Abstract

In this thesis, we present a distributed model predictive control (MPC) framework for the cooperative control of multi-agent systems. The agents are dynamically decoupled but connected through a communication network and are required to perform general cooperative tasks that may go beyond classical setpoint stabilization. The cooperative task is encoded as a shared objective function that is minimized collectively by the agents. Each agent optimizes an artificial reference as an intermediate step towards the cooperative objective, along with a control input to track it. This introduces a flexible mechanism to guide the agents' interaction towards a solution to the cooperative task. Importantly, this solution is not predetermined but emerges from the optimized interactions of the agents. Various further advantages of the framework are discussed. The framework readily handles, but is not limited to, common and important cooperative tasks such as consensus and synchronization, in which agents must agree on an *a priori* unknown output value or trajectory.

The main mechanism of using artificial references is based on MPC for tracking, a tracking scheme for potentially time-varying external references. The first contribution of this thesis is the derivation of a global closed-loop transient performance bound for MPC for tracking, under practical assumptions and for the case of periodic reference signals. We show that it is essential to scale the offset cost, which penalizes the deviation of the artificial reference from the external one, with the prediction horizon. This ensures that performance improves as the prediction horizon increases. In fact, we show that infinite-horizon optimal performance is recovered by MPC for tracking in the limit as the prediction horizon tends to infinity. These results serve as an intermediate step for the performance analysis of the distributed MPC framework presented in this thesis, but they are also of independent interest as a contribution to the performance analysis of MPC for tracking. We illustrate our results using the example of a continuous stirred tank reactor.

The second contribution of this thesis is a sequential distributed MPC method for cooperative tasks defined at equilibria. We show that the proposed distributed MPC scheme asymptotically stabilizes a set corresponding to the solutions of the cooperative task. This ensures that the cooperative task is asymptotically achieved in closed loop, while individual constraints of the agents are always satisfied. Furthermore, our use of artificial references for coordination enables a scalable sequential algorithm. The scheme is applied in a numerical simulation of a team of quadrotors seeking output consensus.

The third contribution of this thesis is an iterative distributed MPC method for cooperative tasks defined on periodic trajectories. We again provide guarantees of asymptotic stability for a set corresponding to solutions that achieve the cooperative task as well as possible. Furthermore, individual constraints and coupling constraints between agents are always satisfied. We present both a variant with terminal constraints and terminal costs, and one without. For the former, we derive a closed-loop transient performance bound and show that infinite-horizon optimal performance is recovered as the prediction horizon tends to infinity. We state sufficient conditions not only in a general form to ensure flexibility in the method's applicability, but also comment on practical and intuitive design choices. We demonstrate the applicability of the scheme through numerical examples, including the control of a satellite constellation, a team of quadrotors, and surface vessels.

Chapter 1

Introduction

1.1 Motivation

Model predictive control (MPC) is a well-established modern control methodology. Central to MPC is the repeated solution of a finite-horizon optimal control problem based on a prediction of the system's future evolution from the current state. The first part of the computed optimal input is then applied to the system; the state is measured again; and the procedure is repeated by solving the next finite-horizon problem. MPC enjoys continued success as a control method due to its ability to (i) enforce hard input and state constraints in closed loop, (ii) handle nonlinear systems with multiple inputs, (iii) optimize a performance criterion with closed-loop performance guarantees, and (iv) benefit from rapid advances in computational power and algorithms for online optimization. The applicability of MPC to complex systems and control tasks is supported by a well-developed body of theory providing closed-loop guarantees, such as on constraint satisfaction, stability, and performance. For an introduction to MPC, see (Grüne and Pannek 2017; Rawlings et al. 2020).

In order to control large-scale systems comprising many individual subsystems, it is often not tractable or desirable to solve a single optimization problem. For example, a large number of subsystems leads to scalability issues; communication is only practical locally rather than with a centralized node; or failure to solve a single optimization problem can compromise the entire system. To avoid these shortcomings and to facilitate flexible interconnection structures, *distributed MPC* is used, where optimization problems are distributed among subsystems and their solution relies only on local communication, i.e. on a subset of other subsystems. Distributed MPC is applicable to a wide range of systems composed of subsystems, such as large-scale chemical plants, power networks, distribution networks, and multi-agent systems, with varying control objectives and communication topologies. Due to this complexity, no unifying framework for distributed MPC currently exists, and many distributed MPC schemes have been proposed. We review the distributed MPC schemes relevant to this thesis in Section 1.2.2 below.

