Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization [⋆]

William B. Dunbar [a], Richard M. Murray [b]

[a] *Department of Computer Engineering, Baskin School of Engineering*
University of California, 1156 High Street, Santa Cruz, CA, 95064, USA

[b] *Department of Control and Dynamical Systems, Division of Engineering and Applied Science*
California Institute of Technology, MC 107-81, 1200 E. California Blvd., Pasadena, CA, 91125, USA

Abstract

We consider the control of interacting subsystems whose dynamics and constraints are decoupled, but whose state vectors are coupled non-separably in a single cost function of a finite horizon optimal control problem. For a given cost structure, we generate distributed optimal control problems for each subsystem and establish that a distributed receding horizon control implementation is stabilizing to a neighborhood of the objective state. The implementation requires synchronous updates and the exchange of the most recent optimal control trajectory between coupled subsystems prior to each update. The key requirements for stability are that each subsystem not deviate too far from the previous open-loop state trajectory, and that the receding horizon updates happen sufficiently fast. The venue of multi-vehicle formation stabilization is used to demonstrate the distributed implementation.

Key words: receding horizon control; model predictive control; distributed control; multi-vehicle formations.

1 Introduction

We are interested in the control of a set of dynamically decoupled subsystems that are required to perform a cooperative task. An example of such a situation is a group of vehicles cooperatively converging to a desired formation, as explored in Olfati Saber *et al.* [14], Dunbar and Murray [7], Ren and Beard [15], and Leonard and Fiorelli [10]. One control approach that accommodates a general cooperative objective is receding horizon control. In receding horizon control, or model predictive control, the current control action is determined by solving online, at each sampling instant, a finite horizon optimal control problem. In continuous time formulations, each optimization yields an open-loop control trajectory and the initial portion of the trajectory is applied to the system until the next sampling instant. A survey of receding horizon control is given by Mayne *et al.* [11]. For the problem of interest here, cooperation between subsystems can be incorporated in the optimal control problem by including coupling terms in the cost function, as done in [7] and [14]. In this paper, subsystems that are coupled in the cost function are referred to as *neighbors*. When the subsystems are operating in a real-time distributed environment, as is typically the case with multi-vehicle systems, a centralized implementation is generally not viable, due to the computation and communication requirements of solving the centralized problem at every receding horizon update. In this paper, a *distributed implementation* of receding horizon control is presented in which each subsystem is assigned its own optimal control problem, optimizes only for its own control at each update, and exchanges information only with neighboring subsystems. It is assumed that neighboring subsystems can directly communicate with one another. The motivation for pursuing such a distributed implementation is to enable the autonomy of the individual subsystems while reducing the computation and communication requirements of a centralized implementation.

Previous work on distributed receding horizon control include Jia and Krogh [3], Motee and Sayyar-Rodsaru [13] and Acar [1]. All of these papers address unconstrained coupled LTI subsystem dynamics with quadratic separable cost functions. In another work, Jia and Krogh [8] solve a min-max problem for each

[⋆] This paper was not presented at any IFAC meeting. Corresponding author: W. B. Dunbar. Tel. +01-831-459-1031. Fax +01-831-459-4829.

Email addresses: dunbar@soe.ucsc.edu (William B. Dunbar), murray@cds.caltech.edu (Richard M. Murray).

subsystem, where again coupling comes in the dynamics and the neighboring subsystem states are treated as bounded contracting disturbances. In contrast to this work, subsystems are here coupled via the cost function, and do not view one another as bounded, contracting disturbances. Instead, vehicles communicate their most recent optimal control policy. A work related to [8] is by How and Richards [16], who examine the multi-vehicle case of linear dynamically decoupled subsystems and coupling constraints, e.g., collision avoidance constraints. By their approach, vehicles update sequentially (in order), and robust feasibility is shown assuming initial feasibility. Similar to Jia and Krogh, neighbors whose update has not occurred in the sequence are viewed as bounded, contracting disturbances. In comparison, collision avoidance constraints are only discussed here (Section 5), while more general dynamics are considered and vehicles perform control updates in parallel. Keviczky *et al.* [9] have also recently formulated a distributed model predictive scheme where each subsystem optimizes locally for itself and every neighbor at each update. The primary obstacles to ensuring feasibility and stability by this scheme are stated, and a hierarchical version is also given.

We begin in Section 2 by defining the nonlinear subsystem dynamics and an integrated cost function. Both are specific to a multi-vehicle formation objective (subsystems are henceforth referred to as *vehicles*). However, the theory applies for more general decoupled dynamics and coupling cost functions [5]. In Section 3, the integrated cost is decomposed into distributed integrated costs and a distributed optimal control problem is defined for each vehicle. The distributed receding horizon control algorithm is then defined, and the stability results are given in Section 4. Two key requirements for stability are that the receding horizon updates happen sufficiently fast, and that each distributed optimal state trajectory satisfy a *compatibility constraint*. Loosely speaking, the compatibility constraints ensure that the actual state trajectory of each vehicle is not too far from the trajectory that each neighbor assumes for that vehicle, from one receding horizon update to the next. This is in contrast to the work in [8] and [9], where neighbors are assumed to react worst-case or solely with respect to mutual interests, and so actual and assumed behavior can be considerably different. The compatibility constraints used here incur some conservatism in the closed-loop response, a fact quantified in Section 4. Finally, Section 5 discusses conclusions and extensions.

2 System Description and Objective

In this section, we define the system dynamics and pose an integrated cost function relevant for multi-vehicle formation stabilization. The states of the vehicles are coupled in the cost function, while each vehicle has decoupled dynamics subject to input constraints. We make use of the following notation. The symbol $\|\cdot\|$ denotes any vector norm in $\mathbb{R}^n$, and dimension n follows from the context. For any vector $x \in \mathbb{R}^n$, $\|x\|_P$ denotes the P-weighted 2-norm, defined by $\|x\|_P^2 = x^T P x$, and P is any positive-definite real symmetric matrix. Also, $\lambda_{\max}(P)$ and $\lambda_{\min}(P)$ denote the largest and smallest eigenvalues of P, respectively. The set $B(x;r)$ denotes a closed ball in $\mathbb{R}^n$ with center x and radius r.

Our objective is to stabilize a group of vehicles toward an equilibrium point in a cooperative way using receding horizon control. For each vehicle $i \in \{1, ..., N_a\}$, the state and control vectors are denoted $z_i(t) = (q_i(t), \dot{q}_i(t)) \in \mathbb{R}^{2n}$ and $u_i(t) \in \mathbb{R}^m$, respectively, at any time $t \geq t_0 \in \mathbb{R}$. The vectors $q_i(t) \in \mathbb{R}^n$ and $\dot{q}_i(t) \in \mathbb{R}^n$ are the position and velocity, respectively, of each vehicle i. The *decoupled* second-order, time-invariant nonlinear system dynamics for each vehicle $i \in \{1, ..., N_a\}$ are given by $\ddot{q}_i(t) = g_i(q_i(t), \dot{q}_i(t), u_i(t)), \quad t \geq t_0$, which we shall write in the equivalent form

