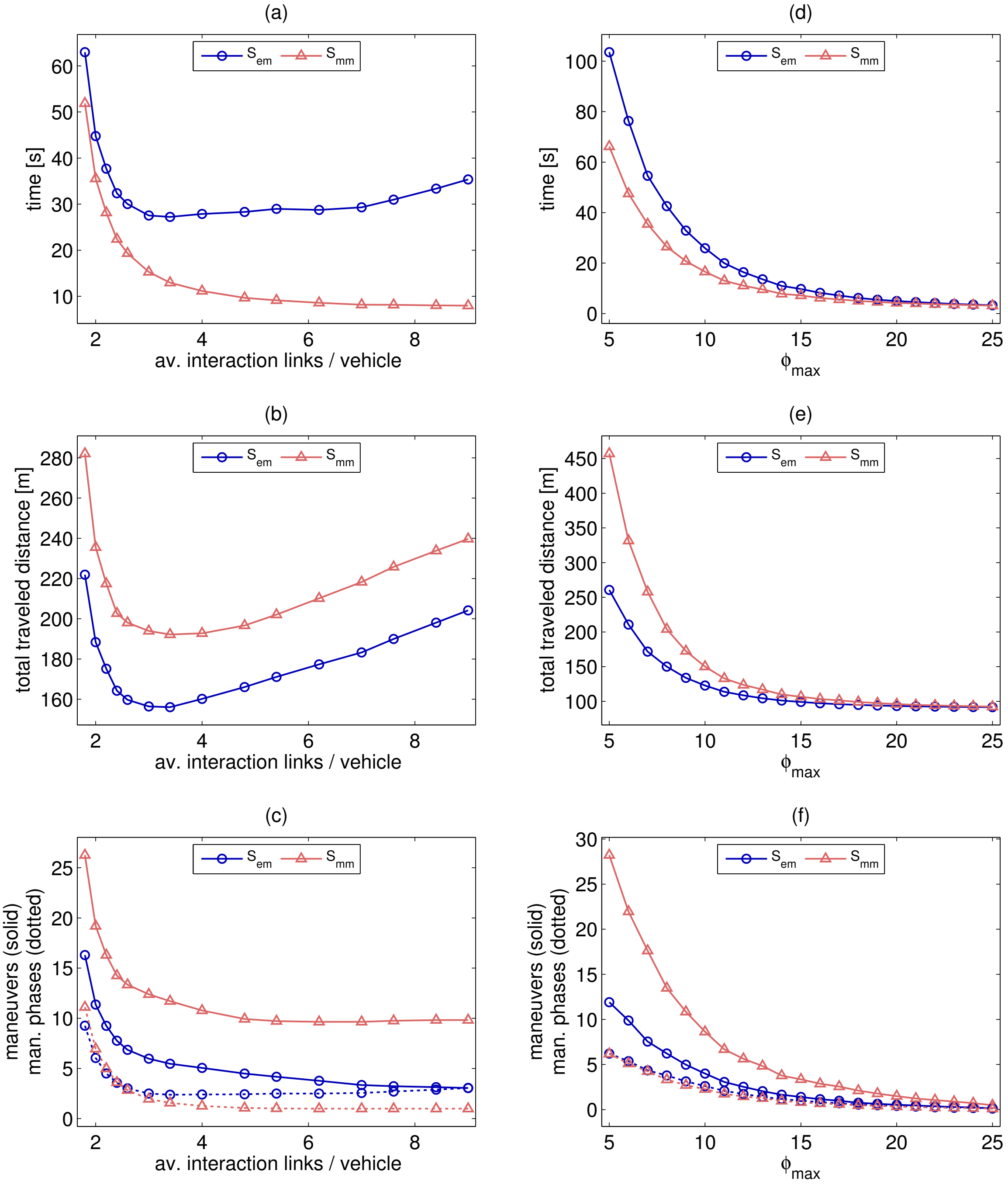


Figure 6.19: Evaluation of S_{em} and S_{mm}: Time, total traveled distance and number of maneuvers (maneuver phases) averaged over 1000 simulated scenarios and plotted against the average number of interaction links and the steering bound $\phi_{\max}$ respectively.

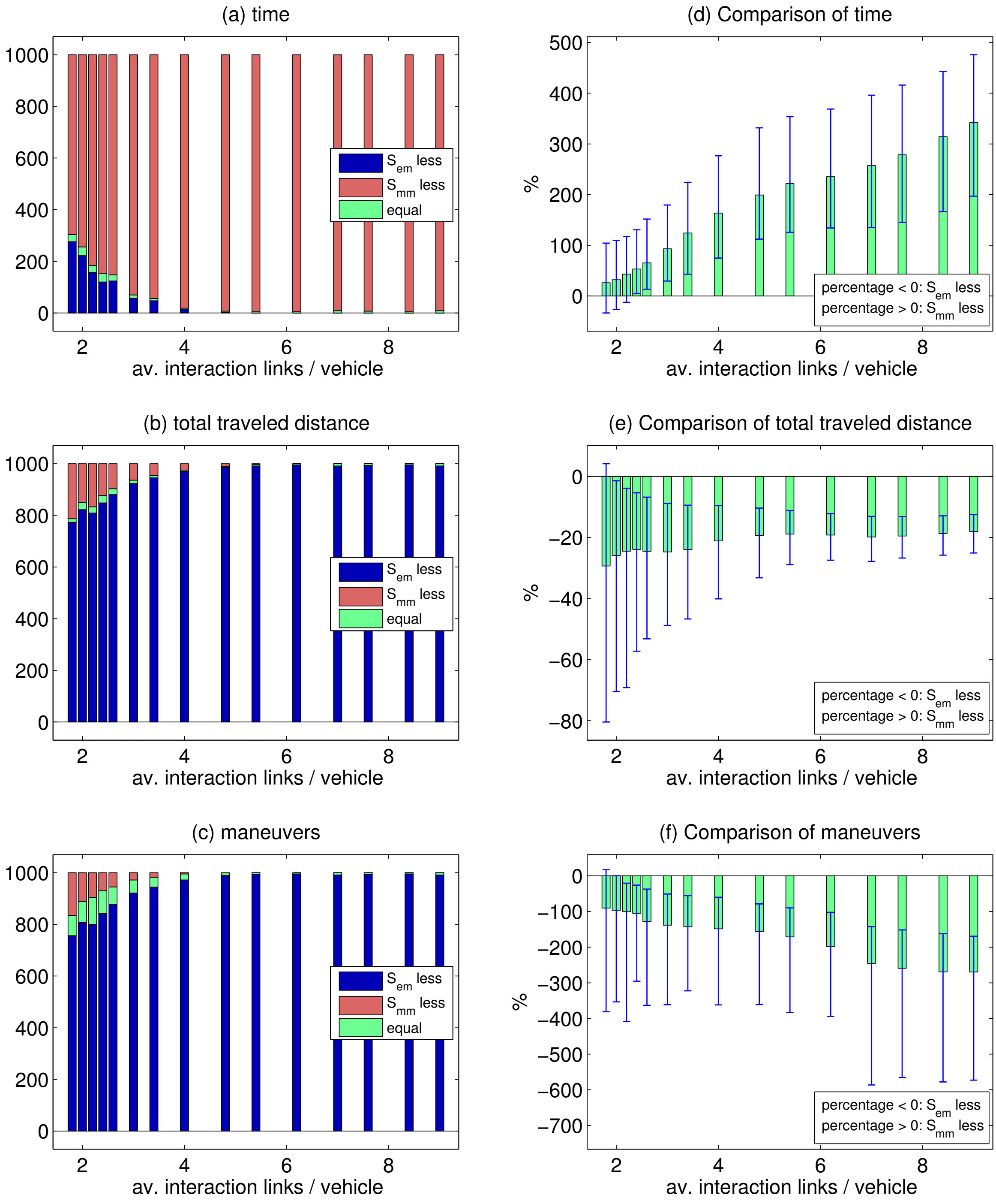


Figure 6.20: Performance evaluation of S_{em} and S_{mm} concerning time, traveled distance and number of maneuvers for different interaction topologies. Plots (a)-(c) illustrate the number of scenarios in which each of the methods performed better as well as the ones with identical outcome. Plots (d)-(f) compare the two methods on a percentage basis.

