UCSC Baskin School of Engineering Technical Report ucsc-crl-06-03

Submitted to the IEEE Transactions on Automatic Control, Number AC06-043

Original: June 15, 2005. Revision: February 6, 2006.

Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems

William B. Dunbar*

Autonomous Systems Lab

Computer Engineering Department

University of California, Santa Cruz, 95064

Abstract

This paper considers the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints. Examples of such systems include certain large scale process control systems, chains of coupled oscillators and supply chain management systems. Receding horizon control is a method of choice in these venues as constraints can be explicitly accommodated. In addition, a distributed control approach is sought to enable the autonomy of the individual subsystems and reduce the computational burden of centralized implementations. In this paper, a distributed receding horizon control algorithm is presented for dynamically coupled nonlinear systems that are subject to decoupled input constraints. By this algorithm, each subsystem computes its own control locally. Provided an initially feasible solution can be found, subsequent feasibility of the algorithm is guaranteed at every update, and asymptotic stabilization is established. The theory is demonstrated in simulation on a set of coupled oscillators that model a walking robot experiment.

*Assistant Professor, Email: dunbar@soe.ucsc.edu, Ph: 831-459-1031.

1 Introduction

The problem of interest is to design a distributed controller for a set of dynamically coupled nonlinear subsystems that are required to perform stabilization in a cooperative way. Examples of such situations where distributed control is desirable include certain large scale process control systems [20] and supply chain management systems [2]. The control approach advocated here is receding horizon control (RHC). In RHC, the current control action is determined by solving a finite horizon optimal control problem online at every update. 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 update. A survey of RHC, also known as model predictive control, is given by Mayne *et al.* [12]. Advantages of RHC are that a large class of performance objectives, dynamic models and constraints can in principle be accommodated.

The work presented here is a continuation of a recent work [7], wherein a distributed implementation of RHC is presented in which neighbors are coupled solely through cost functions. The coupled cost problem formulation is relevant particularly for certain multiple autonomous vehicle missions. While communication network issues (such as limited bandwidth and delay) are paramount in multi-vehicle scenarios, such issues are not addressed here. The reason is that these issues are not dominant factors in the applications of interest, such as supply chain systems and the example of coupled oscillators considered at the end of the paper. In this paper, subsystems that are dynamically coupled are referred to as *neighbors*. As in [7], each subsystem is assigned its own optimal control problem, optimizes only for its own control at each update, and exchanges information with neighboring subsystems. The primary motivations for pursuing such a distributed implementation are to enable the autonomy of the individual subsystems and reduce the computational burden of centralized implementations. The requirement of distributed control in the presence of constraints is particularly true in the case of supply chain problems [4], since stages within a chain would never agree to centralized decision making.

Previous work on distributed RHC of dynamically coupled systems include Jia and Krogh [9], Motee and Sayyar-Rodsaru [15] and Acar [1]. All of these papers address coupled liner time-invariant subsystem dynamics with quadratic separable cost functions. State and input constraints are not included, aside from a stability constraint in [9] that permits state information exchanged between the subsystems to be delayed by one update period. In another work, Jia and Krogh [10] solve a min-max problem for each subsystem, where again coupling comes in the dynamics and the neighboring subsystem states are treated as

bounded disturbances. Stability is obtained by contracting each subsystems state at every sample period, until the objective set is reached. As such, stability does not depend on information updates between neighbors, although such updates may improve performance. More recently, Venkat *et al.* [18, 19] have designed a distributed model predictive control (MPC) algorithm for coupled LTI subsystems and compared it to centralized and decentralized alternatives. In their formulation, subsystems are coupled solely through the control inputs. Consequently, feasibility and stability analysis is leveraged by the diagonally decoupled and linear form of the state dynamics, for which the state solution can be carried out analytically given the set of all control trajectories.

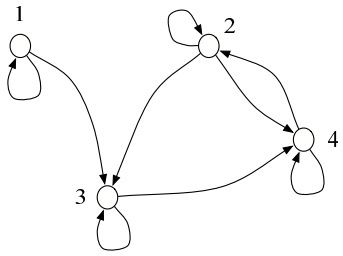
Section 2 begins by defining the nonlinear coupled subsystem dynamics and control objective. In Section 3, distributed optimal control problems are defined for each subsystem, and the distributed RHC algorithm is stated. Feasibility and stability results are then given in Section 4. Key requirements are that the receding horizon updates happen at a sufficient rate, the amount of dynamic coupling remain below a quantitative threshold, and each distributed optimal state trajectory satisfy a *consistency constraint*. The consistency constraint ensures that the computed state trajectory of each subsystem is not too far from the trajectory that each neighbor assumes for that subsystem, at each receding horizon update. In Section 5, the theory is applied to the problem of regulating a set of coupled Van der Pol oscillators that capture the thigh and knee dynamics of a walking robot experiment [8]. Fnally, Section 6 provides conclusions.

2 System Description and Objective

In this section, the system dynamics and control objective are defined. 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. Often, the notation $\|x\|$ is understood to mean $\|x(t)\|$ at some instant of time $t \in \mathbb{R}$.

Our objective is to stabilize a group of $N_a \geq 2$ dynamically coupled agents toward the origin in a cooperative and distributed way using RHC. For each agent $i \in \{1, ..., N_a\}$, the state and control vectors are denoted $z_i(t) \in \mathbb{R}^n$ and $u_i(t) \in \mathbb{R}^m$, respectively, at any time $t \geq t_0 \in \mathbb{R}$. The dimension of every agents state (control) are assumed to be the same, for notational simplicity and without loss of generality. The concatenated vectors are denoted

$$\begin{aligned}\dot{z}_1 &= f_1(z_1, z_3)\\ \dot{z}_2 &= f_2(z_2, z_3, z_4)\\ \dot{z}_3 &= f_3(z_3, z_4)\\ \dot{z}_4 &= f_4(z_4, z_2)\end{aligned} \qquad \Longrightarrow$$



(a) (b)

Figure 1: Example of (a) a set of coupled dynamic equations and (b) the corresponding directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated with the directed information flow. In this example, $\mathcal{V} = \{1, 2, 3, 4\}$ and $\mathcal{E} = \{(1,1), (1,3), (2,2), (2,3), (2,4), (3,3), (3,4), (4,4), (4,2)\}$. The upstream neighbor sets are $\mathcal{N}_1^{\mathrm{u}} = \{1, 3\}$, $\mathcal{N}_2^{\mathrm{u}} = \{2, 3, 4\}$, $\mathcal{N}_3^{\mathrm{u}} = \{3, 4\}$ and $\mathcal{N}_4^{\mathrm{u}} = \{2, 4\}$, and the downstream neighbor sets are $\mathcal{N}_1^{\mathrm{d}} = \{1\}$, $\mathcal{N}_2^{\mathrm{d}} = \{2, 4\}$, $\mathcal{N}_3^{\mathrm{d}} = \{1, 2, 3\}$ and $\mathcal{N}_4^{\mathrm{d}} = \{2, 3, 4\}$. By this convention, arrows in the graph point upstream.

$z = (z_1, ..., z_{N_a})$ and $u = (u_1, ..., u_{N_a})$.

The dynamic coupling between the agents is topologically identified by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., N_a\}$ is the set of nodes (agents) and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of all directed edges between nodes in the graph. The set $\mathcal{E}$ is defined in the following way. If any components of z_j appear in the dynamic equation for agent i, for some $j \in \mathcal{V}$, then j is referred to as an *upstream neighbor* of agent i, and $\mathcal{N}_i^{\mathrm{u}} \subseteq \mathcal{V}$ denotes the set of upstream neighbors of any agent $i \in \mathcal{V}$. The set of all directed edges is defined as $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid j \in \mathcal{N}_i^{\mathrm{u}}, \forall i \in \mathcal{V}\}$. For every $i \in \mathcal{V}$, it is assumed that z_i appears in the dynamic equation for i, and so $i \in \mathcal{N}_i^{\mathrm{u}}$ for every $i \in \mathcal{V}$. In the language of graph theory, then, every node has a self-loop edge in $\mathcal{E}$. Note that $j \in \mathcal{N}_i^{\mathrm{u}}$ does not necessarily imply $i \in \mathcal{N}_j^{\mathrm{u}}$.

It will also be useful to reference the set of agents for which any of the components of z_i arises in their dynamical equation. This set is referred to as the *downstream neighbors* of agent i, and is denoted $\mathcal{N}_i^{\mathrm{d}}$. The set of all directed edges can be equivalently defined as $\mathcal{E} = \{(j, i) \in \mathcal{V} \times \mathcal{V} \mid j \in \mathcal{N}_i^{\mathrm{d}}, \forall i \in \mathcal{V}\}$. Note that $j \in \mathcal{N}_i^{\mathrm{u}}$ if and only if $i \in \mathcal{N}_j^{\mathrm{d}}$, for any $i, j \in \mathcal{V}$. Consider the example system and corresponding directed graph given in Figure 1.

It is assumed in this paper that the graph $\mathcal{G}$ is connected. Consequently, for every $i \in \mathcal{V}$, the set $\left(\mathcal{N}_i^{\mathrm{d}} \cup \mathcal{N}_i^{\mathrm{u}}\right) \setminus \{i\}$ is nonempty, and every agent is dynamically coupled to at least one other agent. It is also assumed that agents can receive information directly from each and every upstream neighbor, and agents can transmit information directly to each and every downstream neighbor. The *coupled* time-invariant nonlinear dynamics for each agent $i \in \mathcal{V}$

is given by

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

where $z_{-i} = (..., z_{j_l}, ...)$, with $j_l \in \mathcal{N}_i^{\mathrm{u}}$ and $l = 1, ..., |\mathcal{N}_i^{\mathrm{u}}|$, denotes the concatenated vector of the states of the upstream neighbors of i, and the ordering of the sub vectors is fixed. Each agent i is also subject to the decoupled input constraints $u_i(t) \in \mathcal{U},\ t \geq t_0$. The set $\mathcal{U}^N$ is the N-times Cartesian product $\mathcal{U} \times \cdots \times \mathcal{U}$. In concatenated vector form, the system dynamics are

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

given initial condition $z(t_0)$, where $f(z, u) = (f_1(z_1, z_{-1}, u_1), ..., f_{N_a}(z_{N_a}, z_{-N_a}, u_{N_a}))$.

Assumption 1. The following holds: (**a**) the function $f : \mathbb{R}^{nN_a} \times \mathbb{R}^{mN_a} \to \mathbb{R}^{nN_a}$ is twice continuously differentiable, and satisfies $0 = f(0, 0)$; (**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**) the set $\mathcal{U}$ is a compact subset of $\mathbb{R}^m$ containing the origin in its interior.

Consider now the linearization of (1) around the origin, denoted as

$$\dot{z}_i(t) = A_{ii} z_i(t) + \sum_{j \in \mathcal{N}_i^{\mathrm{u}}} A_{ij} z_j(t) + B_i u_i(t),$$

where $A_{ij} = \partial f_i / \partial z_j(0, 0)$ and $B_i = \partial f_i / \partial u_i(0, 0)$. As in many RHC formulations [3,12,13], a feedback controller that stabilizes the closed-loop system inside a neighborhood of the origin will be utilized. To design a linear controller based on the linearization while respecting the decentralized information constraints, one can define the output variables for each agent i as $y_i(t) = C_i z(t) = (z_i(t), z_{-i}(t))$. There exist methods for constructing dynamic and static feedback controllers, as done by Corfmat and Morse in [5], to achieve stabilization while respecting the decentralized information constraints. The analysis here is greatly facilitated if, for every $i \in \mathcal{V}$, stabilization is possible with the decoupled static feedback $u_i = K_i z_i$, instead of a feedback $u_i = K_i y_i$ that relies on components of z_{-i}. To that end, the following assumption is made.