$$\dot{z}_i(t) = f_i(z_i(t), u_i(t)), \quad t \geq t_0, \tag{1}$$

where $f_i(z_i(t), u_i(t)) = (\dot{q}_i(t), g_i(q_i(t), \dot{q}_i(t), u_i(t))) \in \mathbb{R}^{2n}$. It is assumed that there is no model error. While the system dynamics can be different for each vehicle, the dimension of every vehicles state (control) is assumed to be the same, for notational simplicity and without loss of generality. Each vehicle i is also subject to the decoupled input constraints $u_i(t) \in \mathcal{U},\ t \geq t_0$, and $\mathcal{U}^N$ is the N-times Cartesian product $\mathcal{U} \times \cdots \times \mathcal{U}$. The concatenated vectors are denoted $q = (q_1, ..., q_{N_a})$, $\dot{q} = (\dot{q}_1, ..., \dot{q}_{N_a})$, $z = (z_1, ..., z_{N_a}) \in \mathbb{R}^{2nN_a}$ and $u = (u_1, ..., u_{N_a}) \in \mathcal{U}^{N_a}$. In concatenated vector form, the system dynamics are

$$\dot{z}(t) = f(z(t), u(t)), \quad t \geq t_0, \quad \text{given } z(t_0), \tag{2}$$

where $f(z, u) = (f_1(z_1, u_1), ..., f_{N_a}(z_{N_a}, u_{N_a}))$. The desired equilibrium point is denoted $z^c = (z_1^c, ..., z_{N_a}^c)$. Since the dynamics are second-order and time-invariant, the desired equilibrium velocity $\dot{q}_i^c = 0$ for every vehicle i, and the desired constant equilibrium position values are denoted $q^c = (q_1^c, ..., q_{N_a}^c)$. We now make some standard assumptions regarding the system (2) and the set $\mathcal{U}$ (e.g., see (A1)–(A3) in [4]).

Assumption 2.1 The following holds: (**a**) $f : \mathbb{R}^{2nN_a} \times \mathbb{R}^{mN_a} \to \mathbb{R}^{2nN_a}$ is twice continuously differentiable, $0 = f(z^c, 0)$, and f linearized around $(z, u) = (z^c, 0)$ is stabilizable; (**b**) the system (2) has a unique, absolutely continuous solution for any initial condition $z(t_0)$ and any piecewise right-continuous control $u : [t_0, \infty) \to \mathcal{U}^{N_a}$; (**c**) $\mathcal{U}$ is a compact subset of $\mathbb{R}^m$ containing the origin in its interior.

Let $u_{\max}$ be the positive scalar constant $u_{\max} = \left\{ \max \|v(t)\| \;\middle|\; v(t) \in \mathcal{U}^{N_a},\ t \geq t_0 \in \mathbb{R} \right\}$. The integrated cost for multi-vehicle formation stabilization is

$$L(z,u) = \sum_{(i,j)\in\ \mathcal{E}_0} \omega \|q_i - q_j + d_{ij}\|^2 \\ + \omega\|q_\Sigma - q_d\|^2 + \nu\|\dot{q}\|^2 + \mu\|u\|^2,$$

given the positive weighting constants $\omega, \nu, \mu \in \mathbb{R}$, and where $\omega\|q_\Sigma - q_d\|^2$ is the *tracking cost*, defined by $q_\Sigma = (q_1+q_2+q_3)/3$ and $q_d = (q_1^c+q_2^c+q_3^c)/3$. The set $\mathcal{E}_0$ is the *set of all pair-wise neighbors* that defines the formation in the following way. First, if $(i,j) \in \mathcal{E}_0$, then $(j,i) \notin \mathcal{E}_0$, and $(i,i) \notin \mathcal{E}_0$ for every vehicle $i \in \{1,...,N_a\}$. Next, for every vehicle i there is at least one pair (i,j) or (j,i) in $\mathcal{E}_0$, i.e., every vehicle has at least one neighbor. Finally, associated with $\mathcal{E}_0$ is the set of constant relative vectors $\mathcal{D} = \{d_{ij} \in \mathbb{R}^n | (i,j) \in \mathcal{E}_0\}$, each of which connects the desired equilibrium positions of a pair of neighboring vehicles, i.e., for any two neighbors i and j, $q_i^c + d_{ij} = q_j^c$. Additionally, the relative vectors in $\mathcal{D}$ are consistent with one another in the sense that, e.g., if (i,j), (j,k) and (i,k) are all in $\mathcal{E}_0$, then $d_{ij} + d_{jk} = d_{ik}$. It is assumed at the outset that $\mathcal{E}_0$ and $\mathcal{D}$ are provided by some supervisory mechanism. Note that $L(z,u) = 0$ if and only if $(z,u) = (z^c, 0)$. Also, while the tracking cost is here defined with vehicles 1, 2 and 3, different and fewer (or more) vehicles can be included in this term without loss of generality. The set of pair-wise neighbors of any vehicle $i \in \{1,...,N_a\}$ is defined as $\mathcal{N}_i = \{j \in \{1,...,N_a\} \mid (i,j) \text{ or } (j,i) \in \mathcal{E}_0\}$. When we refer to the *neighbors* of any vehicle $i \in \{4,...,N_a\}$, we mean the set $\mathcal{N}_i$, and the *neighbors* of any vehicle $i \in \{1,2,3\}$ refers to the set $\mathcal{N}_i \cup \{1,2,3\} \setminus \{i\}$. The integrated cost can be equivalently written as

$$L(z,u) = \|z - z^c\|_Q^2 + \mu\|u\|^2, \qquad (3)$$

where $Q = Q^T > 0$ (Proposition 6.1 in [5]). In the next section, $L(z,u)$ is decomposed into distributed integrated cost functions. Then, distributed optimal control problems and the corresponding distributed receding horizon control algorithm are stated.

3 Distributed Receding Horizon Control

In this section, we introduce notation, define N_a separate optimal control problems and the distributed receding horizon control algorithm. For any vehicle $i \in \{1,...,N_a\}$, let $z_{-i} = (z_{j_1},...,z_{j_k})$ and $u_{-i} = (u_{j_1},...,u_{j_k})$ denote the vectors of the states and controls of the neighbors of i, respectively, where the ordering of the sub vectors is arbitrary but fixed. Also, $\dot{z}_{-i} = f_{-i}(z_{-i}, u_{-i})$ represents the collective decoupled dynamics of the neighbors of any vehicle i. The *distributed integrated cost* in the optimal control problem for any vehicle $i \in \{1,...,N_a\}$ is defined as

$$L_i(z_i, z_{-i}, u_i) = L_i^z(z_i, z_{-i}) + \gamma\mu\|u_i\|^2 + L^d(i), \quad \text{where}$$

$$L_i^z(z_i, z_{-i}) = \sum_{j\in\mathcal{N}_i} \frac{\gamma\omega}{2}\|q_i - q_j + d_{ij}\|^2 + \gamma\nu\|\dot{q}_i\|^2$$

$$\text{and} \quad L^d(i) = \begin{cases} \gamma\omega\|q_\Sigma - q_d\|^2/3, & i \in \{1,2,3\} \\ 0, & \text{otherwise,} \end{cases}$$