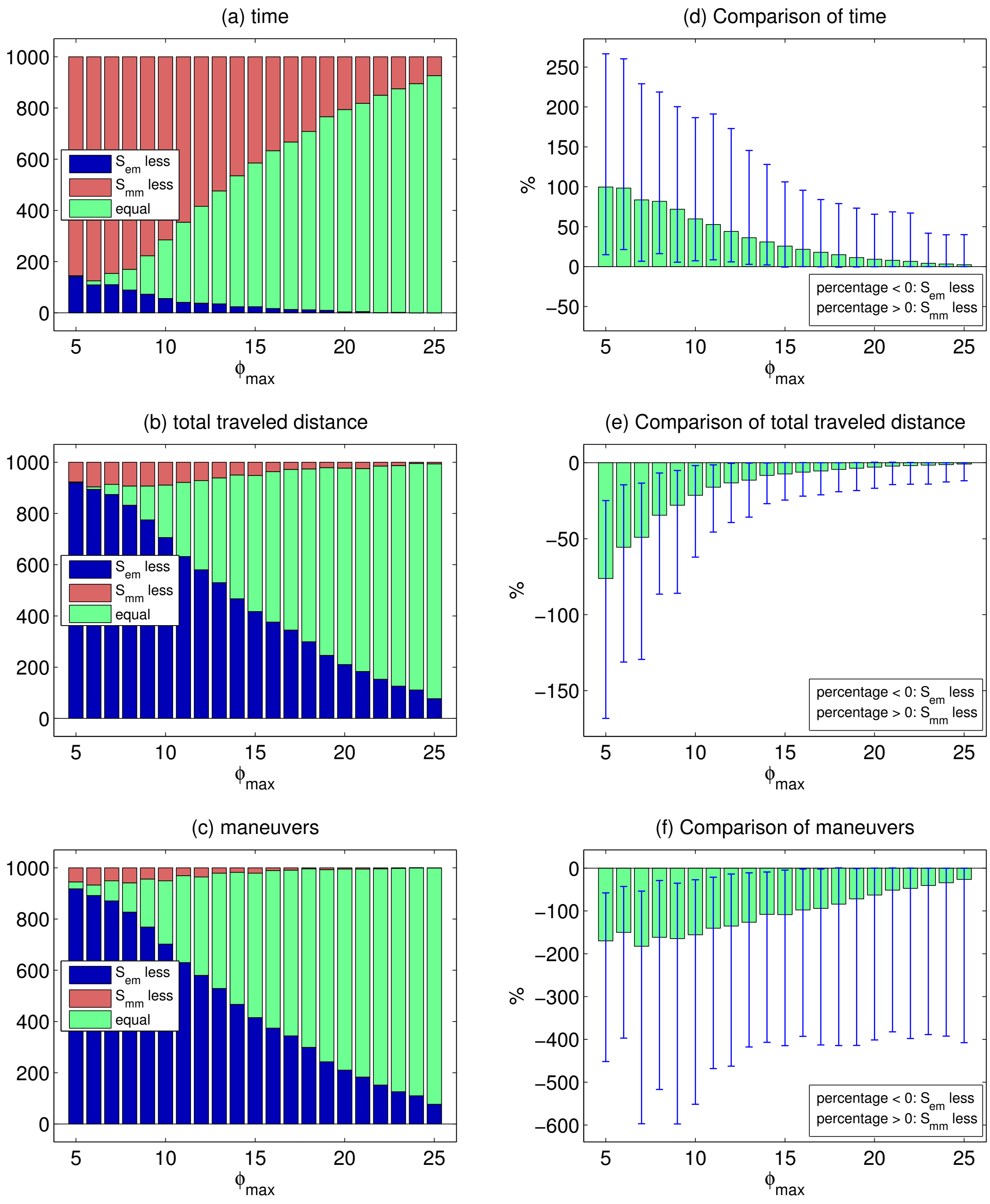


Figure 6.21: Performance evaluation of S_{em} and S_{mm} concerning time, traveled distance and number of maneuvers for different steering bounds. Plots (a)-(c) illustrate the number of scenarios in which each of the methods performed better as well as the ones with identical outcome. Plots (d)-(f) compare the two methods on a percentage basis.

is equal to the number of total maneuvers, since if one vehicle maneuvers all others have to wait.

From Figure 6.19(b) it can be seen that the average total traveled distance decreases for both rendezvous systems for an increasing number of interaction links until it reaches an average value of approximately three. With further interaction links the average total traveled distance increases linearly for both approaches. In contrast to this, the number of total maneuvers shows a hyperbolic decrease with an increasing number of interaction links for both methods throughout the whole domain as can be seen from Figure 6.19(c). Therefore we can conclude that the linear increase has to result from the closed-loop controller. In fact the value of $\hat{v}_i$ increases with the number of distant neighbors and as a result also the input velocity v_i is increased as can be seen from equation (6.18). Since the rate of turning $\dot{\theta}_i = \sigma_{\phi_{\max}}(c_2\omega_i)$ is not affected, the resulting path to the momentary desired position is longer in the case of $\omega_i \neq 0$, which holds whenever robot i is not oriented towards this point. Figure 6.19(c) also shows that compared to the decreasing and than increasing number of maneuver phases for S_{em}, the average number of maneuver phases for S_{mm} continuously decreases with the number of interaction links in the system. This results from the possibility to perform concurrent maneuvers among neighbors with S_{mm}, while it is prohibited in S_{em}.

It can be easily comprehended that because of the degraded mobility, feasible short paths to a desired position become infeasible for vehicles with a lower steering bound in many cases. Thus it is not surprising that with an increasing value for the maximum steering angle the needed time as well as the traveled distance decrease hyperbolically for both approaches as can be observed in Figure 6.19(d) and Figure 6.19(e). A decrease with growing steering bound is also observed for the total number of maneuvers and maneuver phases (Figure 6.19(f)). In fact, the amount of maneuvers will converge to zero as $\phi_{\max} \to \pi/2$ or equivalently as the system approaches the free steering case that was considered in Section 6.4.

Figure 6.20 illustrates the results from comparing the two approaches on a per-scenario basis with respect to the average number of interaction links per vehicle. Looking at Figure 6.20(a) it can be seen that S_{mm} dominates most of the scenarios with respect to time, while the corresponding diagram concerning the total traveled distance depicted in Figure 6.20(b) is nearly inverted. The amount of dominated scenarios with respect to the number of maneuvers (Figure 6.20(c)) is closely related to the amount with respect to the traveled distance aside from a few additional scenarios with equal outcome. The average improvement in needed time of S_{mm} versus S_{em} continuously increases from 26.0% for the minimal connected interaction graph to 341.9% for the fully connected case (cf. Figure 6.20(d)). Concerning the traveled distance, S_{em} needs on average 22.0% less than S_{mm} without major changes with respect to the average number of interaction links per vehicle. However, the standard deviation, which significantly decreases with the number of interaction links, indicates that the outcome varies more in systems with sparse interaction topology. As Figure 6.20(f) indicates, the average number of maneuvers is less for S_{em} for various interaction topologies. The improvement increases from 90.3% at 1.9 average interaction links per vehicle to 269.8% for the complete interaction graph that has nine interaction links per vehicle. The high standard deviation implies that the improvement varies a lot from case to case. Its higher negative value indicates a lower number of cases for which the improvement with S_{em} is much better than the average value. Compared to this there are more cases for which the improvement is less or even negative in some of the runs with less than five average interaction links per vehicle.

Results from comparing the two approaches on a per-scenario basis with respect to the steering angle bound $\phi_{\max}$ are depicted in Figure 6.21. It can be observed from Figs. 6.21(a)-6.21(c) that S_{mm} dominates with respect to time and S_{em} dominates with respect to total traveled distance and the number of maneuvers for low steering bounds. The distribution in the corresponding plots is quite similar. With a more relaxed steering bound the number of simulation runs with identical outcome increases significantly. In most of these runs no maneuvers were carried out since the improved mobility of an increased steering bound enabled the robots to fulfill the stopping criterion exclusively by means of the closed-loop controller. This is also the main reason why the average improvement with respect to time, total traveled distance and maneuvers decreases while the steering bound is increased (Figs. 6.21(d)-6.21(f)). Also it can be observed in all three plots that the standard deviation in direction of the dominating controller is significantly bigger than in the opposite direction.