Assumption 2. For every agent $i \in \mathcal{V}$, there exists a decoupled static feedback $u_i = K_i z_i$ such that $A_{di} \triangleq A_{ii} + B_i K_i$ is Hurwitz, and the closed-loop linear system $\dot{z} = A_{\mathrm{c}} z$ is

asymptotically stable, where $A_{\rm c} \triangleq [f_z(0,0) + f_u(0,0)K]$ and $K = \mathrm{diag}(K_1, ..., K_{N_a})$.

The decoupled linear feedbacks above are referred to as *terminal controllers*. Associated with the closed-loop linearization, denote the block-diagonal Hurwitz matrix $A_{\rm d} = \mathrm{diag}(A_{d1}, ..., A_{dN_a})$ and the off-diagonal matrix $A_{\rm o} = A_{\rm c} - A_{\rm d}$. Assumption 2 inherently presumes decoupled stabilizability, and that the coupling between subsystems in the linearization is sufficiently weak as quantified in the survey paper [17].

3 Distributed Receding Horizon Control

In this section, N_a separate optimal control problems and the distributed RHC algorithm are defined. 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 the distributed implementation presented here, an additional condition on δ is required, namely that it be chosen sufficiently small, as quantified in the next section. Denote the receding horizon update times as $t_k = t_0 + \delta k$, where $k \in \mathbb{N} = \{0, 1, 2, ...\}$. In the following implementation, every distributed RHC law is updated *globally synchronously*, i.e., at the same instant of time t_k for the $k^{\rm th}$-update.

At each update, every agent optimizes only for its own predicted open-loop control, given its current state. Since the dynamics of each agent i depend upon upstream neighboring states z_{-i}, that agent will presume some trajectories for z_{-i} over each prediction horizon. To that end, prior to each update, each agent i *receives* an *assumed* state trajectory $\hat{z}_j$ from each upstream neighbor $j \in \mathcal{N}_i^{\rm u}$. Likewise, agent i transmits an assumed state trajectory $\hat{z}_i$ to every downstream neighbor $j \in \mathcal{N}_i^{\rm d}$, prior to each update. By design, then, the assumed state trajectory for any agent is *the same* in the distributed optimal control problem of every downstream neighbor.

Since the models are used with assumed trajectories for upstream neighbors, there will be a discrepancy, over each optimization time window, between the *predicted* open-loop trajectory and the *actual* trajectory that results from every agent applying their locally predicted control. This discrepancy is quantified by using the following notation. Recall that $z_i(t)$ is the actual state for each agent $i \in \mathcal{V}$ at any time $t \geq t_0$. Associated with update

time t_k, for any $k \in \mathbb{N}$, the trajectories for each agent $i \in \mathcal{V}$ are denoted

$$\begin{aligned} z_i^p(t;t_k) &- \text{the predicted state trajectory,} \\ \hat{z}_i(t;t_k) &- \text{the assumed state trajectory,} \\ u_i^p(t;t_k) &- \text{the predicted control trajectory,} \end{aligned}$$

where $t \in [t_k, t_k + T]$. Consistent with the ordering of z_{-i}, let $\hat{z}_{-i}(t;t_k)$ be the assumed open-loop state trajectories of the upstream neighbors of i, corresponding to update time t_k. The predicted state trajectory satisfies

$$\dot{z}_i^p(t;t_k) = f_i(z_i^p(t;t_k), \hat{z}_{-i}(t;t_k), u_i^p(t;t_k)), \quad t \in [t_k, t_k + T], \tag{3}$$

given $z_i^p(t_k;t_k) = z_i(t_k)$. The assumed state trajectory for each agent $i \in \mathcal{V}$ is given by

$$\hat{z}_i(t;t_k) = \begin{cases} z_i^p(t;t_{k-1}), & t \in [t_k, t_{k-1} + T) \\ z_i^K(t), & t \in [t_{k-1} + T, t_k + T] \end{cases} \tag{4}$$

where z_i^K is the solution to $\dot{z}_i^K(t) = A_{di} z_i^K(t)$ with initial condition $z_i^K(t_{k-1} + T) = z_i^p(t_{k-1} + T; t_{k-1})$. By construction, each assumed state trajectory $\hat{z}_i$ is the remainder of the previously predicted trajectory, concatenated with the closed-loop linearization response that ignores coupling. The collective *actual* state trajectories for the agents over any update window $[t_k, t_{k+1})$ is given by

$$\dot{z}(t) = f(z(t), u^p(t;t_k)), \quad t \in [t_k, t_{k+1}), \tag{5}$$

given $z(t_k)$. While the actual and predicted state trajectories do have the same initial condition $z_i(t_k)$ for each $i \in \mathcal{V}$, they typically diverge over each update window $[t_k, t_{k+1}]$, and $z^p(t_{k+1};t_k) \neq z(t_{k+1})$ in general. The reason is that, while the predicted state trajectories in (3) are based on the assumption that neighbors continue along their previous trajectory, neighbors in fact compute and employ their own updated predicted control trajectory. Therefore, the actual state evolves according to (5). The challenge then is to generate a distributed RHC algorithm that has feasibility and stability properties in the presence of the discrepancy between predicted and actual state trajectories.

A desirable property of any RHC algorithm is to have feasible state and control trajectories at any update, as the trajectories can be used to preempt the optimization algorithm used to solve the optimal control problem. In many formulations, the feasible state tra-

jectory is the remainder of the previous trajectory concatenated with the response under a terminal controller [3, 12, 13]. While $\hat{z}_i(\cdot; t_k)$ is such a trajectory, it cannot be used since $\hat{z}_i(t_k; t_k) \neq z_i(t_k)$. Still, a feasible control trajectory exists. Indeed, *a primary contribution of this paper* is to show that a feasible control is the remainder of the previous control trajectory concatenated with the terminal controller, with the corresponding feasible state trajectory starting from the true state at each update time. The feasible state and control trajectories at any update t_k are denoted $\bar{z}_i(\cdot; t_k)$ and $\bar{u}_i(\cdot; t_k)$, respectively. The feasible state trajectory satisfies

$$\dot{\bar{z}}_i(t; t_k) = f_i\left(\bar{z}_i(t; t_k), \hat{z}_{-i}(t; t_k), \bar{u}_i(t; t_k)\right), \quad t \in [t_k, t_k + T], \tag{6}$$

given initial condition $\bar{z}_i(t_k; t_k) = z_i(t_k)$, and the feasible control is given by

$$\bar{u}_i(t; t_k) = \begin{cases} u_i^p(t; t_{k-1}), & t \in [t_k, t_{k-1} + T) \\ K_i \bar{z}_i(t; t_k), & t \in [t_{k-1} + T, t_k + T] \end{cases}. \tag{7}$$

The feasible control trajectory $\bar{u}_i$ is the remainder of the previously predicted control trajectory, concatenated with the linear control applied to the nonlinear model and based on the decoupled linear responses for each upstream neighbor. In the next section, feasibility and stability will be proven. Note that stability is to be guaranteed for *the closed-loop system*, represented by equation (5), which is defined for all time $t \geq t_0$. In the remainder of this section, each local optimal control problem and the distributed RHC algorithm are defined.

In each local optimal control problem, a cost function will be utilized. For any agent $i \in \mathcal{V}$ at update time t_k, the cost function $J_i(z_i(t_k), u_i^p(\cdot; t_k))$ is given by

$$J_i(z_i(t_k), u_i^p(\cdot; t_k)) = \int_{t_k}^{t_k+T} \|z_i^p(s; t_k)\|_{Q_i}^2 + \|u_i^p(s; t_k)\|_{R_i}^2 \mathrm{d}s \; + \; \|z_i^p(t_k + T; t_k)\|_{P_i}^2,$$

where $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$ and $P_i = P_i^T > 0$. The matrix $P_i = P_i^T > 0$ is chosen to satisfy the Lyapunov equation

$$P_i A_{di} + A_{di}^T P_i = \widehat{Q}_i, \quad \forall i \in \mathcal{V}, \tag{8}$$

where $\widehat{Q}_i = Q_i + K_i^T R_i K_i$. Denoting $P = \mathrm{diag}(P_1, ..., P_{N_a})$ and $\widehat{Q} = \mathrm{diag}(\widehat{Q}_1, ..., \widehat{Q}_{N_a})$, it follows that $PA_{\mathrm{d}} + A_{\mathrm{d}}^T P = -\widehat{Q}$ and $\widehat{Q} > 0$.

Decoupled terminal state constraints will be included in each local optimal control prob-

lem. A lemma used to define the terminal state constraint sets and to guarantee that the terminal controllers are stabilizing inside the sets is now presented. The proof of the lemma utilizes an assumption that limits the amount of coupling between neighboring subsystems in the linearization.

Assumption 3. $P A_{\mathrm{o}} + A_{\mathrm{o}}^T P \leq \widehat{Q}/2$.

Lemma 1. Suppose that Assumptions 1–3 hold. There exists a positive constant $\varepsilon \in (0, \infty)$ such that the set

$$\Omega_\varepsilon \triangleq \left\{ z \in \mathbb{R}^{nN_a} \mid \|z\|_P \leq \varepsilon \right\},$$

is a positively invariant region of attraction for both the closed-loop linearization $\dot{z}(t) = A_{\mathrm{c}} z(t)$ and the closed-loop nonlinear system $\dot{z}(t) = f(z(t), Kz(t))$. Additionally, $Kz \in \mathcal{U}^{N_a}$ for all $z \in \Omega_\varepsilon$.

Proof. This proof follows closely along the lines of the logic given in Section II of [13], but is provided here as some of the steps will be reused in later proofs. Consider the function $V(z) = \|z\|_P^2$. Computing the time derivative of $V(z)$ along a solution of $\dot{z}(t) = A_{\mathrm{c}} z(t)$ yields

$$\begin{aligned}
\dot{V}(z) &= z^T \left(A_{\mathrm{c}}^T P + P A_{\mathrm{c}} \right) z = z^T \left(A_{\mathrm{d}}^T P + P A_{\mathrm{d}} \right) z + z^T \left(A_{\mathrm{o}}^T P + P A_{\mathrm{o}} \right) z \\
&\leq -z^T \widehat{Q} z + z^T \widehat{Q} z (1/2) = -(1/2) z^T \widehat{Q} z \leq -(1/2) \lambda_{\min}(P^{-1/2} \widehat{Q} P^{-1/2}) V(z),
\end{aligned}$$

which holds for all $z(t) \in \mathbb{R}^{nN_a}$. Now, let $\phi(z) \triangleq f(z, Kz) - A_{\mathrm{c}} z$, which satisfies $\phi(0) = 0$ and $\|\phi(z)\|_P / \|z\|_P \to 0$ as $\|z\|_P \to 0$. Computing the time derivative of $V(z)$ along a solution of $\dot{z}(t) = f(z(t), Kz(t))$ yields

$$\begin{aligned}
\dot{V}(z) &= z^T \left(A_{\mathrm{c}}^T P + P A_{\mathrm{c}} \right) z + 2 z^T P \phi(z) \\
&\leq -\frac{1}{2} z^T \widehat{Q} z + 2 \|z\|_P^2 \frac{\|\phi(z)\|_P}{\|z\|_P} \leq -z^T \widehat{Q} z \left[\frac{1}{2} - 2 \lambda_{\max}(\widehat{Q}^{-1/2} P \widehat{Q}^{-1/2}) \frac{\|\phi(z)\|_P}{\|z\|_P} \right].
\end{aligned}$$

Since $\|\phi(z)\|_P / \|z\|_P \to 0$ as $\|z\|_P \to 0$, there exists a constant $\varepsilon_0 \in (0, \infty)$ such that $\|\phi(z)\|_P / \|z\|_P \leq 1/(10 \lambda_{\max}(\widehat{Q}^{-1/2} P \widehat{Q}^{-1/2}))$ whenever $V(z) \leq \varepsilon_0^2$. Therefore,

$$V(z) \leq \varepsilon_0^2 \quad \Longrightarrow \quad \dot{V}(z) \leq -(3/10) z^T \widehat{Q} z \leq -(3/10) \lambda_{\min}(P^{-1/2} \widehat{Q} P^{-1/2}) V(z).$$

Let $\varepsilon \in (0, \varepsilon_0)$ be such that $Kz \in \mathcal{U}^{N_a}$ for all $z \in \Omega_\varepsilon$. Then, any state trajectory of the closed-loop linearization or nonlinear system starting in Ω_ε remains in Ω_ε and converges to the origin, and the control constraints are satisfied everywhere on such trajectories, concluding

the proof. ■

The parameter $\varepsilon \in (0, \infty)$ that satisfied the conditions of the lemma can be found numerically by solving a semi-infinite feasibility problem. Specifically, define the real-valued functions

$$\begin{aligned}\psi_1(z) &\triangleq 2z^T P\phi(z) - (1/2)z^T\widehat{Q}z + (3/10)\lambda_{\min}(P^{-1/2}\widehat{Q}P^{-1/2})z^T Pz,\\ \psi_2(z) &\triangleq d(Kz, \mathcal{U}^{N_a}),\end{aligned}$$

where $d(u, \mathcal{U}^{N_a})$ is the distance from a point u to the set $\mathcal{U}^{N_a}$. The problem is to determine the largest ε such that $\psi_1(z) \leq 0$ and $\psi_2(z) \leq 0$ for all $z \in \Omega_\varepsilon$, and approaches for doing this are discussed in [13].

