Trajectory Optimization for Nonlinear Multi-Agent Systems using Decentralized Learning Model Predictive Control*

Edward L. Zhu\textsuperscript{1}, Yvonne R. Stürz\textsuperscript{2}, Ugo Rosolia\textsuperscript{2}, and Francesco Borrelli\textsuperscript{1}

Abstract—We present a decentralized trajectory optimization scheme based on learning model predictive control for multi-agent systems with nonlinear decoupled dynamics under separable cost and coupled state constraints. By performing the same task iteratively, data from previous task executions is used to construct and improve local time-varying safe sets and an approximate value function. These are used in a decentralized MPC problem as the terminal sets and terminal cost function. Our framework results in a decentralized controller, which requires no communication between agents over each iteration of task execution, and guarantees persistent feasibility, closed-loop stability, and non-decreasing performance of the global system over task iterations. Numerical experiments of a multi-vehicle collision avoidance scenario demonstrate the effectiveness of the proposed scheme.

I. INTRODUCTION

In this paper, we study the problem of decentralized Model Predictive Control (MPC) for dynamically decoupled multi-agent systems under separable cost and coupled state constraints. Multi-agent systems typically exhibit inter-agent coupling, which can be expressed as constraints on the global system. MPC is a well-studied approach to the control of such constrained systems and can be applied in a global manner for centralized control of multi-agent systems with a small number of agents. However, as the number of agents increases, centralized approaches typically become intractable in practice due to limitations in computational power and communication capacities [1]. This gives rise to decentralized and distributed MPC schemes, which leverage the inherent parallelizable structure of multi-agent systems to reduce the required computational effort. Feasibility and stability of MPC are typically obtained using a terminal cost function and terminal constraints in the MPC design [2], which we refer to as terminal components. However, synthesis of these terminal components for the control of nonlinear multi-agent systems can be challenging.

The primary advantage of decentralized MPC over its distributed counterpart is the lower communication demand. While only local information is used in the control of each agent in decentralized schemes, additional communication between agents is required to obtain a control action in distributed methods [3]–[9]. A literature review on distributed MPC is extensive and goes beyond the scope of this conference paper. In decentralized MPC, local controllers are synthesized for each agent, where feasibility and stability properties are attained via robustness of the controller against coupling to other agents. In prior work, conditions for the stability of decentralized MPC for nonlinear systems are investigated. In [10], [11], this is attained via terminal cost synthesis and an online supervisory scheme for modifying the decoupling structure to meet these conditions without destabilizing the system. [12] achieves this by bounding the prediction error of neighboring agents’ states. However, this method assumes the singleton terminal set of the origin, which limits the controller’s domain of attraction. In addition, neither method can deal with coupling state constraints between the agents.

Iterative approaches to the control of multi-agent systems have also been proposed. Of such methods, Sequential Convex Programming (SCP) [13] is similar to the approach proposed in this paper. SCP solves a non-convex optimization problem by successively forming convex approximations about previous solutions. In [14], SCP was used to generate collision-free trajectories for multiple quadcopters in a centralized manner. This was extended into a decentralized formulation in [15], which showed improvements in computational tractability. However, in these works, SCP is a heuristic method and may fail to find a feasible solution. In addition, SCP optimizes over the entire trajectory, which contrasts with the receding horizon approach taken in this work, and may be computationally challenging for long time horizons or fine time discretizations.

In this paper, we propose a decentralized approach to trajectory optimization for nonlinear multi-agent systems. By performing the same task over iterations, we collect data on the agents’ closed-loop behavior to successively improve the construction of terminal components for the local controllers. In particular, as we improve an estimate of the terminal cost (an approximate value function) and expand the terminal constraint set (the domain of this function), we are able to iteratively improve closed-loop performance of the multi-agent system while maintaining feasibility and stability guarantees. The contribution of this work is twofold. We extend the method of Learning Model Predictive Control (LMPC) from [16] to the multi-agent case for dynamically decoupled agents under separable cost and coupled state constraints. In particular, we first propose a procedure for synthesizing the MPC terminal components using data from previous iterations of task execution. We then show that the resulting decentralized LMPC has the properties of persistent feasibility, stability, and non-decreasing performance over...
iterations. We demonstrate the effectiveness of the decentralized method with a numerical example in the context of multi-vehicle collision avoidance, where we observe a significant reduction of computational effort compared to a centralized approach.

II. PRELIMINARIES

A. Notation

For a system of $M$ agents, the set of indices $\{1, \ldots, M\} \in \mathbb{N}_+$ is denoted as $\mathcal{M}$. $\mathbb{N}$ and $\mathbb{N}_+$ denote the set of non-negative and positive natural numbers, respectively, and $\mathbb{N}_a$ denotes the set of non-negative natural numbers up to and including $a \in \mathbb{N}$. Finally, \( \leq \) denotes an element-wise inequality.

B. System Description

Consider the global nonlinear time-invariant discrete-time system composed of $M$ agents

\[ x_{t+1} = f(x_t, u_t), \quad (1) \]

where the global state and input vectors at sampling time $t \in \mathbb{N}$ are formed by stacking those from each agent into a single column, i.e. $x_t = \text{col}_{i \in \mathcal{M}}(x_{i,t}) = [x_{1,t}^\top, \ldots, x_{M,t}^\top]^\top \in \mathbb{R}^n$ and $u_t = \text{col}_{i \in \mathcal{M}}(u_{i,t}) \in \mathbb{R}^m$, where $x_{i,t} \in \mathbb{R}^{n_i}$ and $u_{i,t} \in \mathbb{R}^{m_i}$. Each agent in the global system is subject to local state and input constraints,

\[ x_{i,t} \in \mathcal{X}_i, \quad u_{i,t} \in \mathcal{U}_i, \quad \forall t \in \mathbb{N}, \forall i \in \mathcal{M}. \quad (2) \]

These local constraint sets are assumed to be closed, compact, and include the origin in their relative interiors. The global system is additionally subject to coupling constraints on the system state,

\[ g(x_t) \leq 0, \quad \forall t \in \mathbb{N}. \quad (3) \]

We assume that the agents are dynamically decoupled and locally stabilizable, which means that we can write (interchangeably with \([\mathcal{H}]\)), for all agents $i \in \mathcal{M}$, the local dynamics as

\[ x_{i,t+1} = f_i(x_{i,t}, u_{i,t}). \quad (4) \]