We conclude that with respect to time needed for the rendezvous, S_{mm} is the better choice in all cases. But if energy consumption is an issue and needed time is less important, than S_{em} with a lower mean traveled distance will perform better on an average given the utilized vehicles have a huge turning radius.

6.9 Summary and discussion

In this chapter we proposed solutions to the rendezvous problem for groups of car-like vehicles for the cases of free and bounded steering that have not yet been addressed in literature. For the free steering case we introduced a discontinuous feedback controller, whose convergence was proven by applying the partial stability theorems from Chapter 2. For the bounded steering case the controller was slightly modified and complemented with a switching logic which commands certain vehicles to perform orientation adjustment maneuvers that drive the system out of undesired equilibria. After proving convergence for the complete system we discussed the problem of distributed termination of the rendezvous process that is unsolvable without utilizing some kind of broadcasting strategy. Both approaches have been verified in simulations, while the developed system for the bounded steering case has also been demonstrated in hardware experiments. Finally, the bounded steering approach was extended in order to allow concurrent orientation adjustment maneuvers among neighboring vehicles. The resulting decrease of the convergence time at the expense of a higher overall traveled distance has been verified statistically using the data from extensive simulations.

To show the convergence of the rendezvous control systems we relied on the concept of Filippov solutions, which are guaranteed to exist even for dynamical systems with discontinuous right-hand side. Since the feedback control law is discontinuous, situations might occur in which a part of the system state slides along a discontinuous surface. This can happen to a vehicle's position when its neighbors move in such a way that for the time-derivative of the condition classifying the discontinuous surface we have $\dot{v}_i = 0$. One might argue that the notion of sample-and-hold solutions (cf., e.g., [Cor08]) is more suitable for our specific system, but it is not difficult to find situations in which the proposed control system for the bounded steering case results in a deadlock situation. Further it should be noted that concerning the bounded steering case, Theorem 6.5.5 also holds without the assumption that $v_i \neq 0$ at unstable weak equilibria. At such an equilibrium all vehicles satisfy the maneuver condition (C4) and if none of them is moving, there will be at least one vehicle doing an orientation adjustment maneuver that drives the system away from the discontinuous surface.

The design of the feedback control laws has to be done carefully. It is essential for the equation computing v_i in the free steering case (6.5) as well as in the bounded steering case (6.18) that we divide by $2|N_i|$ instead of $|N_i|$. Otherwise the estimates (6.16) and (6.22) do not hold and it is possible to construct a non-converging example like in Appendix C simply by placing two vehicles with different orientation on opposing tangents of a circle, whose radius does not violate the vehicles' steering bound. With the developed controller for vehicles with bounded steering it might also happen that the maneuver conditions (C2) and (C3) are satisfied for all nodes even directly after a maneuver depending on the value of ε_1. As a consequence, this could result in an increased number of orientation adjustment maneuvers; however, the convergence properties are not affected. But since in real world applications the convergence to a certain region is desired, this situation will not occur given the constant ε_1 was chosen small enough.

In special cases the convergence to the desired equilibrium from the same initial state can be faster even though the steering bound is more restricted. Because of the distributed nature of the system it is very difficult or even impossible for individual nodes to identify such situations. The average convergence time might be significantly improved if vehicles would be able to detect and intelligently react on them. Using a dedicated leader unit to command the gathering also might result in faster convergence due to the possibility to calculate an optimal meeting point. But the fact that the leader is a single point of failure for the whole system justifies the need for a decentralized approach, which provides robustness and scalability to the system. To decrease the communication effort within the system, position and speed of neighboring vehicles might be obtained via acoustic or optical range measurements or by a vision system. Also the transmission of $\hat{e}_i$ to the neighbors in the bounded steering case can be limited to the time intervals when the "local" conditions (C2) and (C3) are satisfied.

7 Conclusion and future work

In this monograph we discussed different problems concerning the coordination and control of motion in systems of autonomous wheeled mobile robots. As most of the related work in literature does not cover car-like robots, there exists a certain demand for motion planning approaches and control systems that are applicable to groups of these kind of vehicles. Therefore we attached great importance to develop approaches that fulfill this criterion.

We reviewed a motion planning approach for formations of nonholonomic mobile robots that is also applicable for vehicles of the car-like type. A closer look at the approach revealed several weak points that significantly lower its applicability. In order to close the identified gaps we introduced different enhancements to the original motion planning approach. In particular, in the original approach, an unconstant reference speed results in an undesired modification of the formation's shape. This problem could be solved by simply synchronizing the vehicles' speeds. Furthermore, we derived an extended equation for the curvature input calculation of the follower robots. This modification now allows changes of the formation's shape along trajectory segments with continuously differentiable curvature in comparison to the need for constant curvature in the original approach, which in practical terms has been a huge disadvantage. Moreover, the originally proposed function for requested modifications of the formation's shape created undesired jumps in the speed and curvature profiles of the follower vehicles. This problem could be solved by introducing higher order polynomial spline functions for the relative position coordinates' alteration. Provided that the given reference speed is at least C^0-continuous and the reference curvature is of class C^1 or smoother, their application results in guaranteed continuous speed and steering profiles for the follower vehicles.

The considered motion planning approach generates values for the vehicles' control inputs based on the speed and curvature values of a reference trajectory. Since closed-loop trajectory tracking requires the absolute position error and the corresponding derivatives one might integrate the kinematic model numerically. An alternative solution is to do the planning directly in absolute coordinates as it was shown in the context of the task of snow shoveling on airfields via formations of autonomous snowplows.

For the snow shoveling task a centrally supervised system architecture was designed with a leader-follower hierarchy on the level of motion coordination. The motion coordination was divided into the subproblems of formation motion planning and tracking of the generated trajectories. For the motion planning part we introduced a trajectory planner that transforms the output of the Task Allocation module into a trajectory for the reference point of the corresponding Formation unit. Based on this reference trajectory each vehicle computes its own desired trajectory. In order to achieve a robust tracking of the planned trajectory, we derived a controller that is based on feedback linearization. Since the shovel's mounting point was chosen as the tracking's reference, the controller efficiently handles sharp turns by achieving optimal breadthwise coverage of the road. Furthermore, the required differentiability class for error free tracking is reduced to C^2 compared to the need for C^3-continuity if the tracking reference is located in the center of the snowplows' rear-axle. In a simulation environment we showed that the developed snow shoveling framework is also able to react on unforeseen changes in the environment such as narrow road segments, obstacles and blocked roads. Besides that, we demonstrated the robustness of the motion coordination solutions in laboratory hardware experiments.

Another problem that was dealt with in this work is concerned with the rendezvous problem for groups of car-like vehicles. Besides the free steering case, which allows arbitrary steering angles below 90°, we considered also the more realistic case of vehicles with bounded steering angle. For the free steering case we introduced a discontinuous feedback controller that depends exclusively on the robot's own state and the position of other vehicles in range. Convergence of the closed-loop system's position variables has been proven by applying versions of Lyapunov's direct method and LaSalle's invariance principle that were generalized to discontinuous systems and reformulated in order to cope with partial state stability. For the bounded steering case the developed controller was slightly modified and complemented with a switching logic that commands certain vehicles to perform orientation adjustment maneuvers that drive the system out of undesired equilibria. After the proof of convergence for the complete system, the problem of distributed termination of the rendezvous process was discussed. It turned out that a solution relying exclusively on already available information is impossible for the proposed control system, such that some kind of broadcasting strategy needs to be implemented. Then we verified both approaches in simulations, while the developed system for the bounded steering case has also been evaluated in hardware experiments with four robots. Finally, the bounded steering approach has been extended such that it allows concurrent orientation adjustment maneuvers among neighboring vehicles. Based on the data from extensive simulation runs we carried out a statistical analysis which showed that the extended version provides faster convergence at the expense of a higher overall traveled distance.

The work in this dissertation could be extended in several aspects. In the snow shoveling application for example, we sometimes face the situation that the intersections are not swept adequately as snow remains in the unswept bends or the center. Finding an appropriate solution to this problem is not obvious as a formation could somehow sweep the intersection completely at its first arrival (low-level solution) or the crossing could be visited multiple times during the cleaning process (high-level solution).