In this thesis, we consider multi-agent systems comprising heterogeneous, nonlinear agents subject to individual and coupling constraints. The agents are dynamically decoupled but connected through a communication network and are required to perform general cooperative tasks that may go beyond classical setpoint stabilization. For example, such systems arise in the control of satellite constellations (Massioni et al. 2011; Sin et al. 2020; Smith and Hadaegh 2005), collective data sampling (Leonard et al. 2007), and the coordination of vehicles or robots (Ang et al. 2018; Dimarogonas and K. H. Johansson 2009; Jadbabaie et al. 2003; Olfati-Saber and Murray 2002; Ryan et al. 2004). Two particularly important abstractions of such cooperative tasks are consensus and synchronization problems, where the agents aim to agree on an output value or trajectory. These arise in multiple applications such as flocking and formation control (Olfati-Saber et al. 2007). Consensus and synchronization have been studied extensively in recent years, e.g. (Cao et al. 2013; Jadbabaie et al. 2003; Olfati-Saber and Murray 2004; Olfati-Saber et al. 2007; Seyboth et al. 2016; Wieland et al. 2011). In these references, control methods other than distributed MPC are used, which limits their applicability to nonlinear systems subject to constraints.

A common objective in MPC is the stabilization of an external, possibly time-varying, reference, which may not correspond to an equilibrium of the system and for which offline-designed parts of MPC, e.g. fixed terminal constraints, are not suitable. Recently, *MPC for tracking* has been introduced as a scheme to address such control objectives (Limón et al. 2008). The main idea in MPC for tracking is to characterize a set of admissible references that the system can track, and to optimize over this set such that the system tracks an *artificial reference* at each time step. By penalizing the offset of this artificial, or *intermediary*, reference from the external one, and under suitable conditions, the closed-loop system converges to the admissible reference that is closest to the external one. In comparison to standard MPC, it retains feasibility independently of the external reference and may allow for shorter prediction horizons. We review MPC for tracking in more detail below.

In the context of multi-agent systems, taking the view of one specific agent, the other agents imply a virtual external reference. For example, if the cooperative goal for agent A is to attain a relative distance to another agent B, agent B's position plus a corresponding offset becomes the external reference to be tracked. This virtual reference is both potentially unpredictable and time-varying, e.g.

agent B might be connected to agents that agent A cannot observe and may react to their actions. Furthermore, the virtual reference may not be attainable at the moment, e.g. because agent B is far outside the admissible state set of agent A, but agent A should still act as well as possible. Due to this insight and because it also enables a straightforward formulation of the cooperative task, we use intermediary references as the main mechanism to couple agents in our distributed MPC framework. This combines the advantages of MPC for tracking with those of distributed MPC to our novel framework for self-organized coordination of multi-agent systems.

The control of multi-agent systems must contend with several challenges. A primary concern is the presence of constraints such as actuator limits, which are intrinsic to real-world systems. Control methods that neglect these constraints often provide constraint satisfaction only for a small set of initial conditions or offer no *a priori* guarantees at all. Another critical issue is resilience to coordination failure. For example, relying on a central governor introduces a single point of failure and requires agents to maintain communication with the central entity. Moreover, multi-agent systems must remain operational under dynamic network topologies, for example when agents malfunction and leave the system, without necessitating a redesign of critical components in the control strategy. This poses a particular challenge in distributed MPC, where terminal constraints are commonly imposed to guarantee recursive feasibility and performance. While some existing approaches address these issues, they typically focus on specific cooperative tasks or dynamics. In contrast, this thesis presents a framework that supports general cooperative tasks and agent dynamics while systematically addressing the above challenges.

In the following two sections, we provide an overview of related work and specify in detail the contributions presented in this thesis.

1.2 Related work

In this section, we provide an overview of the literature related to this thesis. We begin with MPC for tracking schemes, as the use of artificial references is the main mechanism for agent coordination in the methods presented here. Furthermore, in Chapter 3, we establish transient and asymptotic performance bounds for MPC for tracking. We then turn to distributed MPC, which is a multifaceted research field due to the need to accommodate complex control objectives, communication and coordination requirements, and optimization simultaneously. This serves as an introduction to the literature relevant to the distributed MPC methods developed in Chapters 4 and 5.

In the proposed distributed MPC framework, we encode the cooperative task using a function that indicates the distance to a solution to the cooperative task. This idea of using an energy-like function is reminiscent of the use of potential functions in the design of distributed control systems for vehicle coordination; see (Dimarogonas and K. H. Johansson 2009; Ögren et al. 2004; Olfati-Saber and Murray 2002; Tsolakis and Keviczky 2021). These methods are presented for double integrator dynamics (Ögren et al. 2004; Olfati-Saber and Murray 2002) and nonholonomic kinematics (Dimarogonas and K. H. Johansson 2009; Tsolakis and Keviczky 2021).

