Abstract

In this thesis, we consider the analysis and design of first-order optimization algorithms employing systems and control theory. We recast algorithm design problems as controller synthesis problems; techniques from control theory then enable us to systematically construct tailored optimization algorithms adapted to various situations. In particular, we consider two specific classes of optimization algorithms: (i) continuous-time distributed optimization algorithms for constrained convex optimization, and (ii) robust discrete-time optimization algorithms for unconstrained convex optimization.

Concerning (i), we consider a group of agents sharing information over a communication network described by a directed time-invariant graph aiming to cooperatively solve a convex optimization problem with shared equality and inequality constraints. Utilizing geometric control theory in a novel and innovative fashion, in particular Lie bracket averaging techniques, we directly address the core challenge of distributed problems, namely limited local information. Employing saddle-point dynamics, we derive a novel methodology that enables the design of distributed continuous-time optimization algorithms solving a class of optimization problems under minimal assumptions on the graph topology as well as on the structure of the optimization problem. Generalizing this approach, we further establish a systematic way of deriving continuous-time distributed algorithms from non-distributed ones.

Concerning (ii), we consider the problem of analyzing and designing gradient-based discrete-time optimization algorithms for a class of unconstrained optimization problems having strongly convex objective function with Lipschitz continuous gradient. By formulating the problem as a robustness analysis problem and employing a suitable adaptation of the theory of integral quadratic constraints (IQCs), we establish a framework that allows analyzing convergence rates and robustness properties of existing algorithms and enables the design of novel robust optimization algorithms with specified guarantees. Taking advantage of the embedding into integral quadratic constraint theory, we further extend the framework to design algorithms that are capable of exploiting additional structure in the objective function.

Deutsche Kurzfassung

In der vorliegenden Arbeit werden Analyse- und Entwurfsmethoden für gradientenbasierte Optimierungsalgorithmen entwickelt, indem mit Hilfe von systemtheoretischen Ansätzen das Entwurfsproblem in geeigneter Weise als ein Reglerentwurfsproblem umformuliert wird. Dieser Ansatz ermöglicht es, auf systematische Art und Weise maßgeschneiderte Optimierungsalgorithmen zu entwerfen. Dabei werden in dieser Arbeit insbesondere zwei spezielle Klassen von Optimierungsalgorithmen betrachtet: (i) zeitkontinuierliche verteilte Optimierungsalgorithmen zur Lösung konvexer Optimierungsprobleme mit Nebenbedingungen und (ii) robuste zeitdiskrete Optimierungsalgorithmen zur Lösung konvexer Optimierungsprobleme ohne Nebenbedingungen.

In Fall (i) verfolgt eine Gruppe von Recheneinheiten, auch Agenten genannt, das Ziel, kooperativ ein Optimierungsproblem mit gemeinsamen Gleichungs- und Ungleichungsnebenbedingungen zu lösen, wobei jeder Agent nur Zugriff auf eine begrenzte Menge lokal verfügbarer Information hat. Dazu tauschen die Agenten untereinander diese lokalen Informationen über ein Kommunikationsnetz aus, das abstrakt durch einen gerichteten, zeitinvarianten Graph repräsentiert werden kann. Der vorgestellte Ansatz basiert auf einer neuartigen und innovativen Verwendung von Methoden aus der geometrischen Regelung, im Speziellen Lie Klammer Approximationen. Dieser Ansatz erlaubt es, die Hauptschwierigkeit verteilter Probleme, nämlich die nur lokal verfügbare Information, direkt und systematisch anzugehen. Eine Kombination der Methodik mit Sattelpunktdynamiken ermöglicht es dann, verteilte zeitkontinuierliche Optimierungsalgorithmen unter geringen Voraussetzungen an die Struktur des Graphen und des Optimierungsproblems zu entwerfen. Durch weitere Verallgemeinerung dieser Methodik wird zudem ein systematischer Ansatz zum Entwurf verteilter aus nicht verteilten Algorithmen vorgestellt.