Another open problem lies in the fact that the developed rendezvous control systems lack the ability to deal with a limited interaction range. The control laws could be extended in such a way that interaction links are preserved once they are established. One possible solution might be the application of artificial potentials as it has been proposed, e.g., in [ZP07, JE07].

Along the lines of related work on rendezvous and consensus for simpler dynamical systems, the proposed controllers could be extended to directed and time-varying interaction graphs. Further, it might be possible to introduce procedures for collision avoidance such that the no-collision assumption can be omitted (cf., e.g., [SHSŠ07]).

In real world networked systems, one faces packet loss, delay, jitter, etc. Instead of using a simplified model of a perfect interaction topology one could take into account more realistic models which also consider the imperfection of the individual interaction links. For sure this step would significantly increase the difficulty of formulating a convergence proof for an already complicated control system.

Appendices

A Solution to the linear equation system from Subsection 4.3.3

Here, the solution to the linear equation system defined by equations (4.24)-(4.41) in Subsection 4.3.3 is provided. The received values correspond to the coefficients of (4.23) which describes the demanded spline.

By determining that $S_1 = s_o + (s_f - s_o)/4$ and $S_2 = s_o + 3(s_f - s_o)/4$, the solution for the equation system is given by

$$
\begin{aligned}
a_1 &= 0,\\
b_1 &= -\frac{40(q_o - q_f)}{57(S_1 - S_2)^4},\\
c_1 &= \frac{80(3S_1 - S_2)(q_o - q_f)}{57(S_1 - S_2)^4},\\
d_1 &= -\frac{20(S_2 - 3S_1)^2(q_o - q_f)}{19(S_1 - S_2)^4},\\
e_1 &= \frac{20(3S_1 - S_2)^3(q_o - q_f)}{57(S_1 - S_2)^4},\\
f_1 &= \frac{5(S_2 - 3S_1)^4 q_f - (291S_1^4 - 84S_1^3S_2 - 414S_1^2S_2^2 + 396S_1S_2^3 - 109S_2^4)q_o}{114(S_1 - S_2)^4},\\
\\
a_2 &= \frac{24(q_o - q_f)}{19(S_1 - S_2)^5},\\
b_2 &= -\frac{60(S_1 + S_2)(q_o - q_f)}{19(S_1 - S_2)^5},\\
c_2 &= \frac{80(S_1^2 + 7S_1S_2 + S_2^2)(q_o - q_f)}{57(S_1 - S_2)^5},\\
d_2 &= \frac{20(S_1 + S_2)(S_1^2 - 8S_1S_2 + S_2^2)(q_o - q_f)}{19(S_1 - S_2)^5},\\
e_2 &= \frac{20(S_1^4 - 10S_1^3S_2 + 36S_1^2S_2^2 - 10S_1S_2^3 + S_2^4)(q_o - q_f)}{57(S_1 - S_2)^5},\\
f_2 &= \frac{S_1^5(5q_o + 109q_f) - S_2^5(109q_o + 5q_f) - 30S_1^2S_2^3(27q_o + 11q_f)}{114(S_1 - S_2)^5}\\
&\quad + \frac{5S_1S_2^4(101q_o + 13q_f) + 30S_1^3S_2^2(11q_o + 27q_f) - 5S_1^4S_2(13q_o + 101q_f)}{114(S_1 - S_2)^5},
\end{aligned}
$$

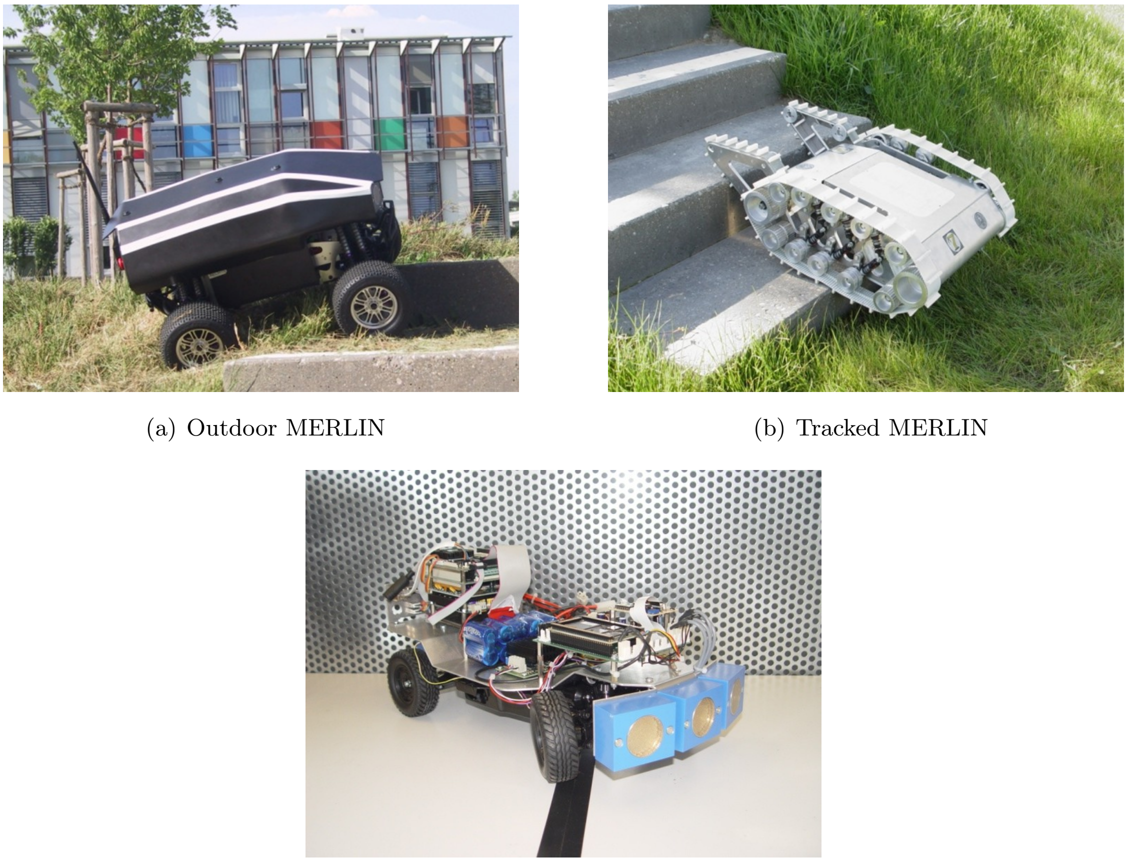
$$
\begin{aligned}
a_3 &= 0,\\
b_3 &= \frac{40(q_o - q_f)}{57(S_1 - S_2)^4},\\
c_3 &= \frac{80(S_1 - 3S_2)(q_o - q_f)}{57(S_1 - S_2)^4},\\
d_3 &= \frac{20(S_1 - 3S_2)^2(q_o - q_f)}{19(S_1 - S_2)^4},\\
e_3 &= \frac{20(S_1 - 3S_2)^3(q_o - q_f)}{57(S_1 - S_2)^4},\\
f_3 &= \frac{5(S_1 - 3S_2)^4 q_o + (109S_1^4 - 396S_1^3S_2 + 414S_1^2S_2^2 + 84S_1S_2^3 - 291S_2^4)q_f}{114(S_1 - S_2)^4}.
\end{aligned}
$$

B The MERLIN vehicles

The MERLIN (Mobile Experimental Robot for Locomotion and Intelligent Navigation) vehicles are a collection of mobile robots based on different locomotion platforms, whose development started in the mid 90's in the context of the European Mars rover development effort MIDD [SRB$^+$97]. Even though the kind of movement is different for each vehicle, they share an identical setup of electronic components for control and sensor processing. Keeping in mind that the performance of a mobile robot depends mainly on suitable sensors to characterize the environment and the actual position, the first generation of MERLINs was designed as test vehicles that allow flexibility and easy access for the evaluation of different sensor configurations [SM02].