and $\gamma \in \mathbb{R}$ is a positive constant. The cost $L^d(i)$ is defined such that only vehicles 1, 2 and 3 have a nonzero fraction of the tracking cost, since it is only these vehicles whose states appear in the tracking cost. By construction, $\sum_{i=1}^{N_a} L_i(z_i, z_{-i}, u_i) = \gamma L(z,u)$. Note that the terms that couple the positions of vehicles are equally weighted in the decomposition, although such a choice is not necessary for the stability results to hold. In every distributed optimal control problem, the same constant prediction horizon $T \in (0,\infty)$ and constant update period $\delta \in (0,T]$ are used. In practice, the update period $\delta \in (0,T]$ is typically the sample interval. By our distributed implementation, an additional condition on δ is required, namely that it be chosen sufficiently small, as quantified in the next section. At each receding horizon update, every optimal control problem is solved synchronously, i.e., at the same instant in time. The common receding horizon update times are denoted $t_k = t_0 + \delta k$, where $k \in \mathbb{N} = \{0,1,2,...\}$. At each update, every vehicle optimizes only for its own open-loop control, given its current state and that of its neighbors. Since each cost $L_i(z_i, z_{-i}, u_i)$ depends upon the neighboring states z_{-i}, each vehicle i must presume some trajectories for z_{-i} over each prediction horizon. To that end, prior to each update, each vehicle i *receives* an *assumed* control trajectory from each neighbor. Then, using the model, the current state and the assumed control for that neighbor, the assumed state trajectories are computed. Likewise, vehicle i transmits an assumed control to all neighbors prior to each optimization. By design, the assumed control for any vehicle is *the same* in every distributed optimal control problem in which it occurs, i.e., every neighbor of i will assume the same trajectories for i over each prediction horizon. To distinguish all of the different trajectories, we introduce the following notation. Recall that $z_i(t)$ and $u_i(t)$ are the actual state and control, respectively, for each vehicle $i \in \{1,...,N_a\}$ at any time $t \geq t_0$. Over any prediction interval $[t_k, t_k + T]$, $k \in \mathbb{N}$, associated with current time t_k, for each vehicle $i \in \{1,...,N_a\}$ we denote

$u_i^p(\tau; t_k)$: the predicted control trajectory,

$u_i^*(\tau; t_k)$: the optimal predicted control trajectory,

$\hat{u}_i(\tau; t_k)$: the assumed control trajectory,

where $\tau \in [t_k, t_k + T]$. The corresponding state trajectories are likewise denoted $z_i^p(\tau; t_k)$, $z_i^*(\tau; t_k)$ and $\hat{z}_i(\tau; t_k)$, and at time $\tau = t_k$, all of these trajectories are equal to the initial condition $z_i(t_k)$. Let $u^p(\tau; t_k)$, $u^*(\tau; t_k)$ and $\hat{u}(\tau; t_k)$ be the concatenated predicted, optimal and assumed control vectors for all vehicles, respectively, with similar notation for the concatenated state vectors. Consistent with z_{-i}, also let $\hat{u}_{-i}(\tau; t_k)$ and $\hat{z}_{-i}(\tau; t_k)$ be the assumed control and state trajectories of the neighbors of i, corresponding to current time t_k. The collection of distributed optimal control problems is now defined.

Problem 3.1 For each vehicle $i \in \{1, ..., N_a\}$ and at any update time t_k, $k \in \mathbb{N}$: Given $z_i(t_k)$, $z_{-i}(t_k)$, and $\hat{u}_i(\tau; t_k)$ and $\hat{u}_{-i}(\tau; t_k)$ for all $\tau \in [t_k, t_k + T]$, find $J_i^*(z_i(t_k), z_{-i}(t_k)) = \min_{u_i^p} J_i(z_i(t_k), z_{-i}(t_k), u_i^p(\cdot; t_k))$, where $J_i(z_i(t_k), z_{-i}(t_k), u_i^p(\cdot; t_k))$ is equal to

$$\int_{t_k}^{t_k+T} L_i(z_i^p(s; t_k), \hat{z}_{-i}(s; t_k), u_i^p(s; t_k)) \, \mathrm{d}s \\ + \ \gamma \| z_i^p(t_k + T; t_k) - z_i^c \|_{P_i}^2,$$

subject to

$$\begin{aligned} &\dot{z}_i^p(\tau; t_k) = f_i(z_i^p(\tau; t_k), u_i^p(\tau; t_k)) \\ &\dot{\hat{z}}_i(\tau; t_k) = f_i(\hat{z}_i(\tau; t_k), \hat{u}_i(\tau; t_k)) \\ &\dot{\hat{z}}_{-i}(\tau; t_k) = f_{-i}(\hat{z}_{-i}(\tau; t_k), \hat{u}_{-i}(\tau; t_k)) \\ &u_i^p(\tau; t_k) \in \mathcal{U} \\ &\| z_i^p(\tau; t_k) - \hat{z}_i(\tau; t_k) \| \le \delta^2 \kappa \end{aligned} \tag{4}$$

for all $\tau \in [t_k, t_k + T]$, with $z_i^p(t_k; t_k) = \hat{z}_i(t_k; t_k) = z_i(t_k)$ and $\hat{z}_{-i}(t_k; t_k) = z_{-i}(t_k)$, and terminal constraint

$$z_i^p(t_k + T; t_k) \in \Omega_i(\varepsilon_i),$$

given the constants $\kappa, \varepsilon_i \in (0, \infty)$, weighting matrix $P_i = P_i^T > 0$, and terminal set $\Omega_i(\varepsilon_i) = \{z \in \mathbb{R}^{2n} \mid \|z - z_i^c\|_{P_i}^2 \le \varepsilon_i\}$. ■

As part of the optimal control problem, the optimized state for i is constrained to be at most a distance of $\delta^2 \kappa$ from the assumed state in Equation (4). We refer to Equation (4) as the state *compatibility constraint*. The constraint is a means of enforcing a degree of consistency between what a vehicle plans to do and what neighbors believe that vehicle will plan to do, proportional to the square of the update period. The optimal control solution to each distributed optimal control problem (assumed to exist) is $u_i^*(\tau; t_k)$, $\tau \in [t_k, t_k + T]$. The closed-loop system for which stability is to be guaranteed is

$$\dot{z}(\tau) = f(z(\tau), u_{\mathrm{RH}}(\tau)), \quad \tau \ge t_0, \tag{5}$$

with the applied *distributed receding horizon control law*

$$u_{\mathrm{RH}}(\tau) = (u_1^*(\tau; t_k), ..., u_{N_a}^*(\tau; t_k)),$$

for $\tau \in [t_k, t_{k+1})$ and any $k \in \mathbb{N}$. The receding horizon control law is updated when each new initial state update $z(t_k) \leftarrow z(t_{k+1})$ is available. Before stating the control algorithm formally, which in turn defines the assumed control for each vehicle at every update, a decoupled terminal controller associated with each terminal cost and constraint set is defined. The linearization of the ith subsystem (1) at $(z_i, u_i) = (z_i^c, 0)$ is denoted $A_i = \frac{\partial f_i}{\partial z_i}(z_i^c, 0)$, $\quad B_i = \frac{\partial f_i}{\partial u_i}(z_i^c, 0)$. By assuming stabilizability for each vehicle i (Assumption 2.1 (a)), a feasible local linear feedback $u_i = K_i(z_i - z_i^c)$ which stabilizes each linearized and nonlinear subsystem (1) in $\Omega_i(\varepsilon_i)$ can be constructed [4,12]. To that end, we make an assumption. First, for each $i \in \{1, ..., N_a\}$, let $z_i^K(t; z_i')$ denote the closed-loop solution to

$$\dot{z}_i^K(t; z_i') = f_i\left(z_i^K(t; z_i'), K_i\left(z_i^K(t; z_i') - z_i^c\right)\right), \tag{6}$$

with $t \ge t_0$, given initial condition z_i'. Also, define the asymptotically stable matrix $A_i^c = A_i + B_i K_i$, and define $Q_i = \lambda_{\max}(Q) I \in \mathbb{R}^{2n \times 2n}$ where Q is the weighting for the integrated cost (3).

Assumption 3.1 For every vehicle $i \in \{1, ..., N_a\}$, the largest positive constant $\varepsilon_i > 0$ is chosen such that: (a) the function $V_i(z_i^K) = \|z_i^K - z_i^c\|_{P_i}^2$ satisfies $\frac{d}{dt} V_i(z_i^K) \le -\|z_i^K - z_i^c\|_{Q_i + \mu K_i^T K_i}^2$ along solutions to (6) for any inital state in $\Omega_i(\varepsilon_i)$, and (b) $u_i = K_i(z_i - z_i^c) \in \mathcal{U}$ for all $z_i \in \Omega_i(\varepsilon_i)$.