In (ii) wird die Analyse und der Entwurf gradientenbasierter zeitdiskreter Optimierungsalgorithmen für eine Klasse von unbeschränkten Optimierungsproblemen betrachtet, deren Kostenfunktion stark konvex ist und einen Lipschitz-stetigen Gradienten besitzt. Durch Umformulierung des Entwurfproblems als ein Problem der robusten Regelung und eine geeignete Modifikation der sogenannten IQC-Theorie (integral quadratic constraints) wird eine Methodik hergeleitet, die es erlaubt, sowohl die Konvergenzraten und Robustheitseigenschaften von bekannten Algorithmen zu analysieren als auch neue Optimierungsalgorithmen zu entwerfen, die vorgegebene Konvergenzraten- und Robustheitsgarantien erfüllen. Die Einbettung in die IQC-Theorie ermöglicht es dabei auch, Algorithmen zu entwerfen, die mögliche strukturelle Eigenschaften der Kostenfunktion explizit ausnutzen können.

1
Introduction

1.1 Motivation and Background

Optimization plays an important role in many fields of applications and is the backbone of a multitude of modern technologies such as real-time control, machine learning or data analytics. Having reliable, fast and flexible optimization algorithms available hence is of key importance. Over the last decades, a variety of optimization algorithms applicable in different situations have been developed and proven themselves in real-world applications. However, the technological progress often requires the adaptation to novel challenges that existing algorithms cannot handle appropriately, hence necessitating the development of tailored algorithms. Though, it is fair to say that the design of new algorithms, but also their analysis, still is more art than science based on experience, expert knowledge and good ideas. A systematic framework to analyze and modify existing or design novel algorithms adapted to different situations is not available yet. On the other hand, many optimization algorithms are, in essence, dynamical systems having an equilibrium characterized by the solutions of a class of optimization problems. Systems and control theory provides a quite mature set of tools for analyzing the convergence and stability properties of equilibria as well as for designing controllers that stabilize a given equilibrium. While the apparent relation between these two areas of research is not a new discovery, its full potential has not been exploited yet.

In the present thesis, we aim to contribute to linking the areas of optimization and systems and control theory and show that the latter provides novel ways to analyze and design optimization algorithms adapted to various problem setups. The main theme is to recast optimization algorithm design problems as particular controller synthesis problems and utilize methods from systems and control theory to enable a systematic design of optimization algorithms, see also Figure 1.1. In this thesis, we consider two particular classes of algorithm design problems, namely distributed optimization and robust optimization in the presence of noise; we are convinced that systems and control theory has the potential to provide a powerful approach to optimization algorithm design in general.

This view is supported by a growing amount of publications and applications following this systems theoretic approach to optimization. Recent advances in machine learning, data science or real-time decision-making as well as optimization-based control techniques such as model predictive control rely to a large extent on efficient optimization algorithms.

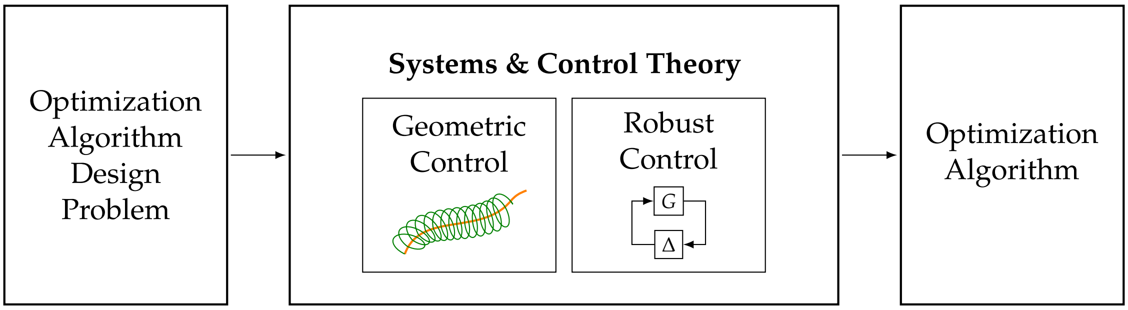

Figure 1.1. An illustration of the premise of the present thesis: We recast optimization algorithm design problems as controller synthesis problems and utilize and extend methods from systems and control theory, in particular geometric control and robust control, to facilitate a systematic design of tailored optimization algorithms.