C. Control Objective

The objective is to design a controller which drives the system state to a goal state $x_F$ by solving the following infinite horizon optimal control problem for the global system

\[ \min_{u_\infty} \sum_{t=0}^\infty h(x_t, u_t) \quad (5a) \]

subject to

\[ x_{t+1} = f(x_t, u_t), \quad \forall t \in \mathbb{N} \quad (5b) \]

\[ x_0 = x_S \quad (5c) \]

\[ x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad \forall t \in \mathbb{N} \quad (5d) \]

\[ g(x_t) \leq 0, \quad \forall t \in \mathbb{N}, \quad (5e) \]

where $x_S = \text{col}_{i \in \mathcal{M}}(x_{i,S})$ is the initial condition of the system and the state and input constraint sets in \((5d)\) are the Cartesian products of the local constraint sets in \((2)\). We assume that the stage cost in \((5a)\) is separable over agents, i.e. $h(x_t, u_t) = \sum_{i \in \mathcal{M}} h_i(x_{i,t}, u_{i,t})$. In addition each local stage cost is continuous and satisfies

\[ h_i(x_{i,F}, 0) = 0 \text{ and } h_i(x, u) > 0 \forall x \in \mathbb{R}^{n_i} \setminus \{x_F\}, \quad u \in \mathbb{R}^{m_i} \setminus \{0\}. \quad (6) \]

The goal state $x_F = \text{col}_{i \in \mathcal{M}}(x_{i,F})$ is assumed to be a feasible equilibrium state of \((\mathcal{H})\). We assume that a locally optimal solution to Problem \((5)\) exists and denote it as $x^*_\infty = \{x_0^0, x_1^0, \ldots, x_t^0, \ldots\}$ and $u^*_\infty = \{u_0^0, u_1^0, \ldots, u_t^0, \ldots\}$.

D. Learning Model Predictive Control

In this section, we briefly introduce and review the key concepts of LMPC \([16]\), which are extended upon in this work. LMPC was proposed as an iterative trajectory optimization method for single-agent nonlinear dynamical systems performing iterative tasks. In particular, the method provides a data-driven approach to terminal set and cost synthesis. We assume that an initial feasible input sequence $u^0 = \{u_0^0, u_1^0, \ldots, u_t^0, \ldots\}$ and closed-loop state trajectory $x^0 = \{x_0^0, x_1^0, \ldots, x_t^0, \ldots\}$ with $\lim_{t \to \infty} x_t^0 = x_F$ exists for \((\mathcal{H})\) and is available at iteration $q = 0$. We note that for iterative tasks, the initial condition of the system is assumed to be the same over iterations, i.e. $x_0^q = x_S, \forall q \in \mathbb{N}$.

LMPC solves a Finite Horizon Optimal Control Problem (FHOCP), which approximates \([\mathcal{H}]\), in a receding horizon fashion. Given the initial condition $x$ at time $t$ of iteration $q$, the FHOCP is defined as

\[ J_N^q(x, t) = \min_{u_{i,t}^N} \sum_{k=t}^{N-1} h(x^q_{k|t}, u^q_{k|t}) + V^{-1}(x^q_{N|t}) \quad (7a) \]

subject to

\[ x^q_{k+1|t} = f(x^q_{k|t}, u^q_{k|t}), \quad \forall k \in \mathbb{N}_N \quad (7b) \]

\[ x^q_0 = x \quad (7c) \]

\[ x^q_{k|t} \in \mathcal{X}, \quad u^q_{k|t} \in \mathcal{U}, \quad \forall k \in \mathbb{N}_N \quad (7d) \]

\[ g(x^q_{k|t}) \leq 0, \quad \forall k \in \mathbb{N}_N \quad (7e) \]

\[ x^q_{N|t} \in \mathcal{SS}^{q-1}, \quad (7f) \]

where $x^q_{k|t}$ and $u^q_{k|t}$ denote the decision variables of the predicted state and input at the sampling time $t+k$ of iteration $q$. The first input $u^q_0$ is then applied to the system \((\mathcal{H})\).

The terminal set at iteration $q-1$ in \((7f)\), called a sampled safe set, is defined as

\[ \mathcal{SS}^{q-1} = \left\{ \bigcup_{p \in Q^{q-1}} \bigcup_{t=0}^\infty x_p^t \right\}, \quad (8) \]

where $Q^{q-1} = \{p \in \mathbb{N}_{q-1} : \lim_{t \to \infty} x_p^t = x_F\}$ is the set of iteration indices up to iteration $q-1$ where the goal state $x_F$ was successfully reached. The sampled safe set collects the closed-loop state trajectories from previous successful iterations, which implies that at iteration $q$, $\forall x \in \mathcal{SS}^q$, there exists a known and feasible sequence of inputs, with $x_0 = x$, such that $f(x, u) \in \mathcal{SS}^q, \forall t \in \mathbb{N}$. We note that by construction, $\mathcal{SS}^{q-1} \subseteq \mathcal{SS}^q$, and $\mathcal{SS}$ is a control invariant set.

For the terminal cost function in \((7a)\), an approximate value function is constructed which returns the minimum
cost-to-go (over iterations 0 to \( q - 1 \)) from each state in the safe set. The cost-to-go from the state at time \( t \) along the closed-loop trajectory \( x^q \) with corresponding input sequence \( u^q \) is defined as \( J_{\infty}^q(x^q, t) = \sum_{k=t}^{\infty} h(x^q_k, u^q_k) \). The approximate value function at iteration \( q - 1 \) is then

\[
V^{q-1}(x) = \begin{cases} 
\min_{(p,t) \in \mathcal{F}^{q-1}(x)} J_{\infty}^p(x, t) & \text{if } x \in \mathcal{SS}^{q-1} \\
+\infty & \text{otherwise}
\end{cases}
\]  

(9)

where \( \mathcal{F}^{q-1}(x) = \{(p,t) \in \mathbb{N}_{q-1} \times \mathbb{N} : x^p_t = x \} \) and \( x^p \in \mathcal{SS}^{q-1} \) returns the set of iteration and time index pairs for the states in previous trajectories which are equal to \( x \). Now, using the approximate value function in (9) as the FHOCP terminal cost function, for a locally optimal solution \( x^*_{i,N} \) and \( u^*_{i,N} \), we obtain the locally optimal LMPC cost, \( J_{N}^q(x^*_i, t) = h(x^*_i, u^*_{i,q}) + \sum_{k=1}^{N-1} h(x^*_{i,k}, u^*_{i,q}) + V^{q-1}(x^*_{N|i}) \). It can be shown that \( J_{N}^q(\cdot, t) \) is a Lyapunov function for the closed-loop trajectory with the equilibrium \( x_F \).