Following the logic presented in Section II of [12], it is straightforward to show that such a positive constant $\varepsilon_i > 0$ exists, and immediate consequence is that $\Omega_i(\varepsilon_i)$ is a positively invariant region of attraction for Equation (6). As such, Assumption 3.1 could alternatively be replaced by an existence lemma and proof, and a design constraint on each ε_i to meet the stated conditions. By construction, $\mathrm{diag}(Q_1, ..., Q_{N_a}) = \lambda_{\max}(Q) I \ge Q$, where Q is the weighting for the integrated cost (3). Denoting $K = \mathrm{diag}(K_1, ..., K_{N_a})$ and $P = \mathrm{diag}(P_1, ..., P_{N_a})$, observe that by Assumption 3.1,

$$\frac{d}{dt} \|z^K(t) - z^c\|_P^2 \le -\|z^K(t) - z^c\|_{Q + \mu K^T K}^2, \tag{7}$$

for all $z_i^K(t) \in \Omega_i(\varepsilon_i)$ and every $i \in \{1, ..., N_a\}$, where $z^K = (z_1^K, ..., z_{N_a}^K)$. The decoupled linear feedbacks are referred to as *terminal controllers*. In the quasi-infinite horizon approach in [4], the (single) terminal controller is never actually employed, as the receding horizon control law is applied for all time. In the *dual-mode* approach in [12], receding horizon control is employed until the state reaches the terminal constraint set, at which point the terminal controller is employed for all future time. The distributed implementation algorithm defined below is based on the quasi-infinite horizon approach, while a dual-mode version is discussed in Section 4. Let $Z_\Sigma \subset \mathbb{R}^{2nN_a}$ denote the set of initial states $z(t)$ which can be steered to $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$ by a piecewise right continuous control $u^p(\cdot; t) : [t, t+T] \to \mathcal{U}^{N_a}$. To achieve convergence, the update period must satisfy $\delta \leq \delta_{\max}$, where the constant $\delta_{\max} \in (0, T]$ is defined in the next section. When results apply for any constant $\delta \in (0, T]$, we set $\delta_{\max} = T$. Following the succinct presentation in [12], we now state the control algorithm.

Algorithm 3.1 At time t_0 with $z(t_0) \in Z_\Sigma$, the *Distributed Receding Horizon Controller* for any vehicle $i \in \{1, ..., N_a\}$ is as follows:

```
*Data:* $z_i(t_0)$, $z_{-i}(t_0)$, $T \in (0, \infty)$, $\delta \in (0, \delta_{\max}]$.
*Initialization:* At time $t_0$, solve Problem 3.1 for vehicle $i$, setting $\hat{u}_i(\tau; t_0) = 0$ and $\hat{u}_{-i}(\tau; t_0) = 0$ for all $\tau \in [t_0, t_0 + T]$ and removing the constraint (4).
*Controller:*
  (1) Over any interval $[t_k, t_{k+1})$, $k \in \mathbb{N}$:
      (a) Apply $u_i^*(\tau; t_k)$, $\tau \in [t_k, t_{k+1})$.
      (b) Compute $\hat{u}_i(\tau; t_{k+1}) = \hat{u}_i(\tau)$ as
          $\hat{u}_i(\tau) = \begin{cases} u_i^*(\tau; t_k), & \tau \in [t_{k+1}, t_k + T) \\ K_i\left(z_i^K(\tau; z_i^k) - z_i^c\right), & \tau \in [t_k + T, t_{k+1} + T] \end{cases}$
          where $z_i^k := z_i^*(t_k + T; t_k)$.
      (c) Transmit $\hat{u}_i(\cdot; t_{k+1})$ to every neighbor and receive $\hat{u}_j(\cdot; t_{k+1})$ from every neighbor $j$.
  (2) At any time $t_k$, $k \in \{1, 2, ...\}$:
      (a) Measure current state $z_i(t_k)$ and measure or receive the current states $z_{-i}(t_k)$.
      (b) Solve Problem 3.1 for vehicle $i$, yielding $u_i^*(\tau; t_k)$, $\tau \in [t_k, t_k + T]$. ■
```

At initialization of Algorithm 3.1, Problem 3.1 is solved for each vehicle without enforcing the compatibility constraint (4) and assuming that every neighbor applies zero control over the prediction interval $[t_0, t_0+T]$. The choice of $\hat{u}(\tau; t_0) = 0$ at initialization is motivated in [5]. When $z(t_0) \in Z_\Sigma$, Problem 3.1 is feasible at initialization, in that the input and terminal constraints are satisfied and every distributed value function $J_i(\cdot)$ is bounded. At every subsequent update t_k, $k \geq 1$, the compatibility constraints are enforced, and each vehicle assumes all neighbors will continue along their previous open-loop plans, finishing with their decoupled linear control laws. Although Algorithm 3.1 requires the solution to Problem 3.1 instantaneously at each update time t_k, a predictive version could be stated to account for non-trivial computation times. Also, the algorithm relies on computing the optimal solution to Problem 3.1 at every update, although the optimal need not be unique. To relax this requirement, a version akin to that in [12] could be stated, wherein each distributed value function $J_i(\cdot)$ satisfies an improvement property from one update to the next. The assumed control trajectories would then be defined in terms of the previous (suboptimal) control.

4 Analysis

In this section, we state the stability results, assess the distributed implementation and discuss alternative formulations. The main result of this subsection is to show that by applying Algorithm 3.1, the closed-loop state $z(t)$ converges to a neighborhood of the objective state z^c, for a sufficiently small upper bound on the update period $\delta_{\max}$. At any time t_k, $k \in \mathbb{N}$, the sum of the optimal distributed value functions is denoted

$$J_\Sigma^*(z(t_k)) = \sum_{i=1}^{N_a} J_i^*(z_i(t_k), z_{-i}(t_k)).$$

We begin by demonstrating that initial feasibility of the implementation implies subsequent feasibility, following the standard arguments in [4] and [12]. The result requires that a modified version of Algorithm 3.1 be implemented, such that the assumed control is defined in terms of a *feasible* control rather than the optimal control. Recall that if $z(t_0) \in Z_\Sigma$, then there exists at least one (not necessarily optimal) input $u^p(\cdot; t_0) : [t_0, t_0 + T] \to \mathcal{U}^{N_a}$ such that the terminal constraints in Problem 3.1 are satisfied.

Lemma 4.1 *Suppose Assumptions 2.1 and 3.1 hold and $z(t_0) \in Z_\Sigma$. Then, for any update period $\delta \in (0, T]$, Problem 3.1 has a feasible solution at any update time t_k, $k \in \{1, 2, ...\}$.*

Proof. By assumption, Problem 3.1 has a feasible solution at time t_0, and feasibility for all subsequent update times is proven by induction. Let the feasible control and state solution at time t_k be $u^p(\cdot; t_k)$ and $z^p(\cdot; t_k)$. A candidate control that can steer $z(t_{k+1}) = z^p(t_{k+1}; t_k)$ to $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$ in time $t_{k+1} + T$ is the assumed control $\hat{u}(\cdot; t_{k+1})$, defined in component form as:

$$\hat{u}_i(\tau) = \begin{cases} u_i^p(\tau; t_k), & \tau \in [t_{k+1}, t_k + T) \\ K_i\left(z_i^K(\tau; z_i^k) - z_i^c\right), & \tau \in [t_k + T, t_{k+1} + T] \end{cases},$$

where $z_i^k = z_i^p(t_k + T; t_k)$. The candidate feasible control at update time t_{k+1} is $u^p(\cdot; t_{k+1}) = \hat{u}(\cdot; t_{k+1})$. The control and terminal constraints remain feasible from the properties of the terminal controllers stated in Assumption 3.1. Also, the compatibility constraints are trivially satisfied since $z_i^p(\cdot; t_{k+1}) = \hat{z}_i(\cdot; t_{k+1})$ for every $i \in \{1, ..., N_a\}$. □

Note that the assumed control defined above is exactly the feasible control trajectory used in Lemma 2 of [4] to show the feasibility result for a centralized implementation. The remaining analysis is based on Algorithm 3.1, and so relies on computing the optimal solution to Problem 3.1 at every update. As such, we require existence of a solution at every update based on *this* algorithm, so that our control policy is well defined.