This development has pushed the need for optimization algorithm design tools, which in turn motivated researchers to more thoroughly investigate and exploit the link between optimization algorithms and systems theory in the last years. Historically, this link can be traced back to the early days of Extremum Seeking Control (Draper & Li, 1951; Leblanc, 1922). While several other results from the last century also can be said to follow similar system theoretic ideas for optimization algorithm design, it was in Brockett (1988, 1991) where the author explicitly utilized geometric control theory to design dynamical systems that are able to solve tasks such as sorting lists or solving linear programs, providing an alternative to classical algorithms for such type of problems. This approach was further pursued by a group of researchers and other tasks, e.g., singular value decomposition, were addressed from a systems theoretic perspective as well (Helmke & Moore, 1994). Still, even more than ten years later, it is stated in Bhaya and Kaszkurewicz (2006) that although "some simple ideas from control theory can be used to systematize a class of approaches to algorithm analysis and design [...] control and system theory ideas have been underexploited in this context". In the last decade, several authors picked up on that (Dürr & Ebenbauer, 2012; Hauswirth, Bolognani, Hug, & Dörfler, 2019; Michalowsky & Ebenbauer, 2014; Wang & Elia, 2010, 2011, just to name a few), mainly from a continuous-time perspective that is probably more common in control theory, which also lead to a regained interest in continuous-time optimization algorithms (Gharesifard & Cortés, 2014; Niederländer & Cortés, 2015; Su, Boyd, & Candes, 2014). Recent advances (Wibisono, Wilson, & Jordan, 2016) further promote this continuous-time perspective enabling a deeper understanding also of discrete-time algorithms. Another approach that turned out to be a very fruitful example for utilizing control theoretic methods in optimization is the interpretation of gradient-based optimization algorithms in a robust control setting: with optimization algorithms typically having to be applicable to a class of objective functions, the idea of interpreting the gradient of the objective function as an uncertainty seems natural. This idea was first followed in Michalowsky and Ebenbauer (2014) in a continuous-time setting and in Lessard, Recht, and Packard (2016) for discrete-time optimization. In the latter work, the authors utilize integral quadratic constraint theory, which is well-established in robust control, to analyze a class of optimization algorithms, thereby unifying the analysis of several popular gradient-based algorithms. In the last two years, this approach has been followed in a number of publications (e.g., Aybat, Fallah, Gürbüzbalaban, and Ozdaglar (2019); Cyrus, Hu, Van Scoy, and Lessard (2018); Fazlyab, Ribeiro, Morari, and Preciado (2018); Michalowsky, Scherer, and Ebenbauer (2020); Safavi, Joshi, França, and Bento (2018); Van Scoy, Freeman, and Lynch (2018)).

1.2 Problem Formulation

In this thesis, we aim to provide a systems theoretic approach to two particular aspects of optimization: (i) distributed optimization over directed graphs and (ii) fast and robust optimization in the presence of noise. Following the premise illustrated in Figure 1.1, we address these two aspects by a proper reformulation of the algorithm design problem as a controller synthesis problem. We formalize the specific problems in the remainder and briefly sketch how we approach them.

Throughout the thesis, we consider (constrained) convex optimization problems

$$\begin{aligned} \underset{z\in\mathbb{R}^p}{\text{minimize}} \quad & H(z) \\ \text{s.t.} \quad & a(z) = 0 \\ & c(z) \leq 0, \end{aligned} \tag{1.1}$$

where $H : \mathbb{R}^p \to \mathbb{R}$, $a : \mathbb{R}^p \to \mathbb{R}^{n_{\text{eq}}}$, $c : \mathbb{R}^p \to \mathbb{R}^{n_{\text{ineq}}}$, $n_{\text{eq}}, n_{\text{ineq}} \in \mathbb{N}_{>0}$, $H, a, c \in \mathcal{C}^2$. Our goal is then to design optimization algorithms, i.e., dynamic systems, both in continuous- and discrete-time, that converge to a minimizer of (1.1), assuming that such a minimizer exists. More precisely, we consider deterministic optimization algorithms described by continuous- or discrete-time dynamic systems

$$x^+(t) = f\big(t, x(t)\big) \tag{1.2a}$$

$$z(t) = h\big(x(t)\big), \tag{1.2b}$$

where $x(t) \in \mathbb{R}^N$ for some $N \geq p$, $z(t) \in \mathbb{R}^p$, and f, h are functions to be designed. In a continuous-time setting, t is a non-negative real number representing the time and $x^+(t) = \dot{x}(t) = \frac{\mathrm{d}x}{\mathrm{d}t}(t)$ is the usual time derivative; in a discrete-time setting, t takes integer values representing time steps and $x^+(t) = x(t+1)$. We concentrate on first-order optimization algorithms, i.e., f may only depend on $H, a, c, \nabla H, \nabla a, \nabla c$ but no higher-order derivatives. In simple words, our overall goal is then formulated as follows:

General Optimization Algorithm Design Problem. "Given a class of optimization problems of the form (1.1) and some design specifications, design optimization algorithms (1.2) such that, as t tends to infinity, $z(t)$ converges to a minimizer of (1.1) for any instance of (1.1) and the specifications are met."