In each local optimal control problem, the terminal state constraint set for each $i \in \mathcal{V}$ is

$$\Omega_i(\varepsilon) \triangleq \left\{ z_i \in \mathbb{R}^n \mid \|z_i\|_{P_i} \leq \varepsilon/\sqrt{N_a} \right\}. \tag{9}$$

By construction, if $z \in \Omega_1(\varepsilon) \times \cdots \times \Omega_{N_a}(\varepsilon)$, then the decoupled controllers can stabilize the system to the origin, since

$$\|z_i\|_{P_i}^2 \leq \frac{\varepsilon^2}{N_a},\ \forall i \in \mathcal{V} \quad \Longrightarrow \quad \sum_{i\in\mathcal{V}} \|z_i\|_{P_i}^2 \leq \varepsilon^2 \quad \Longleftrightarrow \quad z \in \Omega_\varepsilon.$$

Suppose then that at some time $t' \geq t_0$, $z_i(t') \in \Omega_i(\varepsilon)$ for every $i \in \mathcal{V}$. Then, from Lemma 1, stabilization is achieved if every agent employs their decoupled static feedback controller $K_i z_i(t)$ for all time $t \geq t'$. Thus, the objective of the RHC law is to drive each agent i to the set $\Omega_i(\varepsilon)$. Once all agents have reached these sets, they switch to their decoupled controllers for stabilization. Switching from RHC to a terminal controller once the state reaches a suitable neighborhood of the origin is referred to as *dual-mode* RHC [13]. For this reason, the implementation here is considered a dual-mode distributed RHC algorithm. The collection of local optimal control problems is now defined.

Problem 1. Let $\varepsilon \in (0, \infty)$ satisfy the conditions in Lemma 1, and let $q \in \{1, 2, 3, ...\}$ be any positive integer. For each agent $i \in \mathcal{V}$ and at any update time t_k, $k \geq 1$:

Given: $z_i(t_k)$, $\overline{z}_{-i}(t; t_k)$ and $\hat{z}_{-i}(t; t_k)$ for all $t \in [t_k, t_k + T]$;

Find: the control trajectory $u_i^p(\cdot; t_k) : [t_k, t_k + T] \to \mathcal{U}$ that minimizes $J_i(z_i(t_k), u_i^p(\cdot; t_k))$,

subject to the constraints

$$\|z_i^p(t;t_k)\|_{P_i}^2 \leq \|\bar{z}_i(t;t_k)\|_{P_i}^2 + \left[\frac{\delta\varepsilon}{8T\sqrt{N_a}}\right]^2, \ \forall t \in [t_k, t_k+T], \tag{10}$$

$$\|z_i^p(t;t_k) - \hat{z}_i(t;t_k)\|_{P_i}^2 \leq \|\bar{z}_i(t;t_k) - \hat{z}_i(t;t_k)\|_{P_i}^2 + \left[\frac{\varepsilon}{2(q+1)N_a}\right]^2, \ \forall t \in [t_k, t_k+T], \tag{11}$$

where $z_i^p(\cdot;t_k)$ satisfies the dynamic equation (3) and the terminal constraint $z_i^p(t_k+T;t_k) \in \Omega_i(\varepsilon/2)$, with Ω_i defined in (9). ■

Equation (10) is utilized to prove that the distributed RHC algorithm is stabilizing. While many centralized RHC algorithms rely on the (typically local) optimality of the solution $u_i^p(\cdot;t_k)$ at each update t_k [12], the stability results in the next section do not. Instead, the constraint (10) is shown to be sufficient for stability purposes. As such, the minimization of the cost function is strictly for performance purposes in the distributed RHC algorithm.

Equation (11) is referred to as the *consistency constraint*, which requires that each predicted trajectory remain close to the assumed trajectory (that neighbors assume for that agent). In particular, the predicted trajectory z_i^p must remain nearly as close to the assumed trajectory $\hat{z}_i$ as the feasible trajectory $\bar{z}_i$, with an added margin of freedom parameterized by $(\varepsilon/2(q+1)N_a)^2$. In the analysis that follows, the consistency constraint (11) is a key equation in proving that $\bar{z}_i$ is a feasible state trajectory at each update. The constant $q \in \{1,2,3,...\}$ is a design parameter, and the choice for q will be motivated in Section 4.1.

Note that the terminal constraint in each optimal control problem is $\Omega_i(\varepsilon/2)$, although Lemma 1 ensures that the larger terminal set $\Omega_i(\varepsilon)$ suffices as a collective region of attraction for the terminal controllers. In the analysis presented in the next section, it is shown that tightening the terminal set in this way is required to guarantee the feasibility properties. Before stating the distributed RHC algorithm, an assumption is made to facilitate the initialization phase.

Assumption 4. Given $z(t_0)$ at initial time t_0, there exists a feasible control $u_i^p(\tau;t_0) \in \mathcal{U}$, $\tau \in [t_0, t_0+T]$, for each agent $i \in \mathcal{V}$, such that the solution to the full system $\dot{z}(\tau) = f(z(\tau), u^p(\tau;t_0))$, denoted $z^p(\cdot;t_0)$, satisfies $z_i^p(t_0+T;t_0) \in \Omega_i(\varepsilon/2)$ and results in a bounded cost $J_i(z_i(t_0), u_i^p(\cdot;t_0))$ for every $i \in \mathcal{V}$. Moreover, each agent $i \in \mathcal{V}$ has access to $u_i^p(\cdot;t_0)$.

Remark 1. Assumption 4 bypasses the difficult task of actually constructing an initially feasible solution in a distributed way. In fact, finding an initially feasible solution for many optimization problems is often a primary obstacle, whether or not such problems are used in

a control setting. As such, many centralized implementations of RHC likewise assume that an initially feasible solution is available [3,12,13].

Let $Z \subset \mathbb{R}^{nN_a}$ denote the set of initial states for which there exists a control satisfying the conditions in Assumption 4. The control algorithm is now stated.

Algorithm 1. The dual-mode *distributed receding horizon control* law for any agent $i \in \mathcal{V}$ is as follows:

Data: $z(t_0)$, $u_i^p(\cdot;t_0)$ satisfying Assumption 4, $T \in (0,\infty)$, $\delta \in (0,T]$, $q \in \{1,2,3,...\}$.

Initialization: At time t_0, if $z(t_0) \in \Omega_\varepsilon$, then apply the terminal controller $u_i(t) = K_i z_i(t)$, for all $t \geq t_0$. Else:

Controller:

1. Over any interval $[t_k, t_{k+1})$, $k \in \mathbb{N}$:

   (a) At any time $t \in [t_k, t_{k+1})$, if $z(t) \in \Omega_\varepsilon$, then apply the terminal controller $u_i(t') = K_i z_i(t')$, for all $t' \geq t$. Else:

   (b) Compute $\hat{z}_i(\tau;t_{k+1})$ according to (4).

   (c) Transmit $\hat{z}_i(\cdot;t_{k+1})$ to every downstream neighbor $l \in \mathcal{N}_i^{\mathrm{d}}$, and receive $\hat{z}_j(\cdot;t_{k+1})$ from every upstream neighbor $j \in \mathcal{N}_i^{\mathrm{u}}$.

   (d) Apply $u_i^p(\tau;t_k)$, $\tau \in [t_k, t_{k+1})$.

2. At any time t_{k+1}, $k \in \mathbb{N}$:

   (a) Compute $\bar{z}_i(\cdot;t_{k+1})$ according to (6).

   (b) Solve Problem 1 for agent i, yielding $u_i^p(\cdot;t_{k+1})$. ■

Part 1(a) of Algorithm 1 presumes that the every agent can obtain the full state $z(t)$. This requirement is a theoretical artifact needed when employing dual-mode control, so that switching occurs only when the conditions of Lemma 1 are satisfied. In the next section, it is shown that the distributed RHC policy drives the state $z(t_l)$ to Ω_ε after a finite number of updates l, and the state remains in Ω_ε for all future time. If Ω_ε is sufficiently small for stability purposes, then agents do not need access to the full state at *any* update, since RHC can be employed for all time. The next section provides the analysis showing that the distributed RHC algorithm is feasible at every update and stabilizing.

4 Analysis

In this section, feasibility is analyzed first in Section 4.1, followed by stability in Section 4.2. Qualitative and quantitative discussion of the results presented is then given in Section 4.3.

4.1 Feasibility

A desirable property of the implementation is that the existence of a feasible solution to Problem 1 at update $k = 0$ implies the existence of a feasible solution for any subsequent update $k \geq 1$. The *main result of this section* is that, provided an initially feasible solution is available and Assumption 4 holds true, a feasible control solution to Problem 1 for any $i \in \mathcal{V}$ and at any time t_k, $k \geq 1$, is $u_i^p(\cdot; t_k) = \bar{u}_i(\cdot; t_k)$, with $\bar{u}_i$ defined by (7). The corresponding feasible state trajectory defined by (6) is $z_i^p(\cdot; t_k) = \bar{z}_i(\cdot; t_k)$. Before presenting the technical details, it is useful to sketch some of the requirements and challenges in guaranteeing this feasibility result.

For any $i \in \mathcal{V}$ and at any update $k \geq 1$, the control and state pair $(\bar{u}_i(\cdot; t_k), \bar{z}_i(\cdot; t_k))$ is *a feasible solution* to Problem 1 if $\bar{u}_i(\cdot; t_k) : [t_k, t_k + T] \to \mathcal{U}$, equations (3), (10) and (11) are satisfied, and the terminal state constraint $\bar{z}_i(t_k + T; t_k) \in \Omega_i(\varepsilon/2)$ is satisfied. Consider the schematic of the different trajectories involved in the problem in Figure 2. The figure shows how the assumed trajectory $\hat{z}_i$ and the feasible trajectory $\bar{z}_i$ must relate to one another, and to the terminal sets $\Omega_i(\varepsilon)$, $\Omega_i(\varepsilon/2)$ and $\Omega_i(\varepsilon'/2)$ where $\varepsilon' \triangleq \varepsilon q/(q+1)$ and $q \in \{1, 2, 3, ...\}$. As specified in the following lemmas, the positive integer q is a design parameter. Note that $\varepsilon' \in (0, \varepsilon)$ for any choice of q.