The Outdoor MERLIN (cf. Figure B.1(a)) is a car-like robot that was designed for harsh outdoor environments and various weather conditions. It is usually equipped with a vision system, GPS, gyroscope, and an array of ultrasonic sensors surrounding the robot. The Tracked MERLIN (cf. Figure B.1(b)) is equipped with continuous tracks that surround its body. Hence it is able to navigate even in very cluttered environments. The third type of platform, which was used for the experiments carried out in the context of this work, is the Indoor MERLIN (cf. Figure B.1(c)). It is a small-sized car-like robot whose odometric sensors combined with the gyroscope provide adequate measurements of motion in the plane for short-term laboratory hardware experiments.

The MERLINs are equipped with a Siemens C167 microcontroller that processes incoming data from attached sensors and further is responsible for time-critical, low-level control algorithms. Moreover the vehicles are equipped with a stack of PC/104-modules that provide WLAN communication and, compared to the microcontroller, enough computational power for the implementation of high-level algorithms like, e.g., image processing. The MERLIN Control System, a modular structured software framework that is continuously enhanced, provides a powerful basis for implementation and testing of novel control algorithms. Among other things, it contains modules for the basic functionality like communication, access to sensor data, and control of the actuators.



(a) Outdoor MERLIN

(b) Tracked MERLIN

(c) Indoor MERLIN

Figure B.1: Three vehicles from the MERLIN family.

During the last years, MERLIN vehicles have been applied in various projects that, among other things, deal with driver assistant systems [ESS08], cooperative search and rescue missions [KKP+04], as well as cooperative box pushing tasks [Sch04].

C Towards the convergence of a recently proposed rendezvous control law for groups of unicycles

A discontinuous rendezvous control law for unicycle robots has been recently proposed in [DK07]. After a short introduction of the controller, we describe a system configuration with two vehicles for which the proposed control law does not necessarily achieves the desired rendezvous.

C.1 The control law

Theorem 4 in [DK07] states that, assuming the system's static communication graph is connected, N unicycles with kinematics

$$\begin{aligned} \dot{x}_i &= u_i \cos\theta_i, \\ \dot{y}_i &= u_i \sin\theta_i \qquad i \in \{1, \ldots, N\}, \\ \dot{\theta}_i &= \omega_i, \end{aligned}$$

converge to a common configuration in the state space by applying the following control law:

$$u_i = -\operatorname{sgn}(\delta_{xi}\cos\theta_i + \delta_{yi}\sin\theta_i) \cdot \sqrt{\delta_{xi}^2 + \delta_{yi}^2}, \tag{C.1}$$

$$\omega_i = -(\theta_i - \operatorname{atan2}(\delta_{yi}, \delta_{xi})), \tag{C.2}$$

where

$$\begin{aligned} \delta_{xi} &= 2 \sum_{j \in \mathcal{N}_i} \xi'\big((x_i - x_j)^2 + (y_i - y_j)^2\big)(x_i - x_j), \\ \delta_{yi} &= 2 \sum_{j \in \mathcal{N}_i} \xi'\big((x_i - x_j)^2 + (y_i - y_j)^2\big)(y_i - y_j), \end{aligned}$$

and

$$\operatorname{sgn}(x) = \left\{ \begin{array}{cl} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{array} \right. .$$

Here, $\mathcal{N}_i$ denotes the set of neighbors of vehicle i in the underlying interaction graph (cf. Section 6.1). The function $\xi : [0, d^2) \to [0, \infty)$ is defined as

$$\xi(x) = \begin{cases} a_1/(d^2 - x) & \text{if } \mu < x < d^2 \\ a_2 x & \text{if } 0 \le x \le \mu \end{cases}, \tag{C.3}$$

where the parameters are chosen as $d^2 = 2\mu, a_1 = a_2\mu^2$ such that the function is continuously differentiable for all x. Thus, its first derivative can be computed as

$$\xi'(x) = \begin{cases} a_1/(d^2 - x)^2 & \text{if } \mu < x < d^2 \\ a_2 & \text{if } 0 \le x \le \mu \end{cases}.$$

For the proof the authors apply Lyapunov's direct method for nonsmooth systems [SP94] with the Lyapunov function candidate

$$W = \sum_i^N \sum_{j \in \mathcal{N}_i} \xi(||q_i - q_j||^2).$$

They show that $\dot{\widetilde{W}} < 0$ for all configurations that are not an element of the set

$$S = \{(\delta_{xi} = \delta_{yi} = 0) \vee (\delta_{xi} \cos\theta_i + \delta_{yi} \sin\theta_i = 0), \forall i \in \{1, \ldots, N\}\},$$

for which $\dot{\widetilde{W}} = 0$. Further it is stated that because of the definition of control law (C.2), the set S is invariant except for the desired rendezvous configurations for which $\delta_{xi} = \delta_{yi} = 0$ holds for all i.

C.2 A counterexample

In this section we present a simple example in which the agents do not converge to a common configuration under the proposed control law (C.1)-(C.2).

Given two mutually connected agents with the initial configurations $(x_{1,0}, y_{1,0}, \theta_{1,0}) = (-\hat{r}, 0, \pi/2)$ and $(x_{2,0}, y_{2,0}, \theta_{2,0}) = (\hat{r}, 0, -\pi/2)$ with $\hat{r} > 0$ as depicted in Figure C.1 it is easy to see that in this case we have $\delta_{yi} = 0$ and $\cos\theta_i = 0$ for $i = 1, 2$. As a result, $\delta_{xi} \cos\theta_i + \delta_{yi} \sin\theta_i = 0$ and therefore the configuration is an element of the set S. From equations (C.1) and (C.2) we can compute

$$\omega_1 = \omega_2 = \pi/2,$$

as well as

$$u_i = -\sqrt{\delta_{xi}^2 + \delta_{yi}^2},$$

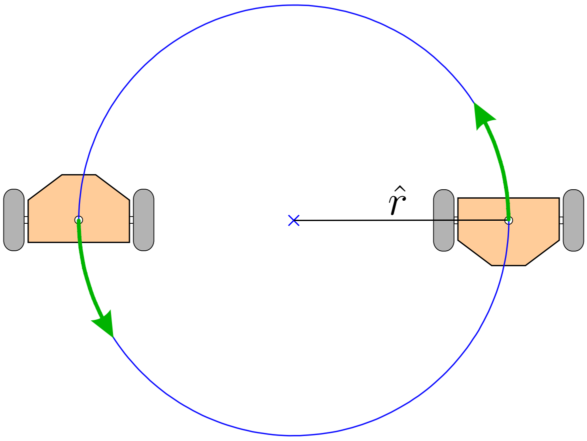


Figure C.1: Initial configuration for which the proposed rendezvous controller fails.

which correspond to the inputs of one possible Filippov solution starting from the specified configuration. By choosing the constants of equation (C.3) as $\mu \geq 4\hat{r}^2$ and $a_2 = \pi/8$ and considering the fact that $\delta_{y1} = \delta_{y2} = 0$ we obtain

$$\begin{aligned} u_i = -|\delta_{xi}| &= -\left| 2 \sum_{j \in \mathcal{N}_i} \xi'\big((x_i - x_j)^2 + (y_i - y_j)^2\big)(x_i - x_j) \right| \\ &= -4\hat{r}\xi'(4\hat{r}^2) = -\frac{\pi}{2}\hat{r}. \end{aligned} \tag{C.4}$$

Now, the signed curve radius of the motion trajectory of vehicle i can be computed as

$$r_i = \frac{u_i}{\omega_i} = -\hat{r}. \tag{C.5}$$

From equations (C.4) and (C.5) we can conclude that the vehicles move backwards on the smallest circle going through both vehicles without changing distance to each other. Therefore $\delta_{xi} \cos\theta_i + \delta_{yi} \sin\theta_i = 0$ and the trajectory is not leaving the set S for all time, which means that no decrease in the function value of W will be observed. Since $\hat{r}$ was chosen to be greater than zero, rendezvous will never be reached. Note that this special case is just one of the possible Filippov solutions, which makes the configuration a weak equilibrium of the closed-loop system. Like for the unstable weak equilibria in sections 6.4 and 6.5, any small perturbation would drive the system out.