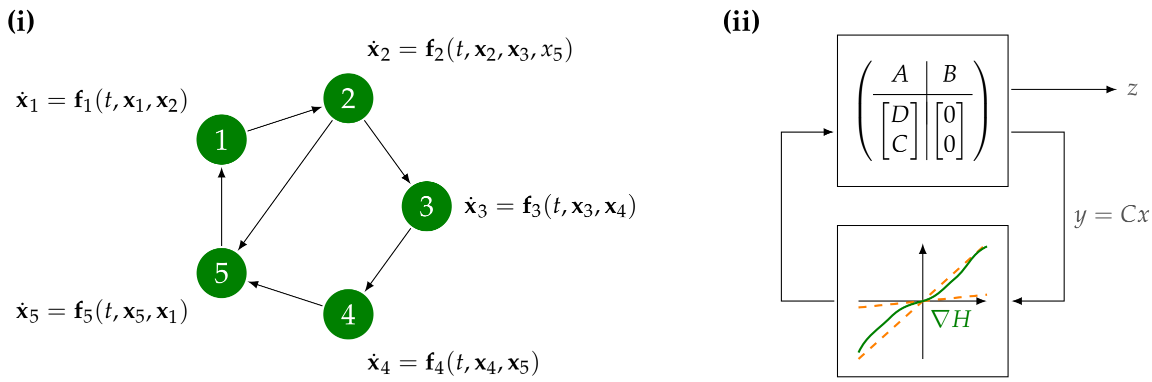

Figure 1.2. Two exemplary problems of type (i) and (ii). **Left (i):** A communication network of $n = 5$ agents with agent states $\mathbf{x}_i$, $i = 1, \dots, 5$. The arrows indicate the directions of information access; for example, in addition to its own state, agent 2 has access to the states $\mathbf{x}_3, \mathbf{x}_5$ of agent 3 and 5 but not to the states of the other agents in the network. This is reflected in the agent dynamics depicted next to the agents and illustrates the main challenge in designing distributed algorithms, namely limited locally available information. **Right (ii):** A block diagram representation of the class of algorithms (1.5) we consider in (ii). We reformulate the algorithm design problem as a robust controller synthesis problem interpreting the gradient of the objective function ∇H as an uncertainty. The challenge is then to design matrices A, B, C, D such that, as t tends to infinity, $z(t)$ converges to a minimizer of the optimization problem (1.1) for all H in some class.

We address the latter problem by designing the functions f, h in such a way that (1.2) has a (globally) asymptotically stable equilibrium at a point $x^\star$ that has the property that $h(x^\star) = z^\star$, where $z^\star$ is a minimizer of (1.1). In this manner, in its core the problem boils down to a stabilization problem with the important distinction that the point to be stabilized is not known a-priori but determined by (1.1). As to the additional design specifications, we are particularly interested in addressing two specific aspects of this general problem: (i) distributed optimization over directed graphs and (ii) fast and robust optimization in the presence of noise. Those two aspects are highly relevant in modern applications; we make this more precise in the following and specify the general problem formulation for (i) and (ii).

(i) Distributed Optimization Algorithms. In distributed optimization, a group of computation units, often called agents, cooperatively tries to solve an optimization problem. The idea of distributed algorithms is to have each agent solve a smaller subproblem using a limited amount of local information only and, by sharing information amongst the agents over a communication network, ensure that the original problem is solved. Distributed algorithms have the advantage that they are usually less error-prone, might require less communication and can respect possible privacy issues.