In this section, Lemma 2 identifies sufficient conditions to ensure that $\hat{z}_i(t; t_k) \in \Omega_i(\varepsilon/2)$ over the interval $t \in [t_{k-1} + T, t_k + T]$, and $\hat{z}_i(t_k + T; t_k) \in \Omega_i(\varepsilon'/2)$, as shown in the figure. Then, Lemma 3 identifies sufficient conditions to ensure that for every $i \in \mathcal{V}$, $\|\bar{z}_i(t; t_k) - \hat{z}_i(t; t_k)\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a})$ for all $t \in [t_k, t_k + T]$, also shown in Figure 2. Using these lemmas, control constraint feasibility is proven in Lemma 4. Finally, Lemmas 2–4 are combined in Theorem 1 to give the main result, which proves that the control and state pair $(\bar{u}_i(\cdot; t_k), \bar{z}_i(\cdot; t_k))$ is a feasible solution to Problem 1 for any $i \in \mathcal{V}$ and at any update $k \geq 1$.

Lemma 2. Suppose that Assumptions 1–4 hold and $z(t_0) \in Z$. For any $k \geq 0$, if Problem 1 has a solution at update time t_k, then

$$\hat{z}_i(t; t_{k+1}) \in \Omega_i(\varepsilon/2), \quad \forall t \in [t_k + T, t_{k+1} + T],$$

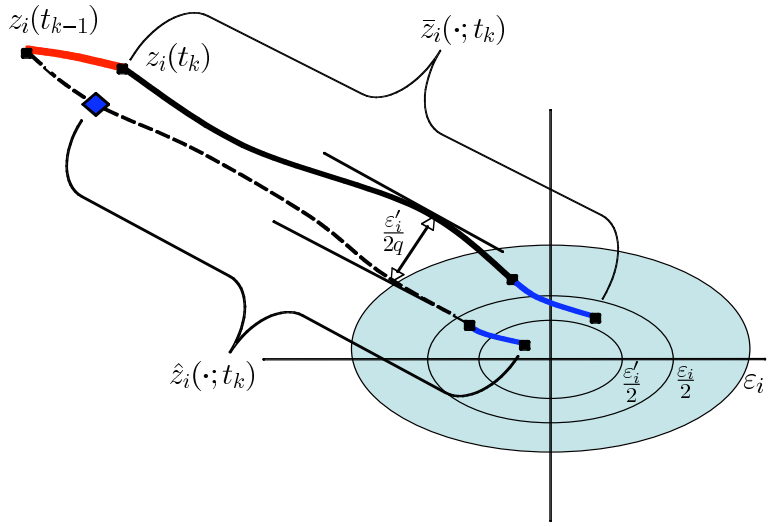


Figure 2: Schematic of the different trajectories involved in the application of the distributed RHC algorithm, for any agent $i \in \mathcal{V}$ and for update times t_{k-1} and t_k. The ellipsoids represent level sets of the function $\|z_i\|_{P_i}^2$, with $\varepsilon_i = \varepsilon/\sqrt{N_a}$ and $\varepsilon_i' = \varepsilon_i q/(q+1)$ for a given positive integer q. The dashed line represents $z_i^p(\cdot;t_{k-1})$, with $z_i^p(t_{k-1};t_{k-1})$ equal to the true initial condition $z_i(t_{k-1})$ and $z_i^p(t_{k-1}+T;t_{k-1})$ reaching the level set $\Omega_i(\varepsilon/2)$ as required in the optimal control problem. All subsystems apply their control $u_i^p(t;t_{k-1})$, for $t \in [t_{k-1},t_k)$, and subsystem i follows the solid (red) line, arriving at $z_i(t_k)$. After δ seconds, on the other hand, the predicted state $z_i^p(t_k;t_{k-1})$ is generally in a different location, represented by the blue diamond. The reason is that i computes $z_i^p(\cdot;t_{k-1})$ assuming neighbors follow $\hat{z}_{-i}(\cdot;t_{k-1})$, when in fact each neighbor is likewise computing and applying an updated predicted control. The assumed trajectory $\hat{z}_i(\cdot;t_k)$ shown is constructed according to (4), comprised of the remainder of $z_i^p(\cdot;t_{k-1})$ concatenated with the decoupled linear controlled response. The feasible trajectory $\bar{z}_i(\cdot;t_k)$ is constructed according to (6) and is also shown in the schematic. To meet certain feasibility requirements (refer to text), parametric conditions are identified to ensure that $\hat{z}_i(t;t_k)$ and $\bar{z}_i(t;t_k)$ are within the indicated ellipsoids over the interval $t \in [t_{k-1}+T, t_k+T]$ (shown by solid blue line), and that $\|\bar{z}_i(t;t_k) - \hat{z}_i(t;t_k)\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a})$ for all $t \in [t_k, t_k+T]$.

$$\text{and} \quad \hat{z}_i(t_{k+1}+T;t_{k+1}) \in \Omega_i(\varepsilon'/2),$$

for every $i \in \mathcal{V}$, where $\varepsilon' \triangleq \varepsilon q/(q+1)$ and $q \in \{1,2,3,...\}$, provided the update parameter δ satisfies

$$\delta \geq \frac{\ln\left[(q+1)^2/q^2\right]}{\min_{i\in\mathcal{V}}\{\lambda_{\min}(P_i^{-1/2}\widehat{Q}_i P_i^{-1/2})\}}. \tag{12}$$

Proof. The following reasoning applies for any agent $i \in \mathcal{V}$. Since Problem 1 has a solution at update time t_k, $\hat{z}_i(\cdot;t_{k+1})$ is well-defined from (4). By construction, it follows from the terminal constraint that $\|\hat{z}_i(t_k+T;t_{k+1})\|_{P_i} = \|z_i^p(t_k+T;t_k)\|_{P_i} \leq \varepsilon/(2\sqrt{N_a})$. With the Lyapunov function $V_i(\hat{z}_i(t)) = \|\hat{z}_i(t;t_{k+1})\|_{P_i}^2$ for $t \in [t_k+T, t_{k+1}+T]$, it follows that

$$\dot{V}_i(\hat{z}_i(t)) \leq -\|\hat{z}_i(t;t_{k+1})\|_{\widehat{Q}_i}^2 \leq -\lambda_{\min}(P_i^{-1/2}\widehat{Q}_i P_i^{-1/2}) V_i(\hat{z}_i(t)).$$

Therefore, $V_i(\hat{z}_i(t_k+T)) \leq \varepsilon^2/(4N_a)$ implies $V_i(\hat{z}_i(t)) \leq \varepsilon^2/(4N_a)$ for all $t \in [t_k+T, t_{k+1}+T]$, completing the first part of the proof. From the Comparison Lemma [11],

$$V_i(\hat{z}_i(t)) \leq \exp[-(t-(t_k+T))\lambda_{\min}(P_i^{-1/2}\widehat{Q}_i P_i^{-1/2})] V_i(\hat{z}_i(t_k+T)).$$

To ensure $\|\hat{z}_i(t_{k+1}+T;t_{k+1})\|_{P_i} \leq \varepsilon q/(2(q+1)\sqrt{N_a})$, a sufficient condition is that

$$\exp[-\delta\lambda_{\min}(P_i^{-1/2}\widehat{Q}_i P_i^{-1/2})] V_i(\hat{z}_i(t_k+T)) \leq \frac{\varepsilon^2}{4N_a}\frac{q^2}{(q+1)^2}.$$

To ensure the bound holds for every $i \in \mathcal{V}$, a sufficient condition on δ is

$$\delta \min_{i\in\mathcal{V}} \left\{\lambda_{\min}(P_i^{-1/2}\widehat{Q}_i P_i^{-1/2})\right\} \geq \ln[(q+1)^2/q^2],$$

concluding the proof. ■

Condition (12) suggests a minimum amount of time required to steer each $z_i^p(t_k+T;t_k)$ from $\Omega_i(\varepsilon/2)$ to $\Omega_i(\varepsilon'/2)$, using the decoupled terminal controllers. The larger the chosen value of q, the smaller $\ln[(q+1)^2/q^2]$ becomes and the smaller the required lower bound on the update parameter. Likewise, ε' approaches ε as q increases, so it should require less time to drive $z_i^p(t_k+T;t_k)$ from $\Omega_i(\varepsilon/2)$ to $\Omega_i(\varepsilon'/2)$ for larger values of q.

The analysis in Lemmas 3 will require a local Lipschitz property on the collective dynamics. In vector form, the collective set of differential equations for the predicted trajectories

(using (3) for each $i \in \mathcal{V}$) is denoted

$$\dot{z}^p(t;t_k) = F(z^p(t;t_k), \hat{z}(t;t_k), u^p(t;t_k)), \quad t \in [t_k, t_k + T],$$

where $F : \mathbb{R}^{nN_a} \times \mathbb{R}^{nN_a} \times \mathbb{R}^{mN_a} \to \mathbb{R}^{nN_a}$, $z^p = (z_1^p, ..., z_{N_a}^p)$, $\hat{z} = (\hat{z}_1, ..., \hat{z}_{N_a})$ and $u^p = (u_1^p, ..., u_{N_a}^p)$. By definition, the function F satisfies $F(z, z', u) = f(z, u)$ whenever $z = z'$.

Assumption 5. Given P and R, there exist positive constants β and γ such that the Lipschitz bound

$$\|F(z, z', u) - F(y, y', v)\|_P \leq \|z - y\|_P + \beta\|z' - y'\|_P + \gamma\|u - v\|,$$

holds for all $z, z', y, y' \in Z$, and $u, v \in \mathcal{U}^{N_a}$.

More generally, the Lipschitz bound would take the form $\|\widehat{F}(z, z', u)\|_P \leq \hat{\alpha}\|z\|_P + \hat{\beta}\|z'\|_P + \hat{\gamma}\|u\|$ for some positive constants $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$. Thus, Assumption 5 presumes that one can identify the Lipschitz constants $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$, and that the differential equation f (or F) is already normalized by $\hat{\alpha}$, so that $\beta = \hat{\beta}/\hat{\alpha}$ and $\gamma = \hat{\gamma}/\hat{\alpha}$. The local Lipschitz constant β represents a normalized measure of the amount of coupling in the collective dynamic model. The following lemma makes use of the Lipschitz parameters stated above, as well as a design parameter $r \in \{1, 2, 3, ...\}$ in addition to the design parameter q introduced in Lemma 2.

Lemma 3. Suppose that Assumptions 1–5 hold and $z(t_0) \in Z$. For any $k \geq 0$, if Problem 1 has a solution at every update time t_l, $l = 0, ..., k$, then,

$$\|\bar{z}(t;t_{k+1}) - \hat{z}(t;t_{k+1})\|_P \leq \frac{\varepsilon}{2(q+1)\sqrt{N_a}}, \quad \forall t \in [t_{k+1}, t_{k+1} + T],$$

provided the following parametric conditions hold:

$$\delta c\sqrt{N_a} \exp\left[\delta(1 + \gamma_{_K})\right] \frac{(q+1)(r+1)}{r} \;\leq\; 1, \tag{13}$$

$$\beta 2T(r+1) \exp\left[T + \delta(1 + \beta + \gamma_{_K})\right] \;\leq\; 1, \tag{14}$$

where $q, r \in \{1, 2, 3, ...\}$, $c = 1/(10\lambda_{\max}(\widehat{Q}^{-1/2} P \widehat{Q}^{-1/2})) + \lambda_{\max}^{1/2}(P^{-1/2} A_{\mathrm{o}}^T P A_{\mathrm{o}} P^{-1/2})$, and $\gamma_{_K} = \gamma \lambda_{\max}^{1/2}(P^{-1/2} K^T K P^{-1/2})$.