Using the properties of the constructed terminal components, the resulting scheme guarantees persistent feasibility and closed-loop stability of the FHOCP. Non-decreasing performance and convergence to a locally optimal solution can also be shown, under mild conditions, over iterations of task execution. Details of the proofs can be found in [16].

III. PROBLEM FORMULATION

In this section, we extend the idea of LMPC from Section II-D to the multi-agent case. Specifically, our formulation leverages data from previous iterations of task execution to synthesize decoupled controllers for each agent by designing local FHOCPs. This allows for an entirely decentralized control scheme for the multi-agent system at each iteration. In the following, we describe the synthesis of the local terminal components, namely the time-varying sampled safe sets and approximate value function. We then present the resulting local FHOCPs, which are solved in a receding horizon manner. Combining these elements, we arrive at a decentralized LMPC procedure for trajectory optimization of multi-agent systems.

A. Time-Varying Sampled Safe Sets and Global Constraint Decomposition

In this part, we address two challenges posed by a decentralized receding horizon approach. Namely 1) how global constraint satisfaction can be enforced through decentralized local constraint satisfaction, and 2) how infinite-time feasibility can be maintained in a receding horizon implementation of (5), particularly when time-varying constraints are introduced.

To address 1), we use the following assumption. Notice that the original global time-invariant constraints are transformed into local time-varying constraints.

Assumption 1. At iteration \( q \), there exists a time-varying local decomposition of the global constraints: \( g^q_{i,t}() \), \( \forall t \in \mathbb{N}, \forall i \in \mathcal{M} \), which can be constructed from feasible trajectories of the global system, such that at each sampling time \( t \), joint local constraint satisfaction is sufficient for global constraint satisfaction, i.e. \( g^q_{i,t}(x_{i,t}) \leq 0, \forall i \in \mathcal{M} \implies g(x_t) \leq 0 \).

Remark 1. Given a feasible trajectory \( x^q_i \), we may always obtain the decomposition \( g^q_{i,t}(x_i) = \|x_i - x^q_{i,t}\| \) for each trajectory point in \( x^q_i \), which satisfies the condition in Assumption 1. In Section V we present a technique for constructing \( g^q_{i,t}(\cdot) \) in a less conservative manner.

Regarding 2), infinite-time constraint satisfaction of FHOCPs is often addressed using control invariant sets. This was done in [16] via (8), which, recall, are control invariant sets by definition. We would like to achieve the same goal of using data to construct sets, which grant the same properties to the decentralized scheme. However, in (5) and due to the constraint decomposition in Assumption 1, we include time-varying constraints for which the classical definition of set invariance does not apply. For this reason, we do not collect all recorded states of the global system, which converge to \( x_F \), into a single invariant set as in [8]. Instead, we interpret the states of agent \( i \) at time \( t \) of previous successful iterations as a sampled subset of the reachable set to \( x_{i,F} \) from time \( t \) to infinity. As such, for each agent \( i \in \mathcal{M} \) at iteration \( q \), we propose to construct time-varying sampled safe sets \( \mathcal{SS}^q_{i,t} \) from data previous iterations of task execution, which have the property of reachability to \( x_{i,F} \). The following assumption allows us to start with a non-empty safe set at the first iteration.

Assumption 2. At iteration \( q = 0 \), an initial feasible input sequence and corresponding state trajectory, which converges to the goal state, exists for each agent (4). Denote this input sequence and state trajectory for agent \( i \in \mathcal{M} \) as \( u^i_{0} \) and \( x^i_{0} \). A decomposition of the global constraint exists such that \( g^0_{i,t}(x^0_{i,t}) \leq 0, \forall x^0_{i,t} \in x^i_{0} \).

Remark 2. Assumption 2 is reasonable as one may provide initial feasible trajectories through demonstration or by employing a highly conservative controller.

Let \( (x, u) \) be the dataset which collects the state trajectories \( x^0_{i}, \ldots, x^q_{i} \) and input sequences \( u^0_{i}, \ldots, u^q_{i} \). Then, for each agent \( i \) at sampling time \( t \) of iteration \( q \), we construct the candidate sampled safe sets as

\[
\mathcal{SS}_{i,t}^q = \bigcup_{p \in \mathcal{I}} \left\{ x_{i,k}^p \in x : k \in \mathcal{T}_i \right\},
\]  

(10)

where \( \mathcal{T}_i \) and \( \mathcal{I} \) contain the time and iteration indices of those previous data points which are selected for the time-varying sampled safe sets. Here, \( \mathcal{I} \) is constructed by choosing the design parameter \( q \geq 1 \) such that we collect the indices of the \( q \) most recent successful iterations. We construct the sliding range of sampling times \( \mathcal{T}_i = \{ \max(t - \ell, 0), \ldots, t, \ldots, t + \bar{\ell} \} \) by choosing the parameters \( \ell \geq 0 \) and \( \bar{\ell} \geq 0 \). By Assumption 1 using the dataset \( (x, u) \), we also construct a decomposition of the global constraint such that

\[
g^q_{i,t}(x_i) \leq 0, \forall x_i \in \mathcal{SS}_{i,t}^q, \forall i \in \mathcal{M}, \forall t \in \mathbb{N}.
\]  

(11)
Remark 3. How the constraint decomposition is constructed depends on the specific problem at hand. In the multi-vehicle collision avoidance example shown in Section VII, we propose a procedure to construct hyperplanes which separate the position states of the sampled safe sets for pairs of agents. If a construction satisfying (11) cannot be found, then we begin to successively shrink the sets \( T_i \) and \( \bar{T}_i \) until we arrive at a valid constraint decomposition and the corresponding sampled safe sets.