In this part, we are aiming for designing distributed continuous-time optimization algorithms (1.2). More precisely, we consider a group of n agents where each agent's state evolves according to its individual agent dynamics described by a differential equation and the complete distributed algorithm (1.2) is given by the collection of all agent dynamics. There is no common understanding of a distributed algorithm; in this thesis, simply put, we call an algorithm distributed if each agent only uses its own state as well as the states of the agents it has access to, where information access is encoded by a graph representation of the communication network. An exemplary illustration of the situation is depicted in Figure 1.2, (i). As visible from this example, in essence, this understanding of a distributed algorithm then amounts to design f in (1.2) in such a way that it respects certain information constraints induced by the communication network. More precisely, we consider a subclass of the general continuous-time algorithm dynamics (1.2) given by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} f_1\big(t, [x_k(t)]_{k \in \mathcal{N}_\mathcal{G}(1)}\big) \\ f_2\big(t, [x_k(t)]_{k \in \mathcal{N}_\mathcal{G}(2)}\big) \\ \vdots \\ f_n\big(t, [x_k(t)]_{k \in \mathcal{N}_\mathcal{G}(n)}\big) \end{bmatrix} \tag{1.3a}$$

$$z(t) = h\big(x(t)\big), \tag{1.3b}$$

where $x_i(t) \in \mathbb{R}^{N_i}$ is the state of the ith agent and $[x_k]_{k \in \mathcal{N}_\mathcal{G}(i)}$ is the collection of states of agents the ith agent has access to via the communication network, i.e., in a graph theoretic language, the out-neighboring states of agent i as well as its own state. Our goal is then to design the functions f_i in (1.3) such that $z(t)$ converges to a minimizer of (1.1). The main challenge in designing distributed algorithms (1.3) compared to general algorithms (1.2) is the limited information available to each agent rendering the stabilization problem a sparse stabilization problem. In other words, from a controller design perspective, there is a limited set of admissible control directions determined by the communication network. Such situations have been investigated a lot in nonlinear control theory, in particular in geometric control, e.g., in controllability analysis or the control of nonholonomic mechanical systems. Our approach relies on employing these ideas in a novel way to design distributed optimization algorithms.

(ii) Robust Optimization Algorithms. In many applications it is of key importance to have optimization algorithms that provide an accurate solution in guaranteed time. However, the convergence time of optimization algorithms is often hard to analyze and further, existing algorithms may not fulfill the imposed convergence rate requirements. Besides, while many algorithms perform well in an idealized setting, they are sensitive towards various disturbances, resulting in slow convergence. Such kind of disturbances arise, for example, in a data-based setting where the optimization problem specifiers H, a, c are generated from data.

In the second problem investigated in this thesis we are concerned with the design of discrete-time optimization algorithms applicable to a class of unconstrained optimization problems that (a) provide specified convergence rate guarantees, (b) are insensible towards noise in the optimization problem data (i.e., H in (1.1)) and (c) are capable of exploiting additional structural properties of the objective function. The class of algorithms we propose in this thesis is motivated by a generalization of existing algorithms. We explain the idea by means of the Heavy Ball Method (Polyak, 1987) that, for scalar optimization ($p = 1$), can be represented in the form (1.2) as

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} 1+\nu_2 & -\nu_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - \begin{bmatrix} \nu_1 \\ 0 \end{bmatrix} \nabla H\big(x_1(t)\big) \tag{1.4a}$$

$$z(t) = x_1(t), \tag{1.4b}$$

where ν_1, ν_2 are real parameters. A suitable choice of parameters for a class of objective functions H has been derived in (Polyak, 1987) where also the convergence of the algorithm (1.4) towards a minimizer $z^\star$ of H has been analyzed. The following question arises: How do these parameters need to be adapted – and how does this affect the convergence rate – if the class of objective functions changes or is refined or if the gradient of the objective function is affected by noise? Going further, how do we need to change (1.4) structurally in order to adapt to such novel situations? In this part of the thesis, we provide answers to such questions by generalizing (1.4) to a particular subclass of optimization algorithms (1.2) of the form

$$x(t+1) = Ax(t) + B\nabla H\big(Cx(t)\big) \tag{1.5a}$$

$$z(t) = Dx(t), \tag{1.5b}$$

where $x(t) = \begin{bmatrix} x_1(t)^\top & \dots & x_n(t)^\top \end{bmatrix}^\top \in \mathbb{R}^{np}$, $x_i(t) \in \mathbb{R}^p$, $i \in \{1, \dots, n\}$, $z(t) \in \mathbb{R}^p$. The design goal then amounts to determine matrices $A \in \mathbb{R}^{np \times np}$, $B \in \mathbb{R}^{np \times p}$, $C \in \mathbb{R}^{p \times np}$, $D \in \mathbb{R}^{p \times np}$, independent of H, such that, for any H in a given class, the dynamics (1.5a) have an asymptotically stable equilibrium at $x^\star$ with the property that $Dx^\star = z^\star$, where $z^\star$ is the minimizer of H. The core idea is to interpret the unknown objective function – or more precisely its gradient – as an uncertainty (see Figure 1.2, (ii)); we then need to render $x^\star$ asymptotically stable for (1.5) for any realization of the uncertainty. In other words, the problem is recast as a robust controller synthesis problem.