Proof. Define the functions

$$y(t;t_l) \triangleq \|z^p(t;t_l) - \hat{z}(t;t_l)\|_P, \quad \Theta(t;t_l) \triangleq \|\bar{z}(t;t_l) - \hat{z}(t;t_l)\|_P,$$

for all $t \in [t_l, t_l + T]$. By assuming the existence of a solution to Problem 1 at each update $t_0, ..., t_k$, the functions $y(t; t_l)$ and $\Theta(t; t_l)$ are all well defined for $l = 1, ..., k$. Additionally, $y(t_{k+1}; t_{k+1})$ and $\Theta(t; t_{k+1})$ are well defined. The proof follows by making use of the local Lipschitz bounds stated in Assumption 5, and recursive use of the triangle and Gronwall-Bellman inequalities. To begin, observe that from the consistency constraint (11), for each $l = 1, ..., k$ and for all $t \in [t_l, t_l + T]$,

$$\|z_i^p(t; t_l) - \hat{z}_i(t; t_l)\|_{P_i}^2 \leq \|\bar{z}_i(t; t_l) - \hat{z}_i(t; t_l)\|_{P_i}^2 + \left[\frac{\varepsilon}{2(q+1)N_a}\right]^2 \quad \forall i \in \mathcal{V},$$

$$\implies y^2(t; t_l) \leq \Theta^2(t; t_l) + \left[\frac{\varepsilon}{2(q+1)\sqrt{N_a}}\right]^2 \implies y(t; t_l) \leq \Theta(t; t_l) + \frac{\varepsilon}{2(q+1)\sqrt{N_a}}.$$

Next, a bound on the deviation of the predicted state from the actual state over any update period $[t_{l-1}, t_l]$ is quantified. Define the function $v(t; t_{l-1}) = \|z(t; t_{l-1}) - z^p(t; t_{l-1})\|_P$ for all $l = 1, ..., k+1$, where $z(t; t_{l-1}) = z(t)$ is the actual closed-loop response, satisfying $\dot{z}(t) = f(z(t), u^p(t; t_{l-1}))$ for all $t \in [t_{l-1}, t_l]$ (same as (5)). Observe that $y(t_l; t_l) = v(t_l; t_{l-1})$ for all $l = 1, ..., k+1$. Also, $v(t; t_0) \equiv 0$ for all $t \in [t_0, t_1]$ by Assumption 4, and so $y(t_1; t_1) \equiv 0$. Using the notation F from Assumption 5, the model for the actual response can also be written as $\dot{z}(t) = F(z(t), z(t), u^p(t; t_l))$. A bound on $v(t; t_{l-1})$ for $l = 2, ..., k+1$ proceeds as follows,

$$\begin{aligned} v(t; t_{l-1}) &= \Big\| \int_{t_{l-1}}^{t} F(z(s; t_{l-1}), z(s; t_{l-1}), u^p(s; t_{l-1})) - F(z^p(s; t_{l-1}), \hat{z}(s; t_{l-1}), u^p(s; t_{l-1})) \mathrm{d}s \Big\|_P \\ &\leq \int_{t_{l-1}}^{t} v(s; t_{l-1}) + \beta \|z(s; t_{l-1}) - \hat{z}(s; t_{l-1}) \pm z^p(s; t_{l-1})\|_P \mathrm{d}s \\ &\leq \int_{t_{l-1}}^{t} (1+\beta) v(s; t_{l-1}) + \beta y(s; t_{l-1}) \mathrm{d}s. \end{aligned}$$

Using the Gronwall-Bellman inequality yields

$$y(t_l; t_l) = v(t_l; t_{l-1}) \leq \beta e^{\delta(1+\beta)} \int_{t_{l-1}}^{t_l} y(s; t_{l-1}) \mathrm{d}s.$$

Next, a bound on $\Theta(t;t_{k+1})$ is derived. On the time interval $t \in [t_{k+1}, t_k + T]$,

$$\begin{aligned}\Theta(t;t_{k+1}) &= \Big\| \bar{z}(t_{k+1};t_{k+1}) - \hat{z}(t_{k+1};t_{k+1}) + \int_{t_{k+1}}^{t} F(\bar{z}(s;t_{k+1}), z^p(s;t_k), u^p(s;t_k)) \\ &\qquad\qquad - F(\hat{z}(s;t_{k+1}), \hat{z}(s;t_k), u^p(s;t_k))\mathrm{d}s \Big\|_P \\ &\leq y(t_{k+1};t_{k+1}) + \int_{t_{k+1}}^{t} \Theta(s;t_{k+1}) + \beta y(s;t_k)\mathrm{d}s \\ &\leq \int_{t_{k+1}}^{t} \Theta(s;t_{k+1})\mathrm{d}s + \beta e^{\delta(1+\beta)} \int_{t_k}^{t_k+T} y(s;t_k)\mathrm{d}s.\end{aligned}$$

Using the Gronwall-Bellman inequality on the time interval $[t_{k+1}, t_k + T]$ yields

$$\Theta(t;t_{k+1}) \leq \Gamma_k, \qquad \text{where} \quad \Gamma_0 \triangleq 0, \quad \Gamma_k \triangleq \beta e^{T+\delta\beta} \int_{t_k}^{t_k+T} y(s;t_k)\mathrm{d}s, \quad k \geq 1.$$

The bound on $\Theta(t;t_{k+1})$ over the time domain $t \in [t_k + T, t_{k+1} + T]$ is given by

$$\begin{aligned}\Theta(t;t_{k+1}) = &\Big\| \bar{z}(t_k+T;t_{k+1}) - \hat{z}(t_k+T;t_{k+1}) + \int_{t_k+T}^{t} F(\bar{z}(s;t_{k+1}), \hat{z}(s;t_{k+1}), K\bar{z}(s;t_{k+1})) \\ &- A_{\mathrm{d}}\hat{z}(s;t_{k+1}) \pm F(\hat{z}(s;t_{k+1}), \hat{z}(s;t_{k+1}), K\hat{z}(s;t_{k+1})) \pm A_{\mathrm{o}}\hat{z}(s;t_{k+1})\mathrm{d}s \Big\|_P \\ \leq &\Theta(t_k+T;t_{k+1}) + \int_{t_k+T}^{t} (1+\gamma_K)\Theta(s;t_{k+1}) + \|\phi(\hat{z}(s;t_{k+1}))\|_P + \|A_{\mathrm{o}}\hat{z}(s;t_{k+1})\|_P \mathrm{d}s,\end{aligned}$$

where $\phi(z) \triangleq F(z,z,Kz) - (A_{\mathrm{d}} + A_{\mathrm{o}})z$. From the proof of Lemma 1, $\|\phi(\hat{z}(s;t_{k+1}))\|_P \leq \|\hat{z}(s;t_{k+1})\|_P / (10\lambda_{\max}(\widehat{Q}^{-1/2}P\widehat{Q}^{-1/2}))$, since $\hat{z}(s;t_{k+1}) \in \Omega_\varepsilon$ for all $s \in [t_k + T, t_{k+1} + T]$ by construction. From Lemma 2, $\hat{z}_i(s;t_{k+1}) \in \Omega_i(\varepsilon/2)$ for all $s \in [t_k + T, t_{k+1} + T]$, and so $\|\hat{z}(s;t_{k+1})\|_P \leq \varepsilon/2$. With c defined in the statement of the lemma, then,

$$\|\phi(\hat{z}(s;t_{k+1}))\|_P + \|A_{\mathrm{o}}\hat{z}(s;t_{k+1})\|_P \;\leq\; c\varepsilon/2, \quad \forall s \in [t_k + T, t_{k+1} + T]$$

Using the bound on $\Theta(t_k + T;t_{k+1})$ and applying the Gronwall-Bellman inequality over the time domain $t \in [t_k + T, t_{k+1} + T]$ yields

$$\Theta(t;t_{k+1}) \leq e^{\delta(1+\gamma_K)} \left[\Gamma_k + \delta c\varepsilon/2\right]. \tag{15}$$

Observe that this bound holds over the entire time interval $[t_{k+1}, t_{k+1}+T]$. From the previous

bound on $y(t;t_k)$ for all $t \in [t_k, t_k+T]$, the bound above can be rewritten as

$$\Theta(t;t_{k+1}) \leq \beta e^{T+\delta(1+\beta+\gamma_K)} \int_{t_k}^{t_k+T} \left[\Theta(s;t_k) + \frac{\varepsilon}{2(q+1)\sqrt{N_a}}\right] \mathrm{d}s \; + \; \frac{\delta c \varepsilon}{2} e^{\delta(1+\gamma_K)}. \qquad (16)$$

By induction on k, the result of the lemma is now proven. For the base case ($k=0$), since $\Gamma_0 = 0$, (15) and (13) imply that

$$\Theta(t;t_1) \; \leq \; \frac{\delta c \varepsilon}{2} e^{\delta(1+\gamma_K)} \; \leq \; \frac{r}{(r+1)(q+1)} \frac{\varepsilon}{2\sqrt{N_a}} \; \leq \; \frac{\varepsilon}{2(q+1)\sqrt{N_a}},$$

for any $r \in \{1,2,3,...\}$ and for all $t \in [t_1, t_1+T]$. Now, assuming $\Theta(t;t_k) \leq \varepsilon/(2(q+1)\sqrt{N_a})$, it must be shown that the same bound holds for $\Theta(t;t_{k+1})$. From (16), the inductive hypothesis and (13), it follows that

$$\begin{aligned} \Theta(t;t_{k+1}) \; &\leq \beta T e^{T+\delta(1+\beta+\gamma_K)} \frac{\varepsilon}{(q+1)\sqrt{N_a}} \; + \; \frac{r}{(r+1)(q+1)} \frac{\varepsilon}{2\sqrt{N_a}} \\ &\leq \frac{1}{2(r+1)} \frac{\varepsilon}{(q+1)\sqrt{N_a}} \; + \; \frac{r}{(r+1)(q+1)} \frac{\varepsilon}{2\sqrt{N_a}} = \frac{\varepsilon}{2(q+1)\sqrt{N_a}}, \end{aligned}$$

where the second line results from condition (14). By the Principle of Mathematical Induction, $\Theta(t;t_{k+1}) \leq \varepsilon/(2(q+1)\sqrt{N_a})$ holds for all $t \in [t_{k+1}, t_{k+1}+T]$ and any $k \geq 0$, concluding the proof. ■

To ensure that the control and state pair $(\bar{u}_i(\cdot;t_k), \bar{z}_i(\cdot;t_k))$ is a feasible solution to Problem 1 at every update $k \geq 1$, the analysis that follows makes use of the bound $\|\bar{z}_i(t;t_k) - \hat{z}_i(t;t_k)\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a})$ for all $t \in [t_k, t_k+T]$ and every $i \in \mathcal{V}$ (shown in Figure 2). A conservative sufficient condition to ensure this bound holds for every $i \in \mathcal{V}$ is to require $\|\bar{z}(t;t_{k+1}) - \hat{z}(t;t_{k+1})\|_P \leq \varepsilon/(2(q+1)\sqrt{N_a})$, and Lemma 3 identifies parametric conditions on the update period δ and coupling parameter β that ensure this bound holds. The reason for doing the analysis on the full vector norm $\|\bar{z} - \hat{z}\|_P$ in Lemma 3, as opposed to the individual norms $\|\bar{z}_i - \hat{z}_i\|_{P_i}$, is that this choice facilitated a proof which relied heavily on the Gronwall-Bellman inequality.

The purpose of the design parameters $q, r \in \{1,2,3,...\}$ can now be clarified. Equation (13) places an upper bound on the update period δ, which can be rewritten and combined

with the lower bound (12) to give

$$\frac{e^{\delta(1+\gamma_K)}\ln\left[(q+1)^2/q^2\right]}{\min_{i\in\mathcal{V}}\{\lambda_{\min}(P_i^{-1/2}\widehat{Q}_iP_i^{-1/2})\}} \;\leq\; \delta e^{\delta(1+\gamma_K)} \;\leq\; \frac{r}{(r+1)(q+1)c\sqrt{N_a}}.$$

The larger the chosen value of q, the smaller the lower and upper bounds on δ. The ability to shift the feasible range for δ is useful for design purposes, as will be demonstrated in the example of coupled oscillators considered in Section 5. Also, larger values of q reduce the margin in the consistency constraint (11) that bounds how much the predicted state can deviate from the assumed state. Equation (14) places an upper bound on the Lipschitz coupling constant β, which can be rewritten as

$$\beta e^{\delta\beta} \leq \frac{1}{2T(r+1)\exp[T+\delta(1+\gamma_K)]}.$$

By increasing the design parameter r, one can increase the upper bound on δ at the price of requiring a tighter bound on β. The utility of being able to choose r will be demonstrated in Section 5 as well. To proceed with the feasibility results of this section, it is now shown that $\bar{u}_i$ satisfies the control constraints.