Lemma 4.2 *Suppose Assumptions 2.1 and 3.1 hold and $z(t_0) \in Z_\Sigma$. Then, by application of Algorithm 3.1 with $\delta_{\max} = T$, Problem 3.1 has a feasible solution at any update time t_k, $k \in \{1, 2, ...\}$. Moreover, the set Z_Σ is a positively invariant set for the closed-loop system* (5).

Proof. The feasibility results follows the same logic in the proof of Lemma 4.1, with the modification that the assumed control is as defined in Algorithm 3.1. Now, suppose $z(t)$ leaves Z_Σ at some time $t = t' \in [t_k, t_{k+1})$, for some $k \in \mathbb{N}$. A feasible control that can steer $z(t')$ to $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$ in time $t' + T$ is $u^p(\cdot; t')$, defined in component form $u_i^p(\cdot; t') = u_i^p(\cdot)$ as follows:

$$u_i^p(\tau) = \begin{cases} u_i^*(\tau; t_k), & \tau \in [t', t_k + T) \\ K_i\left(z_i^K(\tau; z_i^k) - z_i^c\right), & \tau \in [t_k + T, t' + T] \end{cases},$$

where $z_i^k = z_i^*(t_k + T; t_k)$. Thus, $z(t') \in Z_\Sigma$ by contradiction, concluding the proof. □

As a consequence of Lemma 4.2, if $z(t_0) \in Z_\Sigma$, then Algorithm 3.1 can be initialized and applied for all time $t \geq t_0$. In the analysis that follows, we require that the optimal and assumed state trajectories remain bounded.

Assumption 4.1 There exists a constant $\rho_{\max} \in (0, \infty)$ such that $\|z^*(t; t_k) - z^c\| \leq \rho_{\max}$ and $\|\hat{z}(t; t_k) - z^c\| \leq \rho_{\max}$, for all $t \in [t_k, t_k + T]$ and any $k \in \mathbb{N}$.

The following lemma gives a bounding result on the decrease in $J_\Sigma^*(\cdot)$ from one update to the next. Since the compatibility constraints are enforced for update times t_k with $k \geq 1$, the result holds for $k \in \{1, 2, ...\}$.

Lemma 4.3 *Suppose Assumptions 2.1, 3.1 and 4.1 hold and $z(t_0) \in Z_\Sigma$. Then, by application of Algorithm 3.1 with $\delta_{\max} = T$, and for the positive constant ξ defined by*

$$\xi = \gamma\kappa\omega T\left(4\rho_{\max} + T^2\kappa\right)[|\mathcal{E}_0| + 1], \quad (8)$$

the function $J_\Sigma^*(\cdot)$ satisfies

$$J_\Sigma^*(z(t_k + \delta)) - J_\Sigma^*(z(t_k)) \leq \\ -\int_{t_k}^{t_k+\delta} \sum_{i=1}^{N_a} L_i^z\left(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k)\right) \mathrm{d}s + \delta^2\xi,$$

for all $k \in \{1, 2, ...\}$.

Proof. Since $z(t_0) \in Z_\Sigma$, Algorithm 3.1 can be initialized and applied for all time $t \geq t_0$. For any $k \geq 1$, $J_\Sigma^*(z(t_k))$ is equal to

$$\int_{t_k}^{t_k+T} \sum_{i=1}^{N_a} L_i(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k), u_i^*(s; t_k))\mathrm{d}s \\ + \gamma\|z^*(t_k + T; t_k) - z^c\|_P^2.$$

Applying the optimal control for some $\delta \in (0, T]$ seconds, we are now at time $t_{k+1} = t_k + \delta$, with new state update $z(t_{k+1})$. A feasible (suboptimal) control for Problem 3.1 at update time t_{k+1} is $u^p(\cdot; t_{k+1}) = \hat{u}(\cdot; t_{k+1})$; therefore,

$$J_\Sigma^*(z(t_{k+1})) \leq \int_{t_{k+1}}^{t_{k+1}+T} \gamma L(\hat{z}(s; t_{k+1}), \hat{u}(s; t_{k+1}))\,\mathrm{d}s \\ + \gamma\|\hat{z}(t_{k+1} + T; t_{k+1}) - z^c\|_P^2,$$

$$\begin{aligned} &J_\Sigma^*(z(t_{k+1})) - J_\Sigma^*(z(t_k)) \leq \\ &-\int_{t_k}^{t_{k+1}} \sum_{i=1}^{N_a} L_i(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k), u_i^*(s; t_k))\,\mathrm{d}s \\ &+\int_{t_{k+1}}^{t_k+T} \sum_{i=1}^{N_a} L_i(\hat{z}_i(s; t_{k+1}), \hat{z}_{-i}(s; t_{k+1}), \hat{u}_i(s; t_{k+1}))\,\mathrm{d}s \\ &-\int_{t_{k+1}}^{t_k+T} \sum_{i=1}^{N_a} L_i(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k), u_i^*(s; t_k))\,\mathrm{d}s \\ &+\int_{t_k+T}^{t_{k+1}+T} \gamma\|\hat{z}(s; t_{k+1}) - z^c\|_{Q+\mu K^TK}^2\,\mathrm{d}s \\ &+\gamma\Big[\|\hat{z}(t_{k+1} + T; t_{k+1}) - z^c\|_P^2 \\ &\qquad - \|z^*(t_k + T; t_k) - z^c\|_P^2\Big]. \end{aligned}$$

Denote $z' = \hat{z}(t_k + T; t_{k+1}) = z^*(t_k + T; t_k)$. Then, $\hat{z}_i(\tau; t_{k+1}) = z_i^K(\tau; z_i')$, i.e., the solution to Equation (6), for $\tau \in [t_k + T, t_{k+1} + T]$ and every $i \in \{1, ..., N_a\}$. By the properties stated in Assumption 3.1 and Equation (7), the sum of the last three terms in the equality above is nonpositive, and therefore the inequality holds after removing these terms. Additionally, we have for every i, $L_i(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k), u_i^*(s; t_k)) \geq L_i^z(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k))$, and so we have proven the lemma if we can prove that

$$\int_{t_{k+1}}^{t_k+T} \sum_{i=1}^{N_a} \Big[L_i(\hat{z}_i(s; t_{k+1}), \hat{z}_{-i}(s; t_{k+1}), \hat{u}_i(s; t_{k+1})) \\ - L_i(z_i^*(s; t_k), \hat{z}_{-i}(s; t_k), u_i^*(s; t_k))\Big]\mathrm{d}s \leq \delta^2\xi,$$

with ξ given by Equation (8). By definition, $\hat{z}_i(s; t_{k+1}) = z_i^*(s; t_k)$ and $\hat{u}_i(s; t_{k+1}) = u_i^*(s; t_k)$, for $s \in [t_{k+1}, t_k + T]$, and so the integrand above is equal to