1.3 Contributions and Outline

In this thesis, we provide systematic procedures to analyze and design two classes of optimization algorithms. In particular, following the premise of a systems theoretic approach, we embed both problems (i) and (ii) described in Section 1.2 in a systems and control theoretic setup. For each problem (i), (ii), we then provide a framework that applies to a large class of optimization problems, can be extended systematically and allows for an automation of the algorithm design process. The two problems also build the two main chapters of this thesis that we outline in the following.

- In Chapter 2, we address problem (i) and provide a systematic framework to derive continuous-time distributed optimization algorithms from non-distributed ones. The approach is based on Lie bracket averaging techniques. We propose a two-step procedure

where the first step consists of finding certain Lie bracket representations of parts of the algorithm that cannot be implemented in a distributed fashion (we call them non-admissible vector fields) and the second step is to determine distributed approximations thereof. We discuss the procedure by means of saddle-point dynamics and show how it can be applied to other non-distributed optimization algorithms.
More specifically, concerning the first step, we provide Lie bracket representations of a large class of non-admissible vector fields (Proposition 1, Lemma 3). Applying the results to saddle-point dynamics, we show that such Lie bracket representations can be obtained under mild assumptions on the optimization problem (Lemma 4). Concerning the second step of determining distributed approximations, we modify the construction procedure from Liu (1997a) and derive a simplified version thereof allowing an explicit representation of the distributed approximation in certain cases (Proposition 2). We finally combine both steps and thereby provide a methodology to derive distributed optimization algorithms from non-distributed ones (Theorem 2). The results of this chapter are based on the papers Michalowsky, Gharesifard, and Ebenbauer (2017a, 2018, 2020) and some parts of the text, in particular in Section 2.4, are identical.

- In Chapter 3, we address problem (ii). By formulating the problem as a robustness analysis problem and making use of a suitable adaptation of the theory of integral quadratic constraints, we establish a framework that allows to analyze convergence rates and robustness properties of existing algorithms and design novel optimization algorithms that are robust towards noise, fulfill specified guarantees and are capable of exploiting additional structure in the objective function.
  Specifically, our main contributions are as follows: We propose a class of gradient-based algorithms that generalizes existing algorithms and derive necessary and sufficient conditions for these algorithms to be capable of solving a class of optimization problems (Theorem 3). Embedding the problem in the framework of robust control, we then derive convex analysis tools by means of linear matrix inequalities (LMIs), both in regard to convergence rates (Theorem 7) and robustness (Theorem 8). To this end, we provide a general procedure to obtain multipliers for exponential stability results from standard ones (Lemma 7) and utilize this to derive a class of multipliers generalizing those proposed in Boczar, Lessard, and Recht (2015); Freeman (2018); Lessard et al. (2016) (Theorem 6). We further provide convex synthesis conditions allowing the design of novel algorithms with specified robustness properties (Theorem 9) and show how to additionally exploit structural characteristics of the objective function (Lemma 10). The results presented in this chapter have been submitted to a large extent (Michalowsky, Scherer, & Ebenbauer, 2020) and are partly based on Michalowsky and Ebenbauer (2014, 2016).

- In Chapter 4, we give a summary of our results and discuss future research directions.

To streamline the presentation, we introduce the notation as well as the required technical background in a summarized form in Appendix A; an overview of the notation is also provided on page 147. All technical proofs of the mathematical statements are collected in Appendix B; Appendix C contains some additional material.

2 Design of Distributed Optimization Algorithms

Nowadays, nearly all devices we use in our daily life are equipped with microprocessors and connected to a network, be it cars, smartphones or fridges. This ubiquity of computational power and the growing interconnectedness opens up new possibilities but also novel challenges have to be faced. In particular, limitations in communication and questions of privacy lead to limited locally available information. This lack of information is a major difficulty to be addressed and distributed algorithms are designed as a remedy to such problems. With optimization being one of the key enablers of modern technology, the idea of solving optimization problems in a distributed fashion also got in the focus of interest in the last decades. Therein, a group of computation units (often also called *agents*) cooperatively tries to solve the problem. The idea of many distributed optimization algorithms is to have each agent solve a smaller subproblem and, by sharing information over a communication network, ensure that the original problem is solved.