Lemma 4. Suppose that Assumptions 1–5 hold, $z(t_0) \in Z$ and conditions (12)–(14) are satisfied. For any $k \geq 0$, if Problem 1 has a solution at every update time t_l, $l = 0, ..., k$, then, $\bar{u}_i(\tau; t_{k+1}) \in \mathcal{U}$ for all $\tau \in [t_{k+1}, t_{k+1} + T]$ and for every $i \in \mathcal{V}$.

Proof. Since Problem 1 has a feasible solution at t_k, $\bar{u}_i(\cdot; t_{k+1})$ is well-defined. Since Lemma 3 will be invoked, a feasible solution to Problem 1 must be assume at each update $t_0, ..., t_k$. Now, since $\bar{u}_i(t; t_{k+1}) = u_i^p(t; t_k)$ for all $t \in [t_{k+1}, t_k + T]$, it need only be shown that the remainder of $\bar{u}_i$ is in $\mathcal{U}$. A sufficient condition for this is if $\bar{z}_i(t; t_{k+1}) \in \Omega_i(\varepsilon)$ for all $t \in [t_k + T, t_{k+1}+T]$, since ε is chosen to satisfy the conditions of Lemma 1 and, consequently, $K_i z_i \in \mathcal{U}$ for all $i \in \mathcal{V}$ when $z \in \Omega_\varepsilon$. From Lemma 3, $\|\bar{z}_i(t; t_{k+1}) - \hat{z}_i(t; t_{k+1})\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a})$ for all $t \in [t_k+T, t_{k+1}+T]$. From Lemma 2, $\|\hat{z}_i(t; t_{k+1})\|_{P_i} \leq \varepsilon/(2\sqrt{N_a})$ for all $t \in [t_k+T, t_{k+1}+T]$. Using the triangle inequality gives $\|\bar{z}_i(t; t_{k+1})\|_{P_i} = \|\bar{z}_i(t; t_{k+1}) \pm \hat{z}_i(t; t_{k+1})\|_{P_i} \leq \|\bar{z}_i(t; t_{k+1}) - \hat{z}_i(t; t_{k+1})\|_{P_i} + \|\hat{z}_i(t; t_{k+1})\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a}) + \varepsilon/(2\sqrt{N_a}) < \varepsilon/\sqrt{N_a}$, since $q \geq 1$. Therefore, $\bar{z}_i(t; t_{k+1}) \in \Omega_i(\varepsilon)$ for all $t \in [t_k + T, t_{k+1} + T]$ for every $i \in \mathcal{V}$, concluding the proof. ■

The first main theorem of the paper is now stated.

Theorem 1. Suppose that Assumptions 1–4 hold, $z(t_0) \in Z$ and conditions (12)–(14) are satisfied. Then, for every agent $i \in \mathcal{V}$, the control and state pair $(\bar{u}_i(\cdot; t_k), \bar{z}_i(\cdot; t_k))$, defined by equations (6) and (7), is a feasible solution to Problem 1 at every update $k \geq 1$.

Proof. The proof follows by strong induction. First, the $k = 1$ case. The trajectory $\bar{u}_i(\cdot; t_1)$ trivially satisfies the constraint (10). The corresponding state trajectory $\bar{z}_i(\cdot; t_1)$ satisfies the dynamic equation (3) and the consistency constraint (11). Now, observe that $\hat{z}_i(t_1; t_1) = z_i^p(t_1; t_0) = \bar{z}_i(t_1; t_1) = z_i(t_1)$ for every $i \in \mathcal{V}$. Additionally, $\bar{z}_i(t; t_1) = z_i^p(t; t_0)$ for all $t \in [t_1, t_0 + T]$, and so $\bar{z}_i(t_0 + T; t_1) \in \Omega_i(\varepsilon/2)$. By the invariance properties of the terminal controller and the conditions in Lemma 1, it follows that the terminal state and control constraints are also satisfied, concluding the $k = 1$ case. Now, the induction step. By assumption, suppose $u_i^p(\cdot; t_l) = \bar{u}_i(\cdot; t_l)$ is a feasible solution for $l = 1, ..., k$. It must be shown that $\bar{u}_i(\cdot; t_{k+1})$ is a feasible solution at update $k + 1$. As before, the constraint (10) and the consistency constraint (11) are trivially satisfied, and $\bar{z}_i(\cdot; t_{k+1})$ is the corresponding state trajectory that satisfies the dynamic equation. Since there is a solution for Problem 1 at updates $l = 1, ..., k$, Lemmas 2–4 can be invoked. Lemma 4 guarantees control constraint feasibility. The terminal constraint requires $\bar{z}_i(t_{k+1} + T; t_{k+1}) \in \Omega_i(\varepsilon/2)$, for each $i \in \mathcal{V}$. From Lemma 3, $\|\bar{z}_i(t_{k+1} + T; t_{k+1}) - \hat{z}_i(t_{k+1} + T; t_{k+1})\|_{P_i} \leq \varepsilon/(2(q+1)\sqrt{N_a})$, and Lemma 2 guarantees that $\|\hat{z}_i(t_{k+1} + T; t_{k+1})\|_{P_i} \leq \varepsilon q/(2(q+1)\sqrt{N_a})$. Combining these two bounds and using the triangle inequality implies $\|\bar{z}_i(t_{k+1} + T; t_{k+1})\| \leq \varepsilon/(2\sqrt{N_a})$ for each $i \in \mathcal{V}$, concluding the proof. ■

The existence of a feasible solution at every update is beneficial for numerical implementation purposes. In the next section, the stability of the closed-loop system is analyzed.

4.2 Stability

The stability of the closed-loop system (5) is now analyzed.

Theorem 2. Suppose that Assumptions 1–5 hold, $z(t_0) \in Z$, conditions (12)–(14) are satisfied, and the following parametric conditions hold

$$T \geq 8\delta, \quad (q+1) \geq 2\frac{T - \delta}{\delta}. \tag{17}$$

Then, by application of Algorithm 1, the closed-loop system (5) is asymptotically stabilized to the origin.

Proof. From part 1(a) of Algorithm 1 and Lemma 1, if $z(t) \in \Omega_\varepsilon$ for any $t \geq 0$, the terminal controllers take over and stabilize the system to the origin. Therefore, it remains to show that if $z(t_0) \in Z \setminus \Omega_\varepsilon$, then by application of Algorithm 1, the closed-loop system (5) is driven to the set Ω_ε in finite time. Define the non-negative function

$$\mathbb{V}_k = \int_{t_k}^{t_k+T} \|z^p(s;t_k)\|_P \, \mathrm{d}s.$$

In the following, it is shown that for any $k \geq 0$, if $z(t) \in Z \setminus \Omega_\varepsilon$ for all $t \in [t_k, t_{k+1}]$, then there exists a constant $\eta \in (0, \infty)$ such that $\mathbb{V}_{k+1} \leq \mathbb{V}_k - \eta$. By the constraint (10)

$$\|z_i^p(t;t_k)\|_{P_i}^2 \leq \|\bar{z}_i(t;t_k)\|_{P_i}^2 + \left[\frac{\delta\varepsilon}{8T\sqrt{N_a}}\right]^2, \forall i \in \mathcal{V} \implies \|z^p(t;t_k)\|_P^2 \leq \left[\|\bar{z}(t;t_k)\|_P + \frac{\delta\varepsilon}{8T}\right]^2,$$

which implies $\|z^p(t;t_k)\|_P \leq \|\bar{z}(t;t_k)\|_P + \delta\varepsilon/(8T)$. Therefore,

$$\mathbb{V}_{k+1} = \int_{t_{k+1}}^{t_{k+1}+T} \|z^p(s;t_{k+1})\|_P \, \mathrm{d}s \leq \int_{t_{k+1}}^{t_{k+1}+T} \|\bar{z}(s;t_{k+1})\|_P \, \mathrm{d}s + \frac{\delta\varepsilon}{8}.$$

Subtracting $\mathbb{V}_k$ from $\mathbb{V}_{k+1}$ gives

$$\begin{aligned}\mathbb{V}_{k+1} - \mathbb{V}_k \leq &- \int_{t_k}^{t_{k+1}} \|z^p(s;t_k)\|_P \, \mathrm{d}s + \int_{t_k+T}^{t_{k+1}+T} \|\bar{z}(s;t_{k+1})\|_P \, \mathrm{d}s \\ &+ \int_{t_{k+1}}^{t_k+T} [\|\bar{z}(s;t_{k+1})\|_P - \|\hat{z}(s;t_{k+1})\|_P] \, \mathrm{d}s + \frac{\delta\varepsilon}{8}.\end{aligned}$$

The actual closed-loop state response $z(t;t_k) = z(t)$ for $t \in [t_k, t_{k+1}]$ can be bounded as

$$\|z(t;t_k)\|_P \leq \|z(t;t_k) - z^p(t;t_k)\|_P + \|z^p(t;t_k)\|_P.$$

Assuming $z(t) \in Z \setminus \Omega_\varepsilon$ for all $t \in [t_k, t_{k+1}]$, it must be that

$$\|z^p(t;t_k)\|_P \geq \varepsilon - \|z(t;t_k) - z^p(t;t_k)\|_P.$$

From the proof of Lemma 3,

$$\|z(t;t_k) - z^p(t;t_k)\|_P = v(t;t_k) \leq \beta e^{(t-t_k)(1+\beta)} \int_{t_k}^{t_{k+1}} y(s;t_k)\mathrm{d}s \leq \frac{\beta\delta e^{\delta(1+\beta)}\varepsilon}{(q+1)\sqrt{N_a}}.$$

Therefore, using (14), (17), and that $q, r, N_a \geq 1$,

$$\|z^p(t;t_k)\|_P \;\geq\; \varepsilon - \frac{\beta T e^{\delta(1+\beta)}\varepsilon}{8(q+1)\sqrt{N_a}} \;\geq\; \varepsilon - \frac{\varepsilon}{16(r+1)(q+1)\sqrt{N_a}e^{T+\delta\gamma_K}} \;\geq\; \frac{63\varepsilon}{64}.$$

From the proof of Lemma 4, on the interval $t \in [t_k + T, t_{k+1} + T]$

$$\|\bar{z}_i(t;t_{k+1})\|_{P_i} \leq \frac{\varepsilon(q+2)}{2(q+1)\sqrt{N_a}} \quad\Longrightarrow\quad \|\bar{z}(s;t_{k+1})\|_P \leq \frac{\varepsilon(q+2)}{2(q+1)}.$$

For the remaining term, using the triangle inequality,

$$\|\bar{z}(t;t_{k+1})\|_P - \|\hat{z}(t;t_{k+1})\|_P \;\leq\; \|\bar{z}(t;t_{k+1}) - \hat{z}(t;t_{k+1})\|_P = \Theta(t;t_{k+1}).$$

From the proof of Lemma 3 and (14), on the interval $t \in [t_{k+1}, t_k + T]$

$$\begin{aligned}\|\bar{z}(t;t_{k+1}) - \hat{z}(t;t_{k+1})\|_P \;&\leq\; \beta e^{T+\delta\beta}\int_{t_k}^{t_k+T} y(s;t_k)\mathrm{d}s \;\leq\; \frac{\beta T e^{T+\delta\beta}\varepsilon}{(q+1)\sqrt{N_a}} \\ &\leq\; \frac{\varepsilon}{2(r+1)(q+1)\sqrt{N_a}e^{\delta(1+\gamma_K)}} \;\leq\; \frac{\varepsilon}{4(q+1)}.\end{aligned}$$

Combining terms and integrating yields

$$\mathbb{V}_{k+1} - \mathbb{V}_k \;\leq \delta\varepsilon\left[-\frac{63}{64} + \frac{q+2}{2(q+1)} + \frac{T-\delta}{4\delta(q+1)} + \frac{1}{8}\right]$$

From (17), $(T-\delta)/(4\delta(q+1)) \leq 1/8$, and $q+1 \geq 14$ which implies $(q+2)/(q+1) \leq 15/14$. Combining these bounds yields

$$\mathbb{V}_{k+1} - \mathbb{V}_k \;\leq \delta\varepsilon\left[-\frac{47}{64} + \frac{15}{28}\right] \;\leq\; -\frac{\delta\varepsilon}{6} \triangleq -\eta.$$

Thus, for any $k \geq 0$, if $z(t_k), z(t_{k+1}) \in Z \setminus \Omega_\varepsilon$, then there exists a constant $\eta \in (0, \infty)$ such that $\mathbb{V}_{k+1} \leq \mathbb{V}_k - \eta$. From this inequality, it follows by contradiction that there exists a finite time t' such that $z(t') \in \Omega_\varepsilon$. If this were not the case, the inequality implies $\mathbb{V}_k \to -\infty$ as $k \to \infty$. However, $\mathbb{V}_k \geq 0$; therefore, there exists a finite time t' such that $z(t') \in \Omega_\varepsilon$, concluding the proof. ■

The feasibility and stability results in this paper are closely related to those of Michalska and Mayne [13], who demonstrated robust feasibility and stability in the presence of model

error by placing parametric bounds on (combinations of) the update period and a Lipschitz constant. While there is no model error here, bounds are likewise derived to ensure robustness to the bounded discrepancy between what agents do, and what their neighbors believe they will do.