Throughout this thesis, we refer to optimization methods as *decentralized*, following (Nedić et al. 2018), if they can be implemented on a network of processors that only need to communicate locally with each other, rather than with a central processing or aggregation unit.

1.2.1 MPC for tracking

As described earlier, MPC for tracking was introduced in (Limón et al. 2008) as a tracking MPC scheme for linear systems that integrates the behaviour of a reference governor (cf. (Bemporad et al. 1997; Gilbert et al. 1994)) and tracking MPC into a single framework. In the context of generalized predictive control, Rossiter et al. (1996) also introduced an artificial reference to relax a terminal constraint and ensure feasibility independently of reference changes. MPC for tracking has been extended to nonlinear systems in (Limón et al. 2018), using terminal costs and terminal constraints, and in (Soloperto et al. 2023), without these components.

Periodic references: The problem of tracking periodic external references is addressed in (Limón et al. 2016) for linear systems using terminal equality constraints, and in (J. Köhler et al. 2020b) for nonlinear systems using (potentially) set-based terminal constraints. In both cases, the artificial reference is allowed to be an admissible periodic trajectory with a pre-specified period length. For linear systems, Krupa et al. (2022) propose a harmonic parameterization of the artificial reference, which renders the optimization problem independent of the period length of the external reference.

Optimality: Local optimality of MPC for tracking has been examined in (Ferramosca et al. 2011; Ferramosca et al. 2009; Limón et al. 2018). These works derive conditions on the cost function penalizing the distance between the artificial reference and the external reference, such that the closed loop recovers the local optimality property of standard MPC. A global closed-loop performance analysis, previously absent from the literature, is part of the contribution of this thesis, and performance bounds are derived in Chapters 3 and 5. Recently, based on this analysis, Ehmann et al. (2025) derived global performance bounds of the MPC for tracking scheme without terminal components presented in (Soloperto et al. 2023).

Distributed MPC for tracking: MPC for tracking has also been extended to distributed MPC for linear systems in (Ferramosca et al. 2013) using a Jacobi-type decentralized optimization method similar to (Stewart et al. 2010). Razzanelli and Pannocchia (2016) enhance the scheme in (Ferramosca et al. 2013) by exploiting the graph structure connecting the subsystems to reduce the computational and communication burden. A scheme using a cyclic varying prediction horizon length is proposed in (Kögel and Findeisen 2013) to facilitate separate terminal constraints. In (Conte et al. 2016), also in the linear setting, an optimization problem with artificial references that is suitable for distributed optimization is formulated and the distributed synthesis of the used components is presented. Aboudonia et al. (2022) present a distributed MPC scheme for tracking where the terminal constraints are computed online to respect the distributed structure of the system. In (Deng et al. 2024), a distributed MPC scheme using artificial references is designed to steer linear agents towards output consensus while tracking an externally provided periodic reference. These approaches could be used to track an externally provided reference that achieves a cooperative goal, but self-organization is not explored.

See also (Krupa et al. 2024) for a recent tutorial treatment of MPC for tracking and further references.

1.2.2 Distributed MPC

As pointed out earlier, distributed MPC is a vast and diverse research area due to the necessity to consider complex control objectives, communication, coordination, and optimization together. We therefore provide only a brief overview of some methods used in distributed MPC. For a more comprehensive survey of distributed MPC, we refer to (Alvarado et al. 2011; Christofides et al. 2013; Maestre et al. 2014; Maestre and Negenborn 2014; Müller and Allgöwer 2017; Scattolini 2009).

We follow in our overview mainly the categorization used in (Müller and Allgöwer 2017) into iterative schemes, sequential schemes, schemes using techniques for robustness, and schemes based on consistency constraints.

Iterative distributed MPC methods: A major part of the MPC schemes are iterative methods based on distributed or decentralized optimization, where a central optimization problem is solved by iteratively computing solutions to local optimization problems and communicating iterates. Examples are (Conte et al. 2016; Fawal et al. 1998; Ferramosca et al. 2013; Giselsson et al. 2013; Kögel and Findeisen 2012; Maestre et al. 2011; Rosenfelder et al. 2022; Scheu and Marquardt 2011; Shorinwa and Schwager 2024). These methods generally recover centralized performance and guarantees if sufficient iterations are carried out, but require multiple communication steps in each time step. Some work therefore pays special attention to iterative distributed MPC schemes based on suboptimal solutions, e.g. (Bestler and Graichen 2019; Dang Doan et al. 2017; Giselsson and Rantzer 2014; J. Köhler et al. 2019; Stewart et al. 2011; Stomberg et al. 2025a; Wu et al. 2025). The distributed MPC framework presented in Chapter 5 is iterative.