The time-varying sampled safe sets \( S_{i,t}^q \) collect the state trajectory points from previous successful iterations for which there exists a trajectory to the goal state that satisfies the global constraints. We show in Section IV that the sampled safe sets are a non-empty family of reachable sets to \( x_{i,F} \). In fact, the sampled safe set synthesized for agent \( i \) at time \( t \) and iteration \( q \) will always contain \( x_{i,t}^q \), where \( q \) is the index of the last successful iteration.

Remark 4. In practice, there will be a finite number of \( t_i^q + \Delta \) sampled safe sets for each agent at each iteration, where \( t_i^q \) is the trajectory length for agent \( i \) at iteration \( q \). We can therefore replace \( \bar{T}_i \) with the range of sampling times \( T_i^q = \{ \max(t - \Delta, 0), \ldots, t, \ldots, \min(t + \Delta, t_i^q) \} \). It is straightforward to extend the sampled safe sets by an arbitrary number using \( S_{i,t}^q \) for the input \( u_{i,t} = 0, \forall t > t_i^q + \Delta \).

B. Value Function Approximation

For an input sequence \( u_i^q \) and closed-loop state trajectory \( x_i^q \) of agent \( i \) at iteration \( q \), we define the cost-to-go from the state \( x_{i,t}^q \) at time \( t \) along the closed-loop trajectory to be

\[
J_{i,\infty}^q (x_{i,t}^q, t) = \sum_{k=t}^{\infty} h_i(x_{i,k}^q, u_{i,k}^q).
\]

The iteration cost for agent \( i \) at iteration \( q \) can then be written as \( J_{i,\infty}^q (x_i^q, s, 0) \), recalling that the system is initialized at the same state \( x_s \) for each iteration \( q \). This leads to the definition of the approximate value function \( V_{i,t}^q (\cdot) \) at time \( t \) over the sampled safe set \( S_{i,t}^q \) as

\[
V_{i,t}^q (x_i, t) = \left\{ \begin{array}{ll}
\min_{(p,k) \in F_{i,t}^q (x_i)} P_{i,\infty}^p (x_i, k), & \text{if } x_i \in S_{i,t}^q \\
+\infty, & \text{otherwise,}
\end{array} \right.
\]

where \( F_{i,t}^q (x_i) \) is defined in the same way as in (9). Thus, for a state \( x_i \) whose value is equal to some state in \( S_{i,t}^q \), \( V_{i,t}^q (x_i, t) \) returns the minimum cost-to-go of the remainder of the trajectories in \( S_{i,k}^q \) for \( k \geq t \).

C. The Finite Horizon Optimal Control Problem

Synthesis of the terminal components is achieved using the function \( \text{synthesizeFHOC} \) as summarized in Algorithm 1, which acts as a global coordinator between iterations of task execution in our decentralized framework. As such, after performing the task at iteration \( q - 1 \), we obtain for agent \( i \) the decomposition of the global constraint \( g_{i,t}^{q-1} (\cdot) \), the sampled safe sets \( S_{i,t}^{q-1} \), and the approximate value function \( V_{i,t}^{q-1} (\cdot, t) \), which are constructed using data from iterations up to and including \( q - 1 \).

To obtain the control action for agent \( i \) at sampling time \( t \) of iteration \( q \), we solve the following decoupled FHOCP with horizon length \( N \) and initial condition \( x_i \).

\[
\min_{u_{i,t}} \sum_{k=0}^{N-1} h_i(x_{i,k}^q, u_{i,k}^q) + V_{i,t}^{q-1} (x_{i,N|t}^q, t + N)
\]

s.t.

\[
\begin{align*}
& x_{i,k+1}^q = f_i(x_{i,k}^q, u_{i,k}^q) \quad \forall k \in \mathbb{N}_{N-1} \\
& x_{i,0}^q = x_i \\
& x_{i,k}^q \in X_i, u_{i,k}^q \in U_i \quad \forall k \in \mathbb{N}_{N-1} \\
& g_{i,t+k}^{q-1} (x_{i,k}^q) \preceq 0 \quad \forall k \in \mathbb{N}_{N-1} \\
& x_{i,N|t}^q \in S_{i,t}^{q-1}.
\end{align*}
\]

we denote the FHOCP as \( P_{i,N}^q (x_i, t) \), where (14b) and (14d) represent the system dynamics and initial condition, respectively. The local state and input constraints are given in (14d). (14e) enforces satisfaction of the decomposed constraint for each agent, which recall, is sufficient for global constraint satisfaction. Finally, (14f) ensures that the terminal state is a member of the time-varying sampled safe set \( S_{i,t}^{q-1} \). We denote the locally optimal value of the FHOCP cost in (14a) as \( J_{i,N}^q (x_i, t) \).

Let \( u_{i,t}^q (x_i) = \{ u_{i,0}^q (x_i), \ldots, u_{i,N-1}^q (x_i) \} \) denote the input sequence which minimizes (14) for initial state \( x_i \) at sampling time \( t \), and \( x_i^q = \{ x_{i,0}^q, \ldots, x_{i,N|t}^q \} \) be the corresponding state trajectory beginning at \( x_{i,0}^q = x_i \). In typical receding horizon fashion, for each agent \( i \in \mathcal{M} \), the first element of \( u_{i,t}^q (x_i) \) is applied to system (4), which defines the state feedback policy

\[
u_{i,t}^q = \kappa_i^q (x_i, t) \equiv u_{i,0}^q (x_i),
\]

with \( x_i = x_{i,t}^q \). This results in the closed-loop state trajectory \( x_i^q = \{ x_{i,0}^q, x_{i,1}^q, \ldots, x_{i,t}^q, \ldots \} \) and input sequence \( u_i^q = \{ u_{i,0}^q, u_{i,1}^q, \ldots, u_{i,t}^q, \ldots \} \) for agent \( i \) at iteration \( q \).

Remark 5. In practical applications, the closed-loop trajectories are of finite length. As in [16], and is common in literature, we adopt the infinite time formulation for simplicity.

Remark 6. Due to the construction of the sampled safe sets as collections of discrete trajectory points, \( P_{i,N}^q (x_i, t) \) is a mixed integer nonlinear program, which can be computationally intensive to solve. However, the problem structure can be exploited for computational efficiency, e.g. by leveraging parallel computations. Certain nonlinear systems may also admit a convex relaxation of the sampled safe set and approximate value function while maintaining performance guarantees [17].