Many of the existing approaches to distributed optimization problems heavily rely on the assumption that the underlying communication network is of undirected nature, meaning that when one agent shares information with a second agent, the second agent will share his information as well. However, many practical problems do not have this property, e.g., due to the directed nature of sensors in networks of physical agents or due to privacy reasons. Further, the available distributed algorithms typically require strong assumptions on the structure of the optimization problem, e.g., that the objective function is a sum of individual objective functions of each agent only or that the constraints are only imposed between agents that also share information with each other. This heavily limits the class of optimization problems these algorithms are able to solve or necessitates modifications of the existing communication network. Existing approaches trying to address and relax these limitations typically utilize specifically tailored modifications of a distributed algorithm, but no general approach is known in literature.

In this thesis, we aim to take a different approach and establish a framework that allows to systematically design continuous-time distributed optimization algorithms. From a more general perspective, this thesis also provides a systematic procedure for deriving distributed algorithms from non-distributed ones. Our methodology is applicable to a quite general class of convex optimization problems under rather mild assumptions on the com-

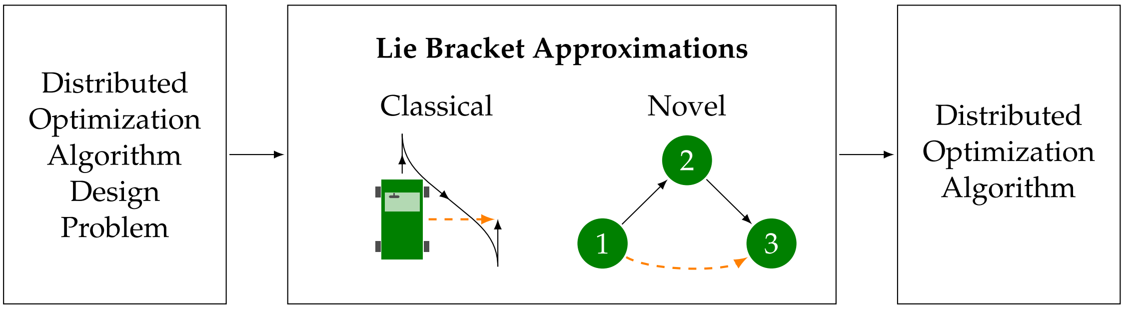

Figure 2.1. An illustration of our approach. We use Lie bracket approximations to systematically derive distributed algorithms from non-distributed ones. We employ Lie bracket approximation techniques which have proven useful to reveal and utilize hidden control directions (dashed) of nonlinear systems. Our approach further advances into that direction, where the hidden control direction is identified with a non-existing communication link (dashed).

munication structure. Following the main premise of this thesis, our approach builds upon well-established tools from systems and control theory. In particular, we use Lie bracket approximation techniques in a novel and innovative fashion. Lie bracket approximations have been extensively used in nonlinear control theory, e.g., in motion planning problems for nonholonomic mechanical systems (Z. Li & Canny, 2012, and references therein), where they enable steering a system into directions not directly accessible. We utilize Lie bracket approximations in a similar way in the sense that they enable an agent to use information somewhere available in the network but not directly accessible via a communication link (see Figure 2.1 for an illustration).

Background and Related Work. Over the last decades, distributed optimization and the closely related field of distributed control has been a very active area of research with high practical relevance, see, e.g., Boyd, Parikh, Chu, Peleato, and Eckstein (2011); Bullo, Cortés, and Martínez (2009); Zhao and Dörfler (2015) for applications. In the following, we give a brief overview of research in distributed optimization; due to the sheer amount of literature available in this area, we do not aim for a complete overview but refer the reader to the recent survey paper Nedić and Liu (2018). In distributed optimization, usually optimization problems (1.1) are considered, where, additionally, the objective function H is assumed to be a sum of individual objective functions associated to each of the n agents. More precisely, it is assumed that

$$H(z) = \sum_{i=1}^{n} H_i(z), \tag{2.1}$$

where $H_i : \mathbb{R}^p \to \mathbb{R}$ is the individual objective function associated to agent i. The goal is to solve the problem by dividing it into n smaller subproblems; each agent then solves a part of the original problem (1.1) and communicates its solution over a given communication network. Existing algorithms following such an approach can mainly be distinguished by