4.3 Comparison of Complexity Bounds

In this section, the computation and communication complexity bounds are compared between the distributed RHC algorithm and a centralized RHC implementation. In the centralized implementation, a single node is presumed to do all computations and communicate directly with all agents. For both implementations, complexity bounds are compared for a single RHC update period. It is also assumed for both implementations that the optimization problem is discretized and transcribed into a nonlinear programming problem (NLP). In general, an NLP with ρ variables has computational complexity $\mathcal{O}(\rho^3)$. If the optimization problem is quadratic, the exponent on the variables changes from 3 to 2. It is assumed that the optimization problem is formulated to have the control as the free variable, and the state is assumed to be determined uniquely from the differential equation. Let the time domain $[t_k, t_k + T]$ for any RHC update k be discretized into $N_d - 1$ intervals, and so there are N_d discretization points in time.

For computational complexity bounds, consider the cost of solving an optimal control problem at any RHC update. In a single centralized optimal control problem, the total number of variables at each update is mN_dN_a, since there are mN_d variables per agent (m dimensional vector of control variables at each time discretization point) and there are N_a total agents. Consequently, the computational complexity is $\mathcal{O}\left((mN_dN_a)^3\right)$. In contrast, from step 2(b) of Algorithm 1, the computational complexity bound for any single agent $i \in \mathcal{V}$ is $\mathcal{O}\left((mN_d)^3\right)$. The distributed implementation clearly offers a substantial savings in computational cost, particularly if N_a is large.

The communication complexity is defined as being the total number of variables being transmitted and received during any single RHC update period $[t_k, t_{k+1})$. For example, if a node sends ρ_1 variables and receives ρ_2 variables, the bound for that node is $\rho_1 + \rho_2$. For a dual-mode implementation of RHC, be it centralized or distributed, the full state $z(t)$ must be monitored continuously (in theory) to determine if the state has entered Ω_ε, at which time the control switches from RHC to the terminal controllers. Since the cost of monitoring $z(t)$ is mutual, it is left out of the comparison between communication costs of the two implementations.

The centralized implementation requires that every agent send its initial condition (dimension n) to the computing node. However, this cost is being ignored, as we are ignoring the cost of the availability of $z(t)$. The centralized RHC update is complete once the computing node transmits the RHC law to every agent. Since the update period is typically much smaller than the planning horizon, one could assume $\delta N_d = T$. In this case, the cost of transmitting the RHC law to each agent is proportional to m, the dimension of the control vector. Since the centralized computing node must send m variables to every agent, the communication complexity at the centralized node is mN_a. To compute the communication complexity bound for the distributed implementation for any agent $i \in \mathcal{V}$, refer to Algorithm 1. Between RHC update times, i must transmit a trajectory to $|\mathcal{N}_i^{\mathrm{d}}|$ neighbors and receive a trajectory from $|\mathcal{N}_i^{\mathrm{u}}|$ neighbors, where any such trajectory is proportional to nN_d. Thus, the communication complexity at distributed node i is $nN_d(|\mathcal{N}_i^{\mathrm{u}}| + |\mathcal{N}_i^{\mathrm{d}}|)$.

In comparing the bounds, since it is typical that $m \leq n$, the stated communication cost of the distributed implementation is *typically higher* than the stated communication cost of the centralized node implementation. However, if $m \approx n$ and the graph $\mathcal{G}$ is sparse such that $|\mathcal{N}_i^{\mathrm{u}}| + |\mathcal{N}_i^{\mathrm{d}}|$ is small for any $i \in \mathcal{V}$, then the cost of the distributed implementation need not be substantially higher than that of the centralized node implementation. The computation and communication complexity bound comparisons are summarized in Table 1.

Table 1: Comparison of computation and communication complexity bounds for centralized and distributed RHC algorithms over a single RHC update period. Variables are the number of agents N_a, the dimension of each agents control vector m, the dimension of each agents state vector n, the common number of time intervals $N_d - 1$ in each discretized optimization problem, and the number of upstream $|\mathcal{N}_i^{\mathrm{u}}|$ and downstream $|\mathcal{N}_i^{\mathrm{d}}|$ neighbors of any agent $i \in \mathcal{V}$.

	Centralized Node	Distributed Node (any $i \in \mathcal{V}$)
Computation	$\mathcal{O}\left((mN_dN_a)^3\right)$	$\mathcal{O}\left((mN_d)^3\right)$
Communication	mN_a	$nN_d(\lvert\mathcal{N}_i^{\mathrm{u}}\rvert + \lvert\mathcal{N}_i^{\mathrm{d}}\rvert)$

5 Coupled Oscillators

In this section, the example of three coupled Van der Pol oscillators is considered for application of the distributed RHC algorithm. The three oscillators modeled here are physically meaningful in that they capture the thigh and knee dynamics of a walking robot experi-

ment [8]. In the following, $\theta_1 \in [-\pi/2, \pi/2]$ is the relative angle between the two thighs, $\theta_2 \in [0, \pi/2]$ is the right knee angle (relative to the right thigh), and $\theta_3 \in [-\pi/2, 0]$ is the left knee angle (relative to left thigh). The angle θ_3 is negative by convention [8]. The controlled equations of motion in units of $(\text{rad/sec})^2$ are

$$\begin{aligned}
\ddot{\theta}_1(t) &= 0.1\left[1 - 5.25\theta_1^2(t)\right]\dot{\theta}_1(t) - \theta_1(t) + u_1(t) \\
\ddot{\theta}_2(t) &= 0.01\left[1 - p_2\left(\theta_2(t) - \theta_{2e}\right)^2\right]\dot{\theta}_2(t) - 4(\theta_2(t) - \theta_{2e}) \\
&\quad + 0.057\theta_1(t)\dot{\theta}_1(t) + 0.1(\dot{\theta}_2(t) - \dot{\theta}_3(t)) + u_2(t) \\
\ddot{\theta}_2(t) &= 0.01\left[1 - p_3\left(\theta_3(t) - \theta_{3e}\right)^2\right]\dot{\theta}_3(t) - 4(\theta_3(t) - \theta_{3e}) \\
&\quad + 0.057\theta_1(t)\dot{\theta}_1(t) + 0.1(\dot{\theta}_3(t) - \dot{\theta}_2(t)) + u_3(t), \\
&\text{subject to} \quad |u_i(t)| \leq 1, \ \forall t \geq 0, \ i = 1, 2, 3.
\end{aligned}$$

Two-phase biped locomotion is generated by these equations with zero control (open-loop) and time-varying parameter values, given by $(\theta_{2e}, \theta_{3e}, p_2, p_3)(t) = (-0.227, 0.559, 6070, 192)$ for $t \in [0, \pi)$, and equal to $(-0.559, 0.226, 226, 5240)$ for $t \in [\pi, 2\pi)$. Figure 3 shows the resulting open-loop stable limit cycle response, starting from the initial position $(40, 3, -3)$ degrees, with $\dot{\theta}_i(0) = 0$ for $i = 1, 2, 3$. Through perturbation analysis and the method of harmonic balance, the limit cycle is closely approximated by $\theta_1^{\text{lc}}(t) = (50\pi/180)\cos(t)$, $\theta_2^{\text{lc}}(t) = \theta_{2e} + (3\pi/180 - \theta_{2e})\cos(2t)$, and $\theta_3^{\text{lc}}(t) = \theta_{3e} + (3\pi/180 - \theta_{3e})\cos(2t)$. The chosen

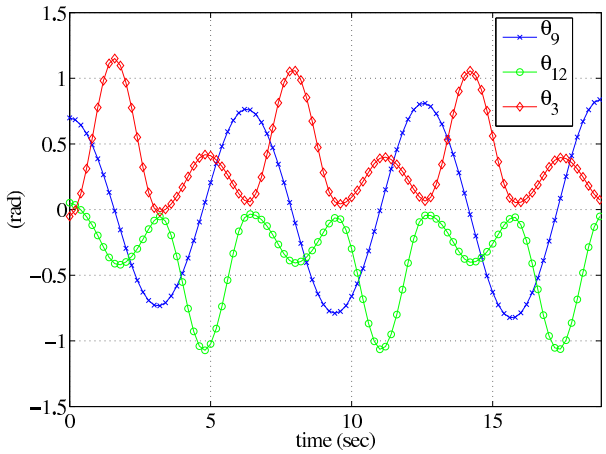


Figure 3: Open-loop stable limit cycle, showing the angular positions starting from $(40, 3, -3)$ degrees with zero initial angular velocity.

initial condition demonstrates the attractivity of the stable limit cycle. For example, note that the amplitude of $\theta_1(t)$ starts at 40 degrees and approaches 50 degrees, the amplitude of

$\theta_1^{\rm lc}(t)$. While the robot has 6 total degrees of freedom when walking in accordance with the limit cycle response above, the remaining degrees-of-freedom (including two ankles and one free foot) can be derived from the three primary degrees-of-freedom θ_1, θ_2 and θ_3 [8].

With zero control, there are two equilibrium conditions. One is the limit cycle defined above, and the other is the unstable fixed point $(\theta_1, \theta_2, \theta_3) = (\theta_{1e}, \theta_{2e}, \theta_{3e})$ with $\theta_{1e} = \dot{\theta}_i = 0$ for $i = 1, 2, 3$. A reasonable control objective is to command torque motors (controls u_i) to drive the three angles from the stable limit cycle response to the fixed point; that is, to stably bring the robot to a stop. To do so within one half-period of the limit cycle response means that one set of parameter values $(\theta_{2e}, \theta_{3e}, p_2, p_3)$ need to be considered in the model. As such, for control purposes, these parameters are assumed to take on the values $(-0.227, 0.559, 6070, 192)$. In this way, discontinuous dynamic equations are also avoided. Now, through a change of variables, the dynamics and input constraints satisfy the conditions of Assumption 1.