D. Decentralized Learning Model Predictive Control

The resulting iterative LMPC scheme for the multi-agent system is described in Algorithm 2, where the for loop over agents from lines 4 to 14 may be executed in an entirely decentralized manner with no communication between agents.
Algorithm 1: synthesizeFHOCP

Input: $q$, $(x, u, t, \theta = \{q, \bar{t}, \ell\})$

1. $q \leftarrow q$
2. $\mathcal{I} \leftarrow \{\max(q - q), \ldots, q\}$
3. $\ell \leftarrow \max\{\bar{t} : p \in \mathcal{I}\}$
4. for $t \leq \ell + t$ do
   5. for $i \in \mathcal{M}$ do
      6. $T_{i,t} \leftarrow \{\max(t - t, 0), \ldots, t, \ldots, \min(t + t, t)\}$
      7. $\mathcal{S}S_{i,t} \leftarrow \bigcup_{p \in \mathcal{Z}} \{x^p \in k: k \in T_{i,t}\}$
   end
9. $(g_{i,t}(\cdot)) \in \mathcal{M} \leftarrow$ obtain decomposition according to Assumption 1
10. end
11. while Any constraint in $g_{i,t}(x^q)$ is violated for any $x^q \in \mathcal{S}S_{i,t}$, $i \in \mathcal{M}$, $t \leq \ell + t$ do
12.     $q \leftarrow q - 1$
13.     if $q = 0$ then
14.         $\ell \leftarrow \max(\ell - 1, 0), \ell \leftarrow \max(\ell - 1, 0)$
15.     end
16. end
17. Repeat lines 2-10 with updated iteration and time ranges;
19. $V_i^q(\cdot, t) \leftarrow$ compute as in (12) and (13) \(\forall x^q \in \mathcal{S}S_{i,t}\),

Output: $\{\mathcal{S}S_{i,t}, V_i^q(\cdot, t), g_{i,t}(\cdot)\}$

Algorithm 2: Decentralized Learning Model Predictive Control

Input: $\epsilon, Z, (x^0, u^0, t^0, I)$

1. $x \leftarrow \{x^0\}$, $u \leftarrow \{u^0\}$, $t \leftarrow \{t^0\}$
2. $\mathcal{S}S_{i,t}^q, V_i^q(\cdot, t), g_{i,t}^q(\cdot) \leftarrow$ synthesizeFHOCP($x, u, t, \theta$);
3. for $q \in \mathbb{N}_Z$ do
4.     for $i \in \mathcal{M}$ in parallel do
5.         $t \leftarrow 0$, $x^q_{0} \leftarrow x_i, S$, $x^q_i \leftarrow \{x_i, S\}$, $u^q_i \leftarrow \emptyset$
6.         while $\|x^q_{i,t} - x_i,F\| > \epsilon$ do
7.             $k^q_i(t) \leftarrow$ solve $P^q_i(x^q_{i,t}, t)$
8.             $x^q_{i,t+1} \leftarrow$ apply $u^q_i(t) = k^q_i(x^q_{i,t}, t)$ and measure state;
9.             $x^q_i \leftarrow x^q_i \cup \{x^q_{i,t+1}\}$, $u^q_i \leftarrow u^q_i \cup \{u^q_i\}$
10.         $t \leftarrow t + 1$
11.     end
12.     $x \leftarrow x \cup \{x^q_i\}$, $u \leftarrow u \cup \{u^q_i\}$
13.     $t_i^q \leftarrow t$
14. end
15. $t \leftarrow t + \{t_i^q\} \in \mathcal{M}$;
16. $\{\mathcal{S}S_{i,t}^q, V_i^q(\cdot, t), g_{i,t}^q(\cdot)\} \leftarrow$ synthesizeFHOCP($x, u, t, \theta$);

IV. PROPERTIES OF THE DECENTRALIZED LMPC

In this section, we show that the sampled safe sets as constructed in (10) have the property of reachability to $x_i,F$ and the approximate value function in (13) is a control Lyapunov function. Using these two properties, we proceed to show that the decentralized LMPC is persistently feasible and asymptotically stable for all sample times $t$ at all iterations $q$, and that task performance is non-decreasing over iterations.

Lemma 1. Let Assumption 2 hold, then for agent $i$ at iteration $q$, the sampled safe sets as constructed in (10) using Algorithm 1 and the dataset $(x, u)$ are reachable to $x_i,F$.

Proof: We first assume that $t, i, \mathcal{I}$ and are chosen according to Algorithm 1 such that the constructed sampled safe sets are feasible for all sample times $t$. Now, at time $t$ and $t + 1$, we obtain the time ranges $T_{i,t} = \{\max(t - t, 0), \ldots, t, \ldots, \min(t + t, t)\}$ and $T_{i,t+1} = \{\max(t - t + 1, 0), \ldots, t + 1, \ldots, t + t + 1\}$, respectively. For the case when $0 \leq t \leq t - 1$, i.e. both lower limits evaluate to zero, the sampled safe sets at time $t$ and $t + 1$ contain $x_{i,F}^p, \ldots, x_{i,F}^{t+1}$, respectively, for all $p \in \mathcal{I}$. Recall that $\mathcal{I}$ contains indices corresponding to successful iterations, i.e. $\lim_{i \to \infty} x_{i,F}^p = x_i,F, \forall p \in \mathcal{I}$. Moreover, by Assumption 2 $\mathcal{I}$ must be non-empty. By the fact that the feasible input sequence $\{u_{i,F}^p, \ldots, u_{i,F}^{t+1}\}$ exists in the dataset, we can see that the property of reachability to $x_i,F$ holds. For the case when $t \geq t$, the sampled safe sets at time $t$ and $t + 1$ contain $\{x_{i,F}^p, \ldots, x_{i,F}^{t+1}\}$ and $\{x_{i,F}^p, \ldots, x_{i,F}^{t+1}\}$. By the same argument as before, we see that reachability holds. We then conclude that the sampled safe sets for agent $i$ at iteration $q$ are reachable to $x_i,F$ for all $t \in \mathbb{N}$.

Lemma 2. Assuming $\mathcal{S}S_{i,t}^q \neq \emptyset$, then the approximate value function defined in (13) for each agent $i$ at iteration $q$ satisfies the condition

$$-V_i^q(x_{i,t}^q, t) + h_i(x_{i,t}^q, u_{i,k}) + V_i^q(f_i(x_{i,t}^q, u_{i,k}), t + 1) = 0,$$

for all $t \in \mathbb{N}$ and is therefore a control Lyapunov function.