$$\begin{aligned} &= \sum_{i=1}^{N_a} \sum_{j \in \mathcal{N}_i} \frac{\gamma\omega}{2}\Big\{\|q_i^*(s; t_k) - q_j^*(s; t_k) + d_{ij}\|^2 \\ &\qquad - \|q_i^*(s; t_k) - \hat{q}_j(s; t_k) + d_{ij}\|^2\Big\} \\ &+ \sum_{(i,j,l) \in \mathcal{E}_c} \frac{\gamma\omega}{27}\Big\{\|q_i^*(s; t_k) + q_j^*(s; t_k) + q_l^*(s; t_k) - 3q_d\|^2 \\ &\qquad - \|q_i^*(s; t_k) + \hat{q}_j(s; t_k) + \hat{q}_l(s; t_k) - 3q_d\|^2\Big\}, \end{aligned}$$

where $\mathcal{E}_c = \{(1,2,3),(3,1,2),(2,3,1)\}$. Using the triangle inequality, we have

$$\begin{aligned}
&\|q_i^*(s;t_k) - q_j^*(s;t_k) + d_{ij}\|^2 \\
&\qquad\qquad - \|q_i^*(s;t_k) - \hat{q}_j(s;t_k) + d_{ij}\|^2 \\
&\le\ 2\|q_i^*(s;t_k) - \hat{q}_j(s;t_k) + d_{ij}\| \cdot \|q_j^*(s;t_k) - \hat{q}_j(s;t_k)\| \\
&\qquad\qquad + \|q_j^*(s;t_k) - \hat{q}_j(s;t_k)\|^2 \\
&\le 2\left[\|q_i^*(s;t_k) - q_i^c\| + \|\hat{q}_j(s;t_k) - q_j^c\|\right]\delta^2\kappa + \delta^4\kappa^2 \\
&\le\ \delta^2\kappa\left[4\rho_{\max} + T^2\kappa\right],
\end{aligned}$$

where we use $d_{ij} = q_j^c - q_i^c$, the bound in Assumption 4.1, the compatibility constraint bound, and that $\delta^2 \le T^2$. Bounding the terms in the tracking cost in the same way, the integrated expression becomes

$$\begin{aligned}
\delta^2\gamma\kappa\omega \int_{t_{k+1}}^{t_k+T} &\left\{ \sum_{i=1}^{N_a} \sum_{j\in\mathcal{N}_i} \frac{1}{2}\left[4\rho_{\max} + T^2\kappa\right] \right. \\
&\left. + \sum_{(i,j,l)\in\mathcal{E}_c} \frac{1}{27}\left[12\rho_{\max} + 4T^2\kappa\right] \right\} \mathrm{d}s \le\ \delta^2\xi,
\end{aligned}$$

where ξ is an upper bound given by Equation (8), with the total number of pairwise neighbors $|\mathcal{E}_0| = \sum_{i=1}^{N_a}\sum_{j\in\mathcal{N}_i} 1/2$. This completes the proof. □

In the following, we demonstrate that by the application of Algorithm 3.1, the closed-loop state trajectory converges to a closed neighborhood of the objective state. In particular, the neighborhood of convergence is a level set of the function $J_\Sigma^*(z(t))$. First, denote the compact level sets as $\Omega_\beta = \{z \in \mathbb{R}^{2nN_a} \mid J_\Sigma^*(z) \le \beta\}$, with constant $\beta \in (0,\infty)$. The set Ω_β is in the interior of Z_Σ if $\beta > 0$ is sufficiently small. Now, for any $\beta \in (0,\infty)$ such that $\Omega_\beta \subset Z_\Sigma$, we can choose a constant $r = r(\beta) \in (0, \rho_{\max})$ with the following properties:

$$B(z^c; r) \subseteq \Omega_{\beta/2} \quad \text{and} \quad r^2 \le \frac{8\beta}{\gamma\lambda_{\min}(Q)}. \tag{9}$$

Our main result demonstrates that, for any $\beta \in (0,\infty)$, the closed-loop state trajectory converges to Ω_β, provided that the update period bound $\delta_{\max}$ in Algorithm 3.1 is proportional to r^2 as defined below, and r satisfies the properties in Equation (9). We require the following assumptions.

Assumption 4.2 The following holds: (**a**) the update period is sufficiently small that the following first-order Taylor series approximation is valid:

$$\begin{aligned}
\sum_{i=1}^{N_a} L_i^z(z_i^*(s;t_k), \hat{z}_{-i}(s;t_k)) &\approx \gamma\|z(t_k) - z^c\|_Q^2 \\
&+ 2\gamma(s - t_k)(z(t_k) - z^c)^T Q f(z(t_k), u^*(t_k;t_k)),
\end{aligned}$$

for all $s \in [t_k, t_k + \delta]$ and any $k \in \mathbb{N}$; (**b**) there exists a Lipschitz constant $\mathcal{K} \in [1,\infty)$ such that for any $z, z' \in Z_\Sigma,\ u, u' \in \mathcal{U}^{N_a}$,

$$\|f(z,u) - f(z',u')\|\ \le\ \mathcal{K}\Big(\|z - z'\|\ +\ \|u - u'\|\Big).$$

The main theorem of the paper is now stated.

Theorem 1 *Suppose Assumptions 2.1, 3.1 and 4.1-4.2 hold, $z(t_0) \in Z_\Sigma$ and for a given constant $\beta \in (0,\infty)$ with $\Omega_\beta \subset Z_\Sigma$, the constant $r = r(\beta) \in (0,\rho_{\max})$ is such that the properties in Equation (9) are satisfied. Then, by application of Algorithm 3.1 with*

$$\delta_{\max} = \frac{\gamma(r/2)^2\lambda_{\min}(Q)}{\xi\ + \gamma\mathcal{K}\rho_{\max}(\rho_{\max} + u_{\max})\lambda_{\max}(Q)}, \tag{10}$$

and ξ given by Equation (8), the closed-loop state trajectory enters $B(z^c; r)$ in finite time and remains in Ω_β for all future time.

Proof. Since $z(t_0) \in Z_\Sigma$, Algorithm 3.1 can be applied for all time $t \ge t_0$. We now reason about the closed-loop state trajectory for time $t \ge t_1$. A straightforward extension of Lemma 4.3 is

$$\begin{aligned}
&J_\Sigma^*(z(\tau)) - J_\Sigma^*(z(t_k)) \\
&\qquad\le\ \delta^2\xi - \int_{t_k}^{\tau} \sum_{i=1}^{N_a} L_i^z\left(z_i^*(s;t_k), \hat{z}_{-i}(s;t_k)\right)\mathrm{d}s,
\end{aligned}$$

for all $\tau \in (t_k, t_k + \delta]$, for any constant $\delta \in (0, \delta_{\max}]$ and any $k \in \{1, 2, ...\}$. The extension follows by the same logic in the proof of Lemma 3 in [4]. After substitution of the Taylor series expressions we have

$$\begin{aligned}
&J_\Sigma^*(z(\tau)) - J_\Sigma^*(z(t_k)) \\
&\quad\le -\gamma(\tau - t_k)\|z(t_k) - z^c\|_Q^2 + (\tau - t_k)^2\gamma C + \delta^2\xi,
\end{aligned}$$

where $C = -(z(t_k) - z^c)^T Q f(z(t_k), u^*(t_k;t_k))$ has the upper bound

$$\begin{aligned}
C &\le \|z(t_k) - z^c\|\|f(z(t_k), u^*(t_k;t_k))\|\lambda_{\max}(Q) \\
&\le \mathcal{K}\rho_{\max}(\rho_{\max} + u_{\max})\lambda_{\max}(Q).
\end{aligned}$$