In the same way counterexamples can be found for similar theorems by the same authors in [DK08a, Theorem 6] and [DK08b, Theorem 2]. The problem of the proposed control law is that depending on a vehicle's relative position with respect to its neighbors, we can achieve arbitrary speeds u_i, while the rate of turning $\dot{\theta}_i$ is not affected. A solution to this would be to include the movement speed u_i into equation (C.2) in such a way that the rate of turning $\dot{\theta}_i$ increases with an increasing movement speed u_i. For the control laws introduced in Section 6.4 and 6.5 this is automatically the case due to the characteristic of the kinematic model of car-like vehicles.

List of Figures

References

[ABLR$^+$06] Vincent Arguenon, Andre Bergues-Lagarde, Christophe Rosenberger, Per Bro, and Waleed Smari. Multi-agent based prototyping of agriculture robots. In *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS06)*, pages 282–288, May 2006.

[ABW05] Yaniv Altshuler, Alfred M. Bruckstein, and Israel A. Wagner. Swarm robotics for a dynamic cleaning problem. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 209–216, June 2005.

[ACBB95] Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robotics & Automation Magazine*, 2(1):27–35, March 1995.

[AKOI96] Tamio Arai, Daisuke Kurabayashi, Jun Ota, and Shinpei Ichikawas. Motion planning for cooperative sweeping with relocating obstacles. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 1513–1518, October 1996.

[AMI89] Hajime Asama, Akihiro Matsumo, and Yoshiki Ishida. Design of an autonomous and distributed robot system: Actress. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS89)*, pages 283–290, Tsukuba, Japan, September 1989.

[AOI$^+$91] Hajime Asama, Koichi Ozaki, Hiroaki Itakura, Akihiro Matsumoto, Yoshiki Ishida, and Isao Endo. Collision avoidance among multiple mobile robots based on rules and communication. In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS91)*, volume 3, pages 1215–1220, Osaka, Japan, November 1991.

[AOSY99] Hideki Ando, Yoshinobu Oasa, Ichiro Suzuki, and Masafumi Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, October 1999.

[APP02] Tamio Arai, Enrico Pagello, and Lynne E. Parker. Editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, October 2002.

[AS06] Mazda Ahmadi and Peter Stone. A multi-robot system for continuous area sweeping tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA06)*, pages 1724–1729, May 2006.

[ASTBK06] Michel Abou-Samah, Chin Pei Tang, Rajankumar M. Bhatt, and Venkat Krovi. A kinematically compatible framework for cooperative payload transport by nonholonomic mobile manipulators. *Autonomous Robots*, 21:227–242, 2006.

[BA98] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, December 1998.

[BC99] Andrea Bacciotti and Francesca Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 4:361–376, June 1999.

[BC04] Timothy D. Barfoot and Christopher M. Clark. Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46:65–78, February 2004.

[BCM08] Francesco Bullo, Jorge Cortés, and Sonia Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, September 2008.

[BCRD02] Timothy D. Barfoot, Christopher M. Clark, Stephen M. Rock, and Gabriele M. T. D'Eleuterio. Kinematic path-planning for formations of mobile robots with a nonholonomic constraint. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS02)*, pages 2819–2824, October 2002.

[BF08] Amit Bhatia and Emilio Frazzoli. Decentralized algorithm for minimum-time rendezvous of dubins vehicles. In *Proceedings of the American Control Conference (ACC08)*, pages 1343–1349, June 2008.

[BGP06] Dario Bauso, Laura Giarré, and Raffaele Pesenti. Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Systems & Control Letters*, 55(11):918–928, November 2006.

[BK04] Calin Belta and Vijay Kumar. Optimal motion generation for groups of robots: A geometric approach. *ASME Journal of Mechanical Design*, 126(1):63–70, January 2004.

[BLH01] Randal W. Beard, Jonathan Lawton, and Fred Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, November 2001.

[BMM08] Paolo Braca, Stefano Marano, and Vincenzo Matta. Enforcing consensus while monitoring the environment in wireless sensor networks. *IEEE Transactions on Signal Processing*, 56(7):3375–3380, July 2008.

[BMSS05] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):375–386, June 2005.

[BTK04] Rajankumar M. Bhatt, Chin Pei Tang, and Venkat Krovi. Geometric motion planning and formation optimization for a fleet of nonholonomic wheeled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA04)*, pages 3276–3281, April 2004.

[BV82] Vivek Borkar and Pravin P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, 27(3):650–655, June 1982.

[CFSZ08] Ruggero Carli, Fabio Fagnani, Alberto Speranzon, and Sandro Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, March 2008.

[CH02] VijaySekhar Chellaboina and Wassim M. Haddad. A unification between partial stability and stability theory for time-varying systems. *IEEE Control Systems Magazine*, 22(6):66–75, December 2002.

[CH03] VijaySekhar Chellaboina and Wassim M. Haddad. Erratum to "A unification between partial stability and stability theory for time-varying systems". *IEEE Control Systems Magazine*, 23(3):101–101, June 2003.

[Cla83] Frank H. Clarke. *Optimization and Nonsmooth Analysis.* John Wiley and Sons, 1983.

[CMB06] Jorge Cortés, Sonia Martínez, and Francesco Bullo. Robust rendezvous for mobile autonomous agents. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, August 2006.

[Cor08] Jorge Cortés. Discontinuous dynamical systems: A tutorial on solutions, nonsmooth analyses, and stability. *IEEE Control Systems Magazine*, 28(3):36–73, June 2008.

[CP07] Giuseppe Conte and Paris Pennesi. On convergence conditions for rendezvous. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC07)*, pages 2375–2378, December 2007.

[DB08] Borys Dabrowsky and Marek Banaszkiewicz. Multi-rover navigation on the lunar surface. *Advances in Space Research*, 42(2):369–378, July 2008.

[DBS05] Frauke Driewer, Herbert Baier, and Klaus Schilling. Robot-human rescue teams: a user requirements analysis. *Advanced Robotics*, 19(8):819–838, 2005.

[DeG74] Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, March 1974.

[DFK$^+$02] Aveek K. Das, Rafael Fierro, Vijay Kumar, James P. Ostrowski, John Spletzer, and Camillo J. Taylor. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825, October 2002.

[DK07] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52(5):916–922, May 2007.

[DK08a] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica*, 44(10):2648–2654, October 2008.

[DK08b] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. Inverse agreement algorithms with application to swarm dispersion for multiple nonholonomic agents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA08)*, pages 1973–1978, May 2008.

[DOK01] Jaydev P. Desai, James P. Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, December 2001.

[DOS98] Alessandro De Luca, Giuseppe Oriolo, and Claude Samson. Feedback control of a nonholonomic car-like robot. In Jean-Paul Laumond, editor, *Planning robot motion*. Springer-Verlag, 1998.

[DS07] Jason C. Derenick and John R. Spletzer. Convex optimization strategies for coordinating large-scale robot formations. *IEEE Transactions on Robotics*, 23(6):1252–1259, December 2007.

[Dub57] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, July 1957.

[dWS92] Carlos Canudas de Wit and Ole J. Sørdalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 37(11):1791–1797, November 1992.

[EH01] Magnus Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, December 2001.

[ESS08] Daniel Eck, Manuel Stahl, and Klaus Schilling. *The Small Outdoor Rover MERLIN and Its Assistance System for Tele-operations*, pages 277–286. Springer-Verlag, 2008.

[Fil88] Aleksei Fedorovich Filippov. *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.

[FM02] Jakob Fredslund and Maja J. Mataric. A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, 18(5):837–846, October 2002.

[FMC08] Jia Fang, A. Stephen Morse, and Ming Cao. Multi-agent rendezvousing with a finite set of candidate rendezvous points. In *Proceedings of the American Control Conference (ACC08)*, pages 765–770, June 2008.

[FS04] Thierry Fraichard and Alexis Scheuer. From reeds and shepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, 20(6):1025–1035, December 2004.

[FZ08] Fabio Fagnani and Sandro Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, May 2008.

[GPFZ07] Arturo Gil-Pinto, Philippe Fraisse, and René Zapata. Decentralized strategy for car-like robot formations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS08)*, pages 4176–4181, November 2007.

[GS03] Marco Gilioli and Klaus Schilling. Autonomous cooperative localization of mobile robots based on ranging systems. In *Proceedings of SPIE Unmanned Ground Vehicle Technology V*, volume 5083, pages 146–154, September 2003.