Sequential distributed MPC methods: An alternative to iterative distributed MPC schemes are sequential ones in which the agents optimize a local optimization problem in sequence. Data is then only communicated before or after optimization and fixed during optimization. Compared to iterative methods, this can reduce the amount of necessary communication considerably, but may potentially scale badly with the number of agents. Examples of such distributed MPC schemes are (Grüne and Worthmann 2012; Richards and How 2007; Trodden and Richards 2013). The distributed MPC framework presented in Chapter 4 is sequential.

Further non-iterative distributed MPC methods: There also exist further non-iterative distributed MPC schemes that rely on emergency manoeuvres and fallback solutions (Borrelli et al. 2005), treat neighbours as disturbances and apply robustness techniques in the design or analysis (Farina and Scattolini 2012; Franco et al. 2008; Lucia et al. 2015; Riverso et al. 2013), or base their design or analysis on *consistency constraints* that ensure agents remain close to communicated plans (Dunbar and Murray 2006; Keviczky et al. 2006; M. Köhler et al. 2022; Wiltz et al. 2025; Zhao and Ding 2015).

The majority of distributed MPC schemes, including those cited above, consider stabilization of a pre-specified equilibrium or tracking of an externally supplied reference trajectory. However, this may be an unsuitable control objective for multi-agent systems performing a cooperative task. Even though the cooperative task may be well-defined, its solution is not known *a priori* and should emerge from the agents' coordinated actions. The framework presented in this thesis provides such an emergence.

Distributed MPC methods for consensus and synchronization: As already mentioned above, consensus and synchronization are important examples of cooperative tasks and have received significant attention in recent years. In (Ferrari-Trecate et al. 2009), a distributed MPC scheme for consensus of single and double integrators subject to input constraints is proposed, where the main arguments are based on geometric properties of the integrator dynamics. Distributed MPC schemes for consensus of multi-agent systems without constraints are proposed in (Zhan and X. Li 2013) for single integrator systems with fixed and switched topologies, in (Zhou and S. Li 2015) for double integrator systems, in (H. Li et al. 2016) for scalar systems, and in (H. Li and Yan 2015) for homogeneous linear systems. In (Zhou and S. Li 2017), a distributed MPC scheme for flocking of unconstrained double integrator agents is presented. Consensus for multi-agent systems comprising double integrators subject to input constraints is treated in (Cheng et al. 2015). Lyu et al. (2021) propose a distributed MPC scheme for flocking of double integrators subject to state and input constraints that ensures collision avoidance. For linear systems subject to input constraints, (Lee et al. 2011) propose a scheme where a consensus trajectory is computed using any consensus algorithm and then supplied to a tracking MPC. For a similar setup, (B. Johansson et al. 2006; B. Johansson et al. 2008) propose a distributed MPC scheme where first a central optimization problem is solved in distributed fashion, and then its solution is used as a setpoint for the agents. This scheme is extended in (Keviczky and K. H. Johansson 2008) to the case where the consensus point is negotiated online in each time step. The eventual consensus point, which needs to be reached at the end of the finite prediction horizon is a decision variable in the central optimization problem. Convergence and stability are analysed for the case when the central optimization problem is solved to optimality in each time step and the case when it is prematurely stopped. In (Gautam et al. 2014), a robust distributed MPC scheme for consensus of linear time-varying uncertain systems is presented. We proposed a sequential distributed MPC scheme for consensus of general linear and heterogeneous agents subject to input and state constraints in (Hirche et al. 2020), which is a special variation of the framework presented in Chapter 4. Synchronization of linear multi-agent systems, that is consensus with respect to a leader agent, is considered in (Dreke et al. 2023). Here, the cost function in the MPC's optimization problem penalizes deviation from an unconstrained synchronization control law, and convergence under the assumption of recursive feasibility is analysed. A similar problem is considered in (Duan et al. 2024), which uses the same cost functions as (Dreke et al. 2023) and combines it with a terminal set constructed based on invariant family of sets to achieve also recursive feasibility. A considerable limitation of these methods is their reliance on linear systems, whereas many real-world systems are intrinsically nonlinear. In contrast, the framework developed in this thesis accommodates nonlinear dynamics and can handle tasks such as consensus and synchronization.