Since $\tau - t_k \le \delta \le \delta_{\max}$, we have

$$\begin{aligned}
&J_\Sigma^*(z(\tau)) - J_\Sigma^*(z(t_k)) \\
&\le -\gamma(\tau - t_k)\|z(t_k) - z^c\|_Q^2 +\ \delta \cdot \delta_{\max}(\gamma C + \xi) \\
&\le -\gamma\lambda_{\min}(Q)\Big[(\tau - t_k)\|z(t_k) - z^c\|^2 - \delta(r/2)^2\Big].
\end{aligned} \tag{11}$$

Setting $\tau = t_k + \delta = t_{k+1}$, the bound above becomes

$$\begin{aligned}
&J_\Sigma^*(z(t_{k+1})) - J_\Sigma^*(z(t_k)) \\
&\qquad\le -\gamma\delta\lambda_{\min}(Q)\left[\|z(t_k) - z^c\|^2 - (r/2)^2\right].
\end{aligned}$$

From this inequality, there exists a finite integer $l \geq 1$ such that $z(t_l) \in B(z^c; r)$. If this were not the case, the inequality implies $J^*_\Sigma(z(t_k)) \to -\infty$ as $k \to \infty$. However, $J^*_\Sigma(z(t_k)) \geq 0$ for any $z(t_k) \in Z_\Sigma$, since the cost functions are all quadratic with nonnegative weighting constants and weighting matrices. Therefore, by contradiction, there exists a finite integer $l \geq 1$ such that $z(t_l) \in B(z^c; r) \subseteq \Omega_{\beta/2}$, verifying the first statement of the theorem. Now, we prove that $z(t) \in \Omega_\beta$ for all time $t \geq t_l$.

For any k, if $z(t_k) \in \Omega_{\beta/2} \setminus B(z^c; r/2)$, then $z(t) \in \Omega_\beta$ for all time $t \in [t_k, t_{k+1}]$ and $z(t_{k+1}) \in \Omega_{\beta/2}$. This is shown first by bounding Equation (11) as

$$J^*_\Sigma(z(\tau)) - J^*_\Sigma(z(t_k)) \leq \gamma \delta_{\max} \lambda_{\min}(Q)(r/2)^2$$

for all $\tau \in (t_k, t_{k+1}]$. Also, $\delta_{\max} < 1/4$, since

$$\delta_{\max} < \frac{\gamma (r/2)^2 \lambda_{\min}(Q)}{\gamma \mathcal{K} \rho_{\max}(\rho_{\max} + u_{\max}) \lambda_{\max}(Q)} < \frac{(r/2)^2}{\rho^2_{\max}}.$$

Therefore, the bound on J^*_Σ becomes

$$J^*_\Sigma(z(\tau)) \leq J^*_\Sigma(z(t_k)) + \frac{\gamma \lambda_{\min}(Q)(r/2)^2}{4} \leq \beta$$

for all $\tau \in (t_k, t_{k+1}]$, using $J^*_\Sigma(z(t_k)) \leq \beta/2$ and Equation (9). Thus, $z(t_k) \in \Omega_{\beta/2} \setminus B(z^c; r/2)$ implies $z(t) \in \Omega_\beta$ for all time $t \in [t_k, t_{k+1}]$. Moreover, $z(t_k) \in \Omega_{\beta/2} \setminus B(z^c; r/2)$ and Equation (11) imply that $J^*_\Sigma(z(t_{k+1})) < J^*_\Sigma(z(t_k))$ and so $z(t_{k+1}) \in \Omega_{\beta/2}$.

Also, for any k, if $z(t_k) \in B(z^c; r/2)$, then $z(t) \in B(z^c; r) \subseteq \Omega_{\beta/2}$ for all time $t \in [t_k, t_{k+1}]$. This follows from the bounding argument

$$\begin{aligned} \|z(t) - z^c\| &\leq \|z(t_k) - z^c\| + \|\int_{t_k}^{t} f(z(s), u_{\mathrm{RH}}(s)) \mathrm{d}s\| \\ &\leq r/2 + (t - t_k)\mathcal{K}(\rho_{\max} + u_{\max}) \\ &\leq r/2 + \delta_{\max}\mathcal{K}(\rho_{\max} + u_{\max}), \end{aligned}$$

for all time $t \in [t_k, t_{k+1}]$, and

$$\delta_{\max}\mathcal{K}(\rho_{\max} + u_{\max}) < \frac{(r/2)^2 \lambda_{\min}(Q)}{\rho_{\max} \lambda_{\max}(Q)} < \frac{r}{2}.$$

Combining the analysis above, we have shown the following: there exists a finite update time t_l such that $z(t_l) \in B(z^c; r) \subset \Omega_\beta$; at any subsequent update time t_k, $k > l$, $z(t_k) \in \Omega_{\beta/2} \subset \Omega_\beta$; finally, for any two subsequent update times t_k and t_{k+1}, with $k \geq l$, $z(t) \in \Omega_\beta$ for all time $t \in [t_k, t_{k+1}]$. This completes the proof. □

The theorem guarantees that, by application of Algorithm 3.1 with $\delta_{\max}$ given by Equation (10), the closed-loop state trajectory enters the the closed ball $B(z^c; r)$ in finite time and remains in the level set Ω_β for all future time. Moreover, the size of the set Ω_β can be made arbitrarily small provided the positive constant r satisfies the conditions in Equation (9). The price for a smaller set of convergence, i.e., by choosing r smaller, is a smaller bound on the update period $\delta_{\max}$, which in turn results in a tighter bound in the compatibility constraints (4). Still, the conditions above for convergence are only sufficient, and simulation results demonstrate that good closed-loop performance and convergence is achieved with an update period larger than required by the theory, as detailed in [5].

Remark 4.1 Observe that the update period bound $\delta_{\max}$ in Equation (10) is proportional to $1/\xi$, which in turn is proportional to $1/\kappa$. So, the compatibility constraint in Equation (4) cannot be independently relaxed by increasing κ, since this results in a smaller bound on δ. Also, γ serves as a convergence parameter in Equation (11), and choosing larger values for γ results in faster convergence of the closed-loop state trajectory to the set Ω_β. However, larger values of γ require smaller values for r from Equation (9), which in turn results in a smaller update period bound $\delta_{\max}$ from Equation (10).

As stated in the introduction, our motivation for pursuing a distributed implementation is to enable the autonomy of the individual subsystems while reducing the computation and communication requirements of a centralized implementation. Since each vehicle is computing its own control locally, the autonomy objective is satisfied. Regarding the cost of computation, the distributed implementation is computationally scalable in that each agent i optimizes only for itself, while numerically integrating $|\mathcal{N}_i|$ equations of motion for neighbors. This is a key advantage over a centralized implementation. Comparing the cost of communication is less straightforward, as the distributed implementation requires the transmission of *trajectories*, as opposed to just current state information, at each update. A qualitative analysis comparing the cost of computation and communication of the distributed and centralized implementations is given in [5].