[GST08] Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. Invariance principles for switching systems via hybrid systems techniques. *Systems & Control Letters*, 57(12):980–986, December 2008.

[HLAS05] João P. Hespanha, Daniel Liberzon, David Angeli, and Eduardo D. Sontag. Nonlinear norm-observability notions and stability of switched systems. *IEEE Transactions on Automatic Control*, 50(2):154–168, February 2005.

[HSB$^+$07] Islam I. Hussein, Daniel J. Scheeres, Anthony M. Bloch, David C. Hyland, and N. Harris McClamroch. Optimal motion planning for dual-spacecraft interferometry. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2):723–737, April 2007.

[HSS06] Martin Hess, Martin Saska, and Klaus Schilling. Formation driving using partical swarm optimization and reactive obstacle avoidance. In *Proceedings of the 1st IFAC Workshop on Multivehicle Systems (MVS'06)*, October 2006.

[HSS07a] Martin Hess, Martin Saska, and Klaus Schilling. Autonomous multi-vehicle formations for cooperative airfield snow shoveling. In *Proceedings of the 3rd European Conference on Mobile Robots (ECMR07)*, Freiburg, Germany, September 2007.

[HSS07b] Martin Hess, Martin Saska, and Klaus Schilling. Enhanced motion planning for dynamic formations of nonholonomic mobile robots. In *Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV07)*, September 2007.

[HV94] Michiel Hazewinkel and Ivan M. Vinogradov. *Encyclopaedia of mathematics*. Springer-Verlag, 1994.

[HV00] Roberto Horowitz and Pravin Varaiya. Control design of an automated highway system. *IEEE Proceedings of the IEEE*, 88(7):913–925, July 2000.

[Ind99] Giovanni Indiveri. Kinematic time-invariant control of a 2D nonholonomic vehicle. In *Proceedings of the 38th IEEE Conference on Decision and Control (CDC99)*, volume 3, pages 2112–2117, 1999.

[JE07] Meng Ji and Magnus Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, August 2007.

[JLM03] Ali Jadbabaie, Jie Lin, and A. Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[JN02] Markus Jäger and Bernhard Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA02)*, volume 4, pages 3577–3582, 2002.

[JSJJ08] Björn Johansson, Alberto Speranzon, Mikael Johansson, and Karl Henrik Johansson. On decentralized negotiation of optimal consensus. *Automatica*, 44(4):1175–1179, April 2008.

[KBS07] Akshay Kashyap, Tamer Basar, and Rayadurgam Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, July 2007.

[KGCC$^+$08] Hadas Kress-Gazit, David C. Conner, Howie Choset, Alfred A. Rizzi, and George J. Pappas. Courteous cars: Decentralized multiagent traffic coordination. *IEEE Robotics & Automation Magazine*, 15(1):30–38, March 2008.

[Kha02] Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall, third edition, 2002.

[KJ08] Tamás Keviczky and Karl Henrik Johansson. A study on distributed model predictive consensus. In *Proceedings of the 17th IFAC World Congress*, Juli 2008.

[KKP$^+$04] Miroslav Kulich, Jan Kout, Libor Preucil, Roman Mazl, Jan Chudoba, Jari Saarinen, Jussi Suomela, Aarne Halme, Frauke Driewer, Herbert Baier, Klaus Schilling, Niramon Ruangpayoongsak, and Hubert Roth. PeLoTe - a heterogenous telematic system for cooperative search and rescue missions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS04)*, October 2004.

[KM08] Soummya Kar and José M. F. Moura. Sensor networks with random links: Topology design for distributed consensus. *IEEE Transactions on Signal Processing*, 56(7):3315–3326, July 2008.

[KOA$^+$96] Daisuke Kurabayashi, Jun Ota, Tamio Arai, Shinpei Ichikawa, Shingo Koga, Hajime Asama, and Isao Endo. Cooperative sweeping by multiple mobile robots with relocating portable obstacles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS96)*, pages 1472–1477, November 1996.

[KOAY96] Daisuke Kurabayashi, Jun Ota, Tamio Arai, and Eiichi Yoshida. Cooperative sweeping by multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA96)*, pages 1744–1749, April 1996.

[Lat91] Jean-Claude Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.

[Lau98] Jean-Paul Laumond, editor. *Robot Motion Planning and Control.* Springer-Verlag, 1998.

[LaV06] Steven M. LaValle. *Planning Algorithms.* Cambridge University Press, 2006.

[LBF04] Zhiyun Lin, Mireille E. Broucke, and Bruce A. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 49(4):622–629, April 2004.

[LG08] Yi Li and Kamal Gupta. Real-time motion planning of multiple formations in virtual environments: Flexible virtual structures and continuum model. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS08)*, pages 1902–1907, September 2008.

[Lib03] Daniel Liberzon. *Switching in Systems and Control.* Birkhäuser, 2003.

[LKYC99] Sungon Lee, Minchul Kim, Youngil Youm, and Wankyun Chung. Control of a car-like mobile robot for parking problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA99)*, pages 1–6, May 1999.

[LMA03] Jie Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem. In *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC03)*, pages 1508–1513, December 2003.

[LMA04] Jie Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem - the asynchronous case. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC04)*, pages 1926–1931, December 2004.

[LMA07a] Jie Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem. part 1: The synchronous case. *SIAM Journal on Control and Optimization (SICON)*, 46(6):2096–2119, 2007.

[LMA07b] Jie Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem. part 2: The asynchronous case. *SIAM Journal on Control and Optimization (SICON)*, 46(6):2120–2147, 2007.

[LMPS05] Dongjun Lee, Oscar Martinez-Palafox, and Mark W. Spong. Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Application. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA05)*, pages 366–371, April 2005.

[LS05] Dongjun Lee and Mark W. Spong. Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA05)*, pages 360–365, April 2005.

[LT97] M. Anthony Lewis and Kar-Han Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4:387–403, 1997.

[LY02] Chaomin Luo and Simon X. Yang. A real-time cooperative sweeping strategy for multiple cleaning robots. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 660–665, October 2002.

[LYC99] Sungon Lee, Youngil Youm, and Wankyun Chung. Control of car-like mobile robots for posture stabilization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS99)*, pages 1745–1750, October 1999.

[Lyn96] Nancy Ann Lynch. *Distributed algorithms.* Morgan Kaufmann, 1996.

[LYS03] Chaomin Luo, Simon X. Yang, and Deborah A. Stacey. Real-time path planning with deadlock avoidance of multiple cleaning robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA03)*, pages 4080–4085, September 2003.

[MAG06] Jose Luis Mancilla-Aguilar and Rafael A. García. An extension of Lasalle's invariance principle for switched systems. *Systems & Control Letters*, 55(5):376–384, May 2006.

[MBCF07] Sonia Martínez, Francesco Bullo, Jorge Cortés, and Emilio Frazzoli. On synchronous robotic networks – part ii: Time complexity of rendezvous and deployment algorithms. *IEEE Transactions on Automatic Control*, 52(12):2214–2226, December 2007.

[MBF04] Joshua A. Marshall, Mireille E. Broucke, and Bruce A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963–1974, November 2004.

[MFP04] Raj Madhavan, Kingsley Fregene, and Lynne E. Parker. Distributed cooperative outdoor multirobot localization and mapping. *Autonomous Robots*, 17:23–39, July 2004.

[MM97] Robert T. M'Closkey and Richard M. Murray. Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Transactions on Automatic Control*, 42(5):614–628, May 1997.

[Mor05] Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, February 2005.

[MSG$^+$08] Silvia Mastellone, Dušan M. Stipanović, Christopher R. Graunke, Koji A. Intlekofer, and Mark W. Spong. Formation control and collision avoidance for multi-agent nonholonomic systems: Theory and experiments. *The International Journal of Robotics Research*, 27(1):107–126, January 2008.

[Mur07] Richard M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, September 2007.

[MY98] Tao Wei Min and How Khee Yin. A decentralized approach for cooperative sweeping by multiple mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS98)*, pages 380–385, October 1998.

[NB06] Giuseppe Notarstefano and Francesco Bullo. Distributed consensus on enclosing shapes and minimum time rendezvous. In *Proceedings of the 45th IEEE Conference on Decision and Control (CDC06)*, pages 4295–4300, December 2006.