Hierarchical distributed MPC for coordination: Farina et al. (2015) present a hierarchical distributed MPC scheme for the coordination of autonomous robots, addressing tasks such as formation control, collision avoidance, and coverage. The proposed scheme has two layers. At an upper level, a suitable optimization problem is solved to obtain reference trajectories. At a lower level, a robust MPC problem is solved by the robots. The scheme is applicable to linear systems subject to additive disturbances. Collision and obstacle avoidance constraints are reformulated as affine constraints to yield a quadratic program. A sequential distributed MPC scheme is presented in (P. N. Köhler et al. 2018) for linear agents that aim to minimize individual objective functions while achieving a cooperative task asymptotically. In this scheme, agents already take control actions while the *a priori* unknown equilibrium fulfilling the cooperative task is negotiated externally, and intermediate steps are communicated to the agents. The agents use time-varying terminal constraints, which are updated to track these intermediate steps. In (Chanfreut et al. 2022), a clustering approach is presented for potentially coupled linear subsystems driven to a target set. The subsystems can dynamically merge into clusters that locally solve optimization problems, treating dynamic connections to other clusters as disturbances. The individual optimization problems are based on a tube-based robust extension of (Limón et al. 2008), as presented in (Limón et al. 2010). Reorganization of the clusters is permitted only if recursive feasibility can be ensured, and convergence of the closed loop to the target set is shown.

Distributed MPC for specific cooperative tasks: For some specific cooperative tasks, bespoken distributed MPC schemes have been developed. In (Dunbar and Caveney 2012), string stability of vehicle platoons with nonlinear dynamics is established by penalizing the deviation from previously communicated trajectories. Zheng et al. (2017) propose a non-iterative distributed MPC scheme using consistency constraints for heterogeneous vehicle platoons. They argue that the agents do not know the desired set point of the platoon *a priori*, since not everyone is connected with the platoon's leader. A terminal equality constraint with respect to the average of the neighbours' outputs is used to promote asymptotic stability of the platoon. Carron and Zeilinger (2020) leverage artificial references for driving a nonlinear multi-agent system to the solution of a coverage problem. In the scheme, each agent is supplied with an external reference that corresponds to a centroid in a Voronoi configuration. In (Rickenbach et al. 2024), on the one hand the scheme of (Carron and Zeilinger 2020) is extended by including collision avoidance constraints, and on the other hand the computation of the next coverage configuration is integrated directly in the MPC's optimization problem. The scheme, which is also based on artificial references, is suitable for covering a potentially unknown environment using a multi-agent system with nonlinear dynamics. Collisions are avoided by prudent updates of the local target sets. These methods focus on either specific dynamics or specific cooperative tasks, but do not provide a framework for general cooperative tasks.

General cooperative tasks: The sequential distributed MPC framework proposed in (Müller 2014; Müller et al. 2012) for agents with nonlinear dynamics can be applied for more general cooperative tasks such as consensus and synchronization. However, the scheme relies on centralized terminal regions that must be specifically designed to match the cooperative task. For the example of consensus, the terminal costs, terminal constraints, and terminal control laws proposed in (Müller 2014; Müller et al. 2012) are also designed through a centralized procedure. This limits the flexibility of the scheme, for instance, when the cooperative task changes or when agents join or leave the system. In addition, it remains an open question how the terminal components can be designed for cooperative tasks beyond consensus and synchronization. Ferranti et al. (2022) present a method for multi-agent trajectory coordination based on distributed MPC. They propose two decentralized optimization algorithms to solve a central trajectory optimization problem for nonlinear agents subject to collision avoidance constraints. The first algorithm applies an alternating direction method of multipliers (ADMM) for nonconvex optimization proposed in (Themelis and Patrinos 2020). The second algorithm extends the first to asynchronous optimization by leveraging the predictions generated in MPC. While the proposed method is suitable for a general class of agents and cooperative tasks with pre-specified solutions, it focuses on solving the key optimization problem, assumed to be always feasible, and on providing convergence guarantees for the algorithms, rather than analysing the dynamic evolution of the closed-loop system. In contrast, our framework focuses on coordinating agents over time to achieve a self-organized solution to a cooperative task, and provides guarantees on closed-loop stability and recursive feasibility.

In summary, there is little work on distributed MPC methods for nonlinear multi-agent systems that perform general cooperative tasks. This thesis proposes a distributed MPC methodology to close this gap. We detail its contributions in the next section.