Proof: We can write the approximate value function in (13) as $V_i^q(x_{i,t}^q, t) = J_{i,\infty}^p(x_{i,t}^q,k^*) = \sum_{k=0}^{\infty} h_i(x_{i,t}^q, u_{i,k}^*)$ where $p^*, k^* = \arg \min_{p,k} J_{i,\infty}^p(x_{i,t}^q,k)$. We have that the approximate value function is a Bellman equation and obtain the following relationship,

$$V_i^q(x_{i,t}^q, t) = h_i(x_{i,t}^q, u_{i,k}^*) + \sum_{k=0}^{\infty} h_i(x_{i,t}^q, u_{i,k}^*),$$

which concludes the proof.
Theorem 1. Consider the system in (4) controlled by the decentralized LMPC (14) and (15). Let Assumption 1 and 2 hold. Then for every agent \(i \in \mathcal{M}\):

1) The decentralized LMPC (14) is feasible for all sample times \(t \in \mathbb{N}\) and iterations \(q \in \mathbb{N}_+\).
2) The equilibrium point \(x_F\) is asymptotically stable for the global system (1) in closed-loop with (14) and (15) for all iterations \(q\).
3) The sampled safe sets \(SS^q_{i,t}\) constructed at each iteration \(q\) are non-empty for all sample times \(t\).

\[\text{Proof:}\] At iteration \(q = 1\), by Assumptions 1 and 2 the sampled safe sets \(SS^q_{i,t}\) are non-empty for all \(t \in \mathbb{N}\) and (14) is feasible at \(t = 0\).

From Lemmas 1 and 2, we have that at iteration \(q\), for each agent \(i\), the terminal constraint sets \(SS^q_{i,t}\) are reachable to \(x_i,F\) and the terminal cost function \(V^q_{i,t}(\cdot, t)\) is a control Lyapunov function for all \(t \in \mathbb{N}\). Additionally at time \(t = 0\), Problem (14) is feasible and therefore the proof for 1) and 2) follows from standard MPC arguments [18].

From 1) and 2), task execution at iteration \(q\) is successful, i.e. \(\lim_{t \to \infty} x^q_{i,t} = x_i,F\). Therefore, by Assumption 1 we may construct \(SS^q_{i,t} \supseteq \{x^q_{i,t}\} \) using Algorithm 1 (which shows 3). Repeating the argument for each iteration \(q\) concludes the proof.

Theorem 2. Consider the system (1) controlled by the decentralized LMPC (14) and (15). Let Assumption 2 hold. Then for every agent \(i \in \mathcal{M}\), the iteration cost \(J^q_{i,N}(x_i,S,0)\) does not increase with the iteration index \(q\).

\[\text{Proof:}\] We prove this theorem by establishing bounds on the LMPC cost for each agent \(i\). We note that the \(x^q_{i,t}\) and \(u^q_{i,t}\) used in the following derivations represent elements from the closed-loop state trajectory and corresponding input sequence. From (12), we have that the iteration cost can be written as

\[
J^q_{i,N}(x_i,S,0) = \sum_{t=0}^{N-1} h_i(x^q_{i,t}, u^q_{i,t}) + \sum_{t=N}^{\infty} h_i(x^q_{i,t}, u^q_{i,t}) \\
\geq \sum_{t=0}^{N-1} h_i(x^q_{i,t}, u^q_{i,t}) + V^q_{i,N}(x^q_{i,N}, N) \\
\geq \min_{u^q_{i,k}} \sum_{k=0}^{N-1} h_i(x^q_{i,k}, u^q_{i,k}) + V^q_{i,N}(x^q_{i,N}, N) \\
= J^q_{i,N}(x_i,S,0).
\]

Next, using the stability result from Theorem 1 we have that at \(t = 0\)

\[
J^q_{i,N}(x^q_{i,0},0) \geq h_i(x^q_{i,0}, u^q_{i,0}) + J^q_{i,N}(x^q_{i,1}, 1) \\
\geq h_i(x^q_{i,0}, u^q_{i,0}) + h_i(x^q_{i,1}, u^q_{i,1}) + J^q_{i,N}(x^q_{i,2}, 2) \\
\geq \lim_{t \to \infty} \sum_{k=0}^{t-1} h_i(x^q_{i,k}, u^q_{i,k}) + \lim_{t \to \infty} J^q_{i,N}(x^q_{i,t}, t).
\]

By Theorem 1, we have also established that for each agent \(\lim_{t \to \infty} x^q_{i,t} = x_i,F\) for all iterations \(q \in \mathbb{N}_+\). Therefore, we have that for agent \(i\)

\[
\lim_{t \to \infty} J^q_{i,N}(x^q_{i,t}, t) = J^q_{i,N}(x_i,F, \infty) = 0.
\]

Combining (17) and (18), we obtain

\[
J^q_{i,N}(x^q_{i,0}, 0) \geq \lim_{t \to \infty} \sum_{k=0}^{t-1} h_i(x^q_{i,k}, u^q_{i,k}) = J^q_{i,N}(x_i,S, 0).
\]

From (16) and (19), and using the fact that for agent \(i\), \(x^q_{i,0} = x_i,S\), we arrive at the final bounds

\[
J^q_{i,N}(x^q_{i,0}, 0) \geq J^q_{i,N}(x_i,S,0) \geq J^q_{i,N}(x_i,S,0),
\]

which proves the theorem.

V. MULTI-VEHICLE RECIPROCAL COLLISION AVOIDANCE

In this section, we present a numerical example of decentralized LMPC in the context of multi-vehicle collision avoidance. We compare the results from the decentralized LMPC with those from a centralized approach. The control objective is for \(M = 3\) vehicles to reach the goal equilibrium points \(\zeta^*_k\) from their respective initial states \(\zeta^*_k\) in minimum time.