We can also assess the closed-loop performance of the distributed implementation. More than in centralized implementations, the *closed-loop performance depends largely on the optimal open-loop trajectories computed at initialization.* One reason for this dependence is that vehicles update their controls under the assumption that neighbors will act on what was previously optimal. As such, the effect of the initial response is propagated into subsequent responses in a more direct way than in centralized implementations. This is true to the extent that a vehicles performance objective is affected by its neighbors, which is a function of the relative weighting between coupling and non coupling terms. For example, if the initial response is sluggish and coupling terms in the cost are heavily weighted, each vehicles subsequent control will likely be sluggish. Another reason for the dependence of performance on initialization is the compatibility constraints, a fact that we now quantify. Let $N_{\mathrm{RH}} \in \mathbb{N}$ be some number of receding horizon updates after time t_0 such that $N_{\mathrm{RH}} \cdot \delta \approx T$. At the optimum, the state compatibility constraint for each vehicle i is

$$\|z_i^*(t;t_k) - \hat{z}_i(t;t_k)\| \leq \delta^2\kappa, \quad t \in [t_k, t_k + T].$$

Over the subinterval of time $[t_k, t_{k-1} + T]$, we have

$$\|z_i^*(t;t_k) - z_i^*(t;z_i(t_{k-1}))\| \leq \delta^2\kappa, \quad t \in [t_k, t_{k-1} + T].$$

For $k = 1$ and at time $t = t_{N_{\mathrm{RH}}}$, where $t_{N_{\mathrm{RH}}} = t_0 + N_{\mathrm{RH}} \cdot \delta \approx t_0 + T$, we therefore have that

$$\|z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_1)) - z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_0))\| \leq \delta^2\kappa.$$

For $k = 2$ and at time $t = t_{N_{\mathrm{RH}}}$, we also have that

$$\|z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_2)) - z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_1))\| \leq \delta^2\kappa.$$

Applying this recursively up to $k = N_{\mathrm{RH}}$, each at time $t = t_{N_{\mathrm{RH}}}$, summing up both sides of the inequalities and applying the triangle inequality gives

$$\|z_i(t_{N_{\mathrm{RH}}}) - z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_0))\| \leq N_{\mathrm{RH}} \cdot \delta^2\kappa \approx T\delta\kappa,$$

where we use the fact that $z_i^*(t_{N_{\mathrm{RH}}}; z_i(t_{N_{\mathrm{RH}}})) = z_i(t_{N_{\mathrm{RH}}})$. After N_{RH} iterations, the current state deviates from the original optimal state, at the appropriate point in time, by at most $T\delta\kappa$. Thus, when the update period is small enough to satisfy the theoretical conditions for convergence, the compatibility constraints imply the closed-loop trajectory must remain relatively close to the trajectory computed at initialization; therefore, the transient response will only be as good as the initial response. If the compatibility constraints are relaxed by choosing a larger update time, one might expect a more optimal transient response at the price of poorer convergence. In fact, simulation experiments show good convergence even in the absence of the compatibility constraints.

5 Conclusions

In this paper, a distributed implementation of receding horizon control is formulated. An integrated cost function relevant for multi-vehicle formation stabilization that couples the states of a set of dynamically decoupled subsystems is first defined. One aspect of the generality of our approach is that the subsystem dynamics are nonlinear and heterogeneous. The coupling cost is decomposed and distributed optimal control problems are then defined. Each distributed problem is augmented with a *compatibility constraint*, which is a central element in the stability analysis by ensuring that actual and assumed responses of each vehicle are not too far from one another. Convergence to a neighborhood of the desired equilibrium point is proven in the absence of explicit uncertainty and for sufficiently fast receding horizon updates. We note that a sufficiently fast update period is also required in [12], which addresses robustness to model error. In contrast, we require sufficiently small δ to mitigate an *engineered* uncertainty, namely due to the discrepancy between the assumed and actual controls of every vehicle, which is a byproduct of our implementation and present even though there is no model error.

Since every optimal control problem is solved synchronously, the distributed receding horizon control law is not technically decentralized, since a globally synchronous implementation requires centralized clock keeping [2]. A locally synchronous, and consequently decentralized, version is currently being explored. In our distributed approach, no communication is required between vehicles while the distributed optimal control problems are being solved. This is an advantage over parallelization methods [2], where every distributed optimization must communicate with neighboring optimizations while iterating. Thus, our distributed implementation would generally incur a lower communication cost than an approach using receding horizon control with parallelization methods. A tradeoff is that for problems that admit parallelization, convergence to the centralized solution is guaranteed. The distributed implementation here, on the other hand, will perform differently in general than the centralized implementation.

Another important extension of our distributed implementation is the ability to handle coupling state constraints, such as collision avoidance constraints. The dual-mode receding horizon approach by Michalska and Mayne [12] addresses robustness to model uncertainty in the presence of generic state constraints by making the constraints more conservative. As described in [5], the same approach can be employed in the distributed implementation here to guarantee constraint satisfaction. However, accommodating coupling state constraints makes the initialization procedure more delicate. In particular, every vehicle must assume a control for each neighbor that is feasible with respect to all coupling constraints. Moreover, the compatibility constraints must be enforced at initialization to guarantee subsequent constraint satisfaction.

In conclusion, we note that great potential exists for stabilizing and scalable distributed receding horizon controllers that can jointly manage objectives/constraints for control over networks *and* control of the networks themselves, particularly in ad-hoc networks. In that light, in a simple one-dimensional multi-vehicle example, Yan and Bitmead recently explored the interaction between the control performance of receding horizon control and the information quality of different information flow architectures [17]. As network flow control problems involve dynamically coupled distributed subsystems, the recent extension of our implementation to the case of dynamically coupled subsystems [6] suggests that our approach may bare relevance to the problem of finding such a control approach with dual-management capability.

Acknowledgements

Partial support for this work was provided by the DARPA SEC program under grant number F33615-98-C-3613 and by AFOSR grant number F49620-01-1-0361. The authors gratefully acknowledge Prof. Jeff Shamma, Dr. Reza Olfati Saber, Prof. Nicolas Petit, and the three anonymous reviewers for their help in preparing and revising this work.

References

[1] L. Acar. Boundaries of the receding horizon control for interconnected systems. *Journal of Optimization Theory and Applications*, 84(2), 1995.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[3] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, February, 2002.

[4] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive scheme with guaranteed stability. *Automatica*, 14(10):1205–1217, 1998.

[5] W. B. Dunbar. *Distributed Receding Horizon Control of Multiagent Systems*. PhD thesis, California Institute of Technology, Pasadena, CA, 91125, April 2004. Available online at `http://etd.caltech.edu/`.

[6] W. B. Dunbar. A distributed receding horizon control algorithm for dynamically coupled nonlinear systems. In *Accepted to IEEE Conference on Decision and Control / IEE European Control Conference*, 2005.

[7] W. B. Dunbar and R. M. Murray. Model predictive control of coordinated multi-vehicle formations. In *Proceedings of the Conference on Decision and Control*, Las Vegas, NV, 2002.

[8] D. Jia and B. H. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the American Control Conference*, 2002.

[9] T. Keviczky, F. Borrelli, and G. J. Balas. A study on decentralized receding horizon control for decoupled systems. In *Proceedings of the American Control Conference*, Boston, MA, 2004.

[10] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the Conference on Decision and Control*, Florida, 2001.

[11] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Contrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[12] H. Michalska and D.Q. Mayne. Robust receeding horizon control of contrained nonlinear systems. *IEEE Trans. Automat. Control*, 38:1623–1632, 1993.

[13] N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, 2003.

[14] R. Olfati-Saber, W. B. Dunbar, and R. M. Murray. Cooperative control of multi-vehicle systems using cost graphs and optimization. In *Proceedings of the American Control Conference*, Denver, CO, 2003.

[15] W. Ren and R.W. Beard. A decentralized scheme for spacecraft formation flying via the virtual structure approach. *AIAA Journal of Guidance, Control and Dynamics*, 27(1):73–82, 2004.

[16] A. Richards and J. How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the American Control Conference*, Boston, MA, 2004.

[17] J. Yan and R. R. Bitmead. Coordinated control and information architectures. In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, 2003.