[ODV02] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, November 2002.

[ÖEH02] Petter Ögren, Magnus Egerstedt, and Xiaoming Hu. A control Lyapunov function approach to multiagent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, October 2002.

[OS05] Reza Olfati-Saber. Ultrafast consensus in small-world networks. In *Proceedings of the American Control Conference*, pages 2371–2378, June 2005.

[OSFM07] Reza Olfati-Saber, J. Alexander Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[OSM04] Reza Olfati-Saber and Richard M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[PCM08] Mike Peasgood, Christopher M. Clark, and John McPhee. A complete and scalable strategy for coordinating multiple robots within roadmaps. *IEEE Transactions on Robotics*, 24(2):283–292, April 2008.

[PS87] Brad E. Paden and Shankar S. Sastry. A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators. *IEEE Transactions on Circuits and Systems*, 34(1):73–82, January 1987.

[PS07] Maurizio Porfiri and Daniel J. Stilwell. Consensus seeking over random weighted directed graphs. *IEEE Transactions on Automatic Control*, 52(9):1767–1773, September 2007.

[RB02] Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, October 2002.

[RB05] Wei Ren and Randal W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, May 2005.

[RBA05] Wei Ren, Randal W. Beard, and Ella M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference*, pages 1859–1864, June 2005.

[RCB+08] Wei Ren, Haiyang Chao, William Bourgeous, Nathan Sorensen, and Yangquan Chen. Experimental validation of consensus algorithms for multi-vehicle cooperative control. *IEEE Transactions on Control Systems Technology*, 16(4):745–752, July 2008.

[Ren07] Wei Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory & Applications*, 1(2):505–512, March 2007.

[Ren08] Wei Ren. On consensus algorithms for double-integrator dynamics. *IEEE Transactions on Automatic Control*, 53(6):1503–1509, July 2008.

[RMC07] Wei Ren, Kevin L. Moore, and Yangquan Chen. High-order and model reference consensus algorithms in cooperative control of multi-vehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 129(5):678–688, September 2007.

[RW99] John H. Reif and Hongyan Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27:171–194, 1999.

[Sam93] Claude Samson. Time-varying feedback stabilization of car-like wheeled mobile robots. *The International Journal of Robotics Research*, 12(1):55–64, 1993.

[SBF07] Stephen L. Smith, Mireille E. Broucke, and Bruce A. Francis. Curve shortening and the rendezvous problem for mobile autonomous robots. *IEEE Transactions on Automatic Control*, 52(6):1154–1159, June 2007.

[Sch04] Klaus Schilling. Navigation by cooperating mobile robots. In *Proceedings of SPIE Intelligent Robots and Computer Vision XXII: Algorithms, Techniques, and Active Vision*, volume 5608, pages 354–359, October 2004.

[SG06] Arpita Sinha and Debasish Ghose. Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents. *IEEE Transactions on Automatic Control*, 51(11):1819–1824, November 2006.

[SG07] Arpita Sinha and Debasish Ghose. Generalization of nonlinear cyclic pursuit. *Automatica*, 43(11):1954–1960, November 2007.

[SHS07] Martin Saska, Martin Hess, and Klaus Schilling. Path planning and motion coordination for compact vehicle-formations. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA'07)*, December 2007.

[SHS08a] Martin Saska, Martin Hess, and Klaus Schilling. Efficient airport snow shoveling by applying autonomous multi-vehicle formations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA08)*, pages 1684–1690, May 2008.

[SHS08b] Martin Saska, Martin Hess, and Klaus Schilling. Route scheduling approach for airport snow shoveling using formations of autonomous ploughs. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV08)*, December 2008.

[SHSŠ07] Dušan M. Stipanović, Peter F. Hokayem, Mark W. Spong, and Dragoslav D. Šiljak. Cooperative avoidance control for multiagent systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):699–707, September 2007.

[Šil78] Dragoslav D. Šiljak. *Large-scale dynamic systems.* North-Holland, 1978.

[SİTT04] Dušan M. Stipanović, Gökhan İnalhan, Rodney Teo, and Claire J. Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296, August 2004.

[SM02] Klaus Schilling and Qinghao Meng. The MERLIN vehicles for outdoor applications. In *Proceedings of SPIE Unmanned Ground Vehicle Technology IV*, volume 4715, pages 43–49, 2002.

[SN04] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots.* MIT Press, 2004.

[SP94] Daniel Shevitz and Brad E. Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on Automatic Control*, 39(9):1910–1914, September 1994.

[Spe06] Alberto Speranzon. *Coordination, Consensus and Communication in Multi-Robot Control Systems.* PhD thesis, KTH School of Electrical Engineering, 2006.

[SRB$^+$97] Klaus Schilling, Lutz Richter, Marco C. Bernasconi, Christoph Jungius, and César Garcia-Marirrodriga. Operations and control of the mobile instrument deployment device on the surface of mars. *Control Engineering Practice*, 5(6):837–844, June 1997.

[SSHH04] William M. Spears, Diana F. Spears, Jerry C. Hamann, and Rodney Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17:137–162, 2004.

[SZ06a] Luca Schenato and Sandro Zampieri. On the performance of randomized communication topologies for rendezvous control of multiple vehicles. In *Proceedings of the Conference on Mathematical Theory of Networks and Systems*, July 2006.

[SZ06b] Luca Schenato and Sandro Zampieri. Optimal rendezvous control for randomized communication topologies. In *Proceedings of the 45th IEEE Conference on Decision and Control (CDC06)*, pages 4339–4344, December 2006.

[TBA86] John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[TPK04] Herbert G. Tanner, George J. Pappas, and Vijay Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, June 2004.

[Tsi84] John N. Tsitsiklis. *Problems in decentralized decision making and computation.* PhD thesis, Massachusetts Institute of Technology, 1984.

[Vor98] Vladimir I. Vorotnikov. *Partial Stability and Control*. Birkhäuser, 1998.

[Vor05] Vladimir I. Vorotnikov. Partial stability and control: The state-of-the-art and development prospects. *Automation and Remote Control*, 66(4):511–561, April 2005.

[WB97] Israel A. Wagner and Alfred M. Bruckstein. Cooperative cleaners: A case of distributed ant-robotics. In Arogyaswami Paulraj, Vwani Roychowdhury, and Charles D. Schaper, editors, *Communications, Computation, Control, and Signal Processing: A Tribute to Thomas Kailath*, pages 289–308. Kluwer Academic Publishers, 1997.

[WH07] Paul K. C. Wang and Fred Y. Hadaegh. Formation flying of multiple spacecraft with autonomous rendezvous and docking capability. *IET Control Theory & Applications*, 1(2):494–504, March 2007.

[WX07] Long Wang and Feng Xiao. Finite-time consensus problems for networks of dynamic agents. *ArXiv Mathematics e-prints*, January 2007.

[XW07] Feng Xiao and Long Wang. Consensus problems for high-dimensional multi-agent systems. *IET Control Theory & Applications*, 1(3):830–837, May 2007.

[XW08] Feng Xiao and Long Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8):1804–1816, September 2008.

[YLL08] Jingjin Yu, Steven M. LaValle, and Daniel Liberzon. Rendezvous without coordinates. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 1803–1808, December 2008.

[YTN05] Yuh Yamashita, Takayuki Tsuzaki, and Hisakazu Nakamura. On a topological condition for strongly asymptotically stable differential inclusions. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC05) and European Control Conference (ECC05)*, pages 5444–5449, 2005.

[ZP07] Michael M. Zavlanos and George J. Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics*, 23(4):812–816, August 2007.

[ZP08] Michael M. Zavlanos and George J. Pappas. Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, 23(1):232–243, February 2008.

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Zitation dieser Publikation

HESS, M. (2010). Motion coordination and control in systems of nonholonomic autonomous vehicles. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 2. Würzburg: Universität Würzburg.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85917
Fax: +49 (0) 931 - 31 - 85970

opus@bibliothek.uni-wuerzburg.de
http://www.opus-bayern.de/uni-wuerzburg/

ISSN (Internet): 1868-7474
ISSN (Print): 1868-7466
ISBN-10: 3-923959-55-9
ISBN-13: 978-3-923959-55-6