A. Agent Model

We model the vehicles using the kinematic bicycle model, which is discretized using forward Euler integration with a time step of \(dt = 0.1\)s as follows

\[
\zeta_{i,t+1} = \zeta_{i,t} + dt \begin{bmatrix} v_{i,t} \cos(\psi_{i,t} + \beta_{i,t}) \\ v_{i,t} \sin(\psi_{i,t} + \beta_{i,t}) \\ \sin(\beta_{i,t})/a_{i,t} \end{bmatrix} = f_i(\zeta_{i,t}, u_{i,t}),
\]

where \(\beta_{i,t} = \arctan(l_r \tan(\delta_{i,t})/(l_f + l_r))\). The state and input variables are \(\zeta_{i,t} = [x_{i,t}, y_{i,t}, \psi_{i,t}, v_{i,t}]^T \in \mathbb{R}^4\) (with units m, m, rad, m/s) and \(u_{i,t} = [\delta_{i,t}, a_{i,t}]^T \in \mathbb{R}^2\) (with units rad, m/s²) respectively. The vehicles are coupled via collision avoidance constraints where we define a circular collision buffer about the geometric center of the vehicle with radius \(r\) and require that for all sample times,

\[
||\zeta_{i,t}(1:2) - \zeta_{j,t}(1:2)||_2 \geq 2r, \forall i, j \in \mathcal{M}, i \neq j,
\]

where \(\zeta_{i,t}(1:2)\) denotes the first two elements of \(\zeta_{i,t}\).

B. Decentralized FHOC Formulation

The decentralized FHOC for agent \(i\) at time \(t\) and iteration \(q\) given initial condition \(\zeta_i\) is formulated as in (14).

We use the stage cost \(h_i(\zeta^q_{i,k}\|z_i^q_{i,k}) = 1_{\zeta_{i,F}}(\zeta^q_{i,k}\|z_i^q_{i,k})\), where the indicator function \(1_{\zeta_{i,F}}(\zeta_{i,F})\), is 0 when \(\zeta_i = \zeta_{i,F}\) and is 1 otherwise. For the local state and input constraints in (14), in addition to the box constraints \(|\zeta^q_{i,k}\|z_i^q_{i,k}| \leq [10, 10, 10, 10]^T\), and \(|u^q_{i,k}\|z_i^q_{i,k}| \leq [0.5, 3]^T, \forall k \in \mathbb{N}_{N-1}\), we also impose constraints on the control rate via \(|u^q_{i,0} - u^q_{i,t-1}| \leq dt \cdot [0.7, 7]^T\)

1A video of the results may be found at https://youtu.be/cB9zckRm5j8
and $|u^q_{i,k+1} - u^q_{i,k}| \leq dt \cdot [0.7, 7]^T$, $\forall k \in \mathbb{N}_{N-2}$. Here, $|\cdot|$ represents the element-wise absolute value.

In [14], we decompose the global constraints into time-varying hyperplane constraints on the position states of each vehicle, i.e. $g_{i,t+k} (\zeta^q_{i,t+k}) = H^q_{i,t+k} \zeta^q_{i,t+k} + h^q_{i,t+k} \leq 0$, with $H^q_{i,t+k} \in \mathbb{R}^{M-1 \times 4}$ and $h^q_{i,t+k} \in \mathbb{R}^{M-1}$ for all $k \in \mathbb{N}_{N-1}$. This implementation of line 9 of Algorithm 1 is achieved by solving the following quadratic program (QP) for all agent pairs $(i,j) \in \mathcal{M}$, $i \neq j$ at time $t$

$$\min_{H_{ij} \in \mathbb{R}^4, h_i, h_j \in \mathbb{R}} \|H_{ij}\|_2 - (h_i + h_j)$$

s.t. $h_i, h_j \geq 0$, $H_{ij}(2:4) = 0$

$$H_{ij}^T x_i + h_i \leq 0, -H_{ij}^T x_i + h_j > 0, \forall x_i \in \mathcal{S}_{i,t}^{q-1}$$

$$H_{ij}^T x_j + h_i > 0, -H_{ij}^T x_j + h_j \leq 0, \forall x_j \in \mathcal{S}_{j,t}^{q-1}$$

which, if feasible, finds two parallel hyperplanes of maximum distance which separate the position states of the sampled safe sets for the pair of agents $(i,j)$ at time $t$.

The QP is feasible when the position states in the sampled safe sets are linearly separable. If all $(\frac{M}{2})$ pairwise QPs are feasible for all time $t$ and the distance between the hyperplanes is no less than $2r$ in all cases, then we construct the constraint decomposition for agent $i$ at time $t$ by stacking the $H_{ij}$ and $h_i$ from all QPs involving agent $i$ into the matrix $H_{ij}^{q-1}$ and vector $h_{ij}^{q-1}$ respectively. Otherwise, following Algorithm 1, we shrink the sets $\mathcal{T}_i$ and $\mathcal{L}$ and retry the QP.

We compute the initial feasible state trajectory and input sequence $(\zeta^0_i, u^0_i)$ using a linear time-varying MPC controller. The decentralized FHOCPs are solved using IPOPT [19] with the ma27 linear solver. The parameters used for this experiment are shown in Table 1.

C. Results and Discussion

As seen in Table 2, the decentralized LMPC converges to a steady state solution where the optimal cost of each iteration is non-increasing. We additionally implemented a centralized LMPC, with the same parameters and solver, for the global system subject to the original constraint (20). This approach achieved a steady state cost of 48, which is only about a 4% difference in cost with respect to the decentralized case. We also compare the computation time for solving a single FTOCP in the decentralized and centralized cases. This is summarized in Table 3. For the former, we record the maximum solve time over agents at each sampling time. For the latter, we record the solve time of the centralized FHOCP at each sampling time. We obtain that over all iterations, computation time of the decentralized case is lower by a factor of 4.6x to 24x.

In Figures 2 and 3, we compare the initial feasible trajectory to the steady state trajectory at convergence. In the initial feasible trajectory, the agents' movements are intentionally staggered in time to guarantee safety. This can be clearly seen in the velocity profile at iteration 0 in Figure 3. At convergence, all three agents begin moving simultaneously and steer to avoid collisions around the intersection point at the origin. We notice that in the steady state input sequence, the acceleration input either saturates or is close to saturating the imposed constraint and resembles a bang-bang controller [20] which switches between acceleration and deceleration at the midpoint of the trajectory.

In Figure 4, we look more closely at the steady state trajectory about the intersection point and see that the collision avoidance constraints are satisfied and are almost active for agents 1 and 3 at 2.4s and agents 2 and 3 at 2.8s. This is clearly reflected in Figure 4 which plots the minimum pairwise distance between the three agents over iterations of decentralized LMPC.

VI. CONCLUSION

In this paper, we presented a decentralized LMPC framework for dynamically decoupled multi-agent systems performing iterative tasks. In particular, we proposed a procedure for decomposing global constraints and synthesizing terminal sets and terminal cost functions for the FHOCP using data from previous iterations of task execution. We showed that the resulting decentralized LMPC has the properties of persistent feasibility, stability, and non-decreasing performance over iterations.

In the multi-vehicle collision avoidance example, due to the parallelization opportunities afforded by the decentralized implementation, we observe a significant improvement in computation time compared to a centralized approach with only a 4% increase in cost. In fact, the steady state solution from the decentralized approach saw saturation of the coupling collision avoidance constraint.

Moving forward, we would like to relax the assumption of perfect model knowledge and investigate approaches which leverage techniques in robust and stochastic optimal control to guarantee performance in the presence of model mismatch.

### Table I

| Parameter Values | $\zeta_{1,S}$ | (0.5, $-\pi/2,0$) | $\zeta_{1,F}$ | (0, $-\pi/2,0$) |
|------------------|---------------|------------------|---------------|------------------|
| $\zeta_{2,S}$   | $(-5,-5,\pi/4)$ | $\zeta_{2,F}$   | (5, $\pi/4$) |
| $\zeta_{3,S}$   | $(5,-5,3\pi/4)$ | $\zeta_{3,F}$   | $(-5,3\pi/4)$ |
| $r_{1,t}$       | 0.5m          | $r$              | 0.75m         |
| $N$             | 20            | $\mathcal{Z}$   | 20            |
| $\epsilon$      | 1e-4          | $\mathcal{Q}$   | 2             |
| $\bar{t}$       | 175           | $\mathcal{I}$   | 2             |

### Table II

#### Optimal Cost of the Decentralized LMPC at Each Iteration

| Iteration | Iteration Cost | Iteration | Iteration Cost |
|-----------|----------------|-----------|----------------|
| $q = 0$   | 296            | $q = 5$   | 51             |
| $q = 1$   | 122            | $q = 6$   | 50             |
| $q = 2$   | 79             | $q = 7$   | 50             |
| $q = 3$   | 55             | $q = 8$   | 50             |
| $q = 4$   | 51             |           |                |

### Table III

| FTOCP Solve Time | Decentralized | Centralized |
|------------------|--------------|-------------|
| Max Time [s]     | 10.2         | 48.3        |
| Min Time [s]     | 1.97         | 15.8        |
| Avg. Time [s]    | 3.35         | 20.5        |
Fig. 1. Snapshots of the decentralized LMPC trajectory at convergence. Circles represent the collision buffers centered at the position of agent 1 (blue), 2 (orange), and 3 (red).

Fig. 2. Initial (light) vs. steady state (dark) position trajectory (top). The starting and goal states are denoted as the square and circle markers respectively.

Fig. 3. Velocity profile and input sequence (bottom) for agent 1 (blue), 2 (orange), and 3 (red) at the first and last iterations. The black dashed lines correspond to the input box constraints.

Fig. 4. Minimum distance between agents at each iteration.

REFERENCES

[1] R. R. Negenborn, B. De Schutter, and J. Hellendoorn, “Multi-agent model predictive control: A survey,” arXiv preprint arXiv:0908.1076, 2009.
[2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” Automatica, vol. 36, no. 6, pp. 789–814, 2000.
[3] A. Bemporad, M. Heemels, M. Johansson, et al., Networked control systems. Springer, 2010, vol. 406.
[4] R. R. Negenborn and J. M. Maestre, “Distributed model predictive control: An overview and roadmap of future research opportunities,” IEEE Control Systems Magazine, vol. 34, no. 4, pp. 87–97, 2014.
[5] W. B. Dunbar and R. M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” Automatica, vol. 42, no. 4, pp. 549–558, 2006.
[6] L. Ferranti, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora, “Coordination of multiple vessels via distributed nonlinear model predictive control,” in 2018 European Control Conference (ECC). IEEE, 2018, pp. 2523–2528.
[7] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger, “Distributed synthesis and stability of cooperative distributed model predictive control for linear systems,” Automatica, vol. 69, pp. 117–125, 2016.
[8] C. E. Luiz and A. P. Schoellig, “Trajectory generation for multiagent point-to-point transitions via distributed model predictive control,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 375–382, 2019.
[9] J. Liu, D. M. de la Peña, and P. D. Christofides, “Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements,” Automatica, vol. 46, no. 1, pp. 52–61, 2010.
[10] A. Alessio and A. Bemporad, “Decentralized model predictive control of constrained linear systems,” in 2007 European Control Conference (ECC). IEEE, 2007, pp. 2813–2818.
[11] A. Alessio, D. Barcelli, and A. Bemporad, “Decentralized model predictive control of dynamically coupled linear systems,” Journal of Process Control, vol. 21, no. 5, pp. 705–714, 2011.
[12] T. Keviczky, F. Borrelli, and G. J. Balas, “Decentralized receding horizon control for large scale dynamically decoupled systems,” Automatica, vol. 42, no. 12, pp. 2105–2115, 2006.
[13] S. Boyd, “Sequential convex programming,” Lecture Notes, 2008.
[14] F. Augugliaro, A. P. Schoellig, and R. D’Andrea, “Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach,” in 2015 IEEE/RSJ international conference on Intelligent Robots and Systems. IEEE, 2015, pp. 5954–5961.
[15] Y. Chen, M. Cutler, and J. P. How, “Decoupled multiagent path planning via incremental sequential convex programming,” in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 5954–5961.
[16] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks: a data-driven control framework,” IEEE Transactions on Automatic Control, vol. 63, no. 7, pp. 1883–1896, 2017.
[17] ——, “Minimum time learning model predictive control,” arXiv preprint arXiv:1911.09239, 2019.
[18] F. Borrelli, A. Bemporad, and M. Morari, Predictive control for linear and hybrid systems. Cambridge University Press, 2017.
[19] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Mathematical programming, vol. 106, no. 1, pp. 25–57, 2006.
[20] D. Liberzon, Calculus of variations and optimal control theory: a concise introduction. Princeton University Press, 2011.