Denoting $z_i = (\theta_i - \theta_{ie}, \dot{\theta}_i)$, the dynamics are linearized around $(z_i, u_i) = (0, 0)$. For $i = 1$, the matrix A_{11} has unstable eigenvalues, and for $i = 2, 3$, the matrix A_{ii} is nearly neutrally stable with eigenvalues $-0.045 \pm 2j$. For all three oscillators, the dynamics are linearly controllable around the origin. In accordance with Assumption 2, the following gain matrices are used to stabilize the linearized dynamics: $K_1 = [3.6\ 5.3]$, $K_2 = K_3 = [2.0\ 4.8]$. The resulting closed-loop matrix $A_{\rm c}$ has eigenvalues $(-1.1, -4.1, -3, -2.3 \pm 0.5j, -2)$. For the cost function J_i, the chosen weights are $Q_i = {\rm diag}(20, 20)$ and $R_i = 0.1$, $i = 1, 2, 3$. Then, each P_i is calculated according to the Lyapunov equation (8). Since the maximum eigenvalue of $PA_{\rm o} + A_{\rm o}^T P - \widehat{Q}/2$ is -7.4, Assumption 3 is satisfied.

After Lemma 1, a semi-infinite optimization problem is defined, and the solution to this problem provides the terminal constraint parameter ε. Rather than solving this problem, $\varepsilon = 0.01$ is used in the numerical results presented here. With $\|z\|_P \leq 0.01$, the terminal controllers $u_i = K_i z_i$ all satisfy the input constraints $|K_i z_i| \leq 1$. In accordance with Assumption 4, a centralized optimal control problem is solved at initial time $t_0 = 0$. In this problem, the sum of the three cost functions $J_1 + J_2 + J_3$ is minimized, enforcing terminal state and input constraints with a horizon time of $T = 6$ seconds. The initial condition is kept the same as that shown in Figure 3.

To solve the centralized optimal control problem, and each of the distributed optimal control problems, the same approach is used. In the spirit of the Nonlinear Trajectory Generation package developed by Milam *et al.* [14], a collocation technique is employed within MATLAB. First, each angular position trajectory $\theta_i(t)$ is parameterized as a $C^2[t_k, t_k +$

T] 6-th order B-spline polynomial. The constraints and cost functions are evaluated at 121 breakpoints over each 6 second time window. The resulting nonlinear programming problem is solved using the `fmincon` function, generating the 27 B-spline coefficients for each position $\theta_i(t)$. Using the concept of differential flatness [16], the control inputs u_i are not parameterized as polynomials for which the coefficients must also be calculated. Instead, each control input is defined in terms of the parameterized positions $\theta_i(t)$ and their derivatives through the dynamics (see [16] for a detailed description of this procedure and when it is applicable). The solution to the centralized optimal control problem at initial time is shown in Figure 4. The position and control trajectories for this open-loop solution

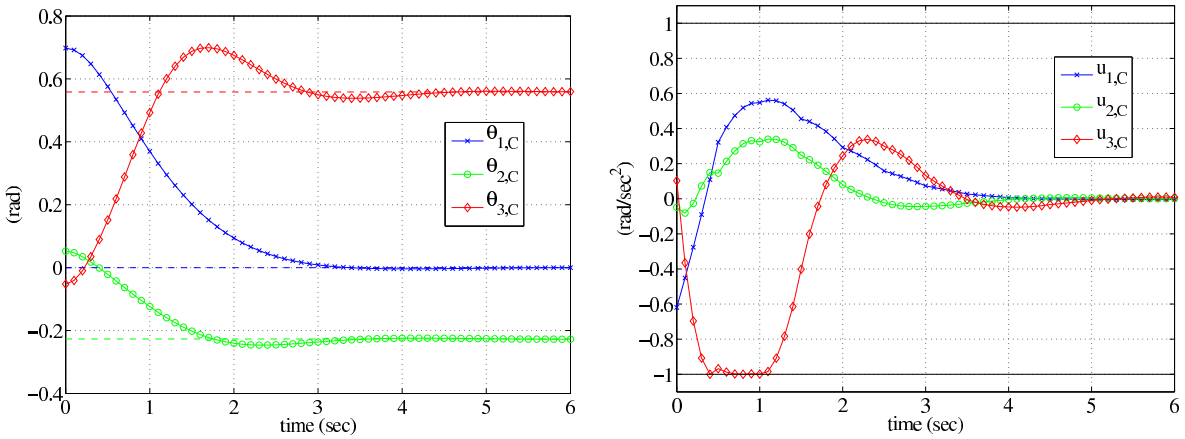


Figure 4: The open-loop response generated by solving a single centralized optimal control problem at initial time $t_0 = 0$. The left plot shows the open-loop angular position trajectories $\theta_{i,\mathrm{C}}$, and the right plot shows the open-loop control trajectories $u_{i,\mathrm{C}}$.

are denoted $\theta_{i,\mathrm{C}}$ and $u_{i,\mathrm{C}}$, respectively, where the subscript C denotes "centralized." Note that the positions are brought suitably close to their fixed point values (shown by dashed lines) within a half-period of π seconds, validating the assumption that the model parameters $(\theta_{2e}, \theta_{3e}, p_2, p_3)$ are constant over the time horizon of 6 seconds.

With an initially feasible solution available, the distributed RHC algorithm can be employed. Before presenting the results, the theoretical conditions are evaluated. In total, the equations that must be satisfied are (12)–(14) and (17). First, in accordance with Assumption 5, the Lipschitz parameters for F need to be identified. Through simulation and application of the triangle inequality, the oscillator dynamics satisfy $\|\widehat{F}(z, z', u)\|_P \leq 4\|z\|_P + 0.1\|z'\|_P + 1\|u\|$. Time scaling is introduced to normalize the horizon time from 6 seconds to 1 second. For the dynamics $\widehat{F}$, let $\tau(t) = t/T \in [0, 1]$ such that $\frac{d}{d\tau} z(\tau) = T\widehat{F}(z(\tau), z'(\tau), u(\tau))$ for all $\tau \in [0, 1]$. Now, the scaled dynamics satisfy $\|T\widehat{F}(z, z', u)\|_P \leq 4T\|z\|_P + 0.1T\|z'\|_P + T\|u\|$.

To get into the normalized form stated in Assumption 5, the dynamics are again scaled by the Lipschitz constant 4, defining $F = \widehat{F}/(4T)$. Then, the normalized Lipschitz bounds become $\|F(z, z', u)\|_P \leq \|z\|_P + \beta\|z'\|_P + \gamma\|u\|$, where $\beta = 0.1/4 = 0.025$ and $\gamma = 1/4 = 0.25$. Choosing the design parameter $q = 60$, the lower bound on δ from (12) is $\delta \geq 0.0397$ seconds. To satisfy this constraint, the update period (for the time-scaled dynamics) is chosen to be $\delta = 0.04$ seconds.

To satisfy the conditions of Lemma 3 (equations (13) and (14)), the parameter c is redefined by requiring that $\|\phi(z)\|_P/\|z\|_P \leq 1/(40\lambda_{\max}(\widehat{Q}^{-1/2}P\widehat{Q}^{-1/2}))$ when $\|z\|_P \leq \varepsilon = 0.01$. Again, one would have to test if the function $\psi_1(z)$ (defined after Lemma 1) remains non-positive in the neighborhood of the origin Ω_ε, to guarantee that ϕ satisfies this bound. With this new value for c, with $\delta = 0.04$ and choosing the design parameter $r = 8$, the left hand side of (13) is 0.985 and the left hand side of (14) is 0.894. Since both numbers are less than one, both conditions (13) and (14) are satisfied. Lastly, equation (17) is a sufficient condition for stability, and it is satisfied for the values $T = 1$, $\delta = 0.04$ and $q = 60$. Therefore, the conditions of the theory guaranteeing feasibility and stability of the distributed RHC algorithm are satisfied. Scaling time back to a planning horizon of 6 seconds corresponds to an update period of 0.24 seconds. In the distributed RHC results below, an update period of 0.3 seconds is used.

While the theory requires constraints (10) and (11) in each distributed optimal control problem (Problem 1), a relaxed version of the problem is employed in the following numerical results. Specifically, both constraints (10) and (11) are removed from each distributed problem. As described in Algorithm 1, prior to each update, agent 1 computes $\hat{z}_1$ and transmits it to agents 2 and 3. Likewise, agents $i = 2$ and 3 compute $\hat{z}_i$ and transmit it to one another, in accordance with the coupling in the oscillator dynamics. The closed-loop position and control responses generated by applying the distributed RHC algorithm with $(\delta, T) = (0.3, 6)$ seconds are shown in Figure 5. The position and control trajectories for this closed-loop solution are denoted $\theta_{i,\mathrm{D}}$ and $u_{i,\mathrm{D}}$, respectively, where the subscript D denotes "distributed." While the algorithm and theory suggest switching to the terminal controllers once $z(t) \in \Omega_\varepsilon$, the distributed receding horizon controllers are employed for all time in these results. At each RH update, each distributed optimal control problem is initialized by assuming $\theta_i(t; t_k) = \theta_i(t_k)$ for all $t \in [t_k, t_k + T]$, rather than computing $\bar{z}_i(t; t_k)$ and warm starting the optimization. Since $\bar{z}_i(t; t_k)$ is not used as a warm start, and (10) and (11) are removed from each optimal control problem, there is no need to compute $\bar{z}_i(t; t_k)$ and step 2(a) in Algorithm 1 is removed.

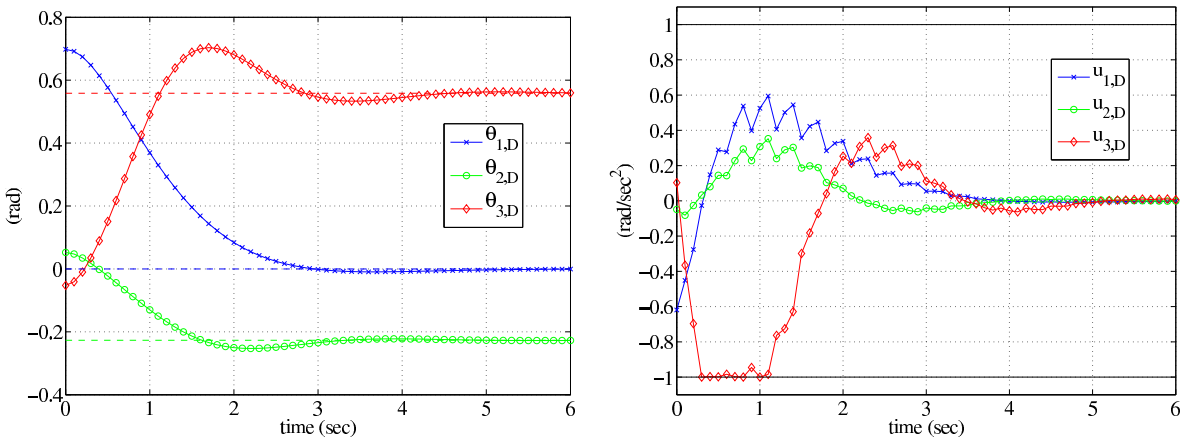


Figure 5: The closed-loop response generated by applying the distributed RHC algorithm, with update period and horizon time $(\delta, T) = (0.3, 6)$ seconds. The left plot shows the closed-loop angular position trajectories $\theta_{i,\mathrm{D}}$, and the right plot shows the closed-loop control trajectories $u_{i,\mathrm{D}}$.

While the open-loop and closed-loop position responses ($\theta_{i,\mathrm{C}}$ and $\theta_{i,\mathrm{D}}$, repsectively) are hard to discern, the closed-loop control response $u_{i,\mathrm{D}}$ is clearly different from the open-loop response $u_{i,\mathrm{C}}$. Over the first 3 seconds, the corrections in $u_{i,\mathrm{D}}$ at update times are due (at least in part) to the inherent error caused by each agent relying on the assumption that neighboring agents are continuing along their previously predicted trajectory ($\hat{z}_i$). After 3 seconds, the closed-loop control response shows less corrective action, as each assumed $\hat{z}_i(\cdot; t_k)$ approaches the true $z_i^p(\cdot; t_k)$ since the closed-loop states are converging toward their desired fixed point values. The more explicitly show the difference between the centralized open-loop and distributed RHC closed-loop responses, Figure 6 shows a plot of the angular position deviation $\Delta\theta_i = \theta_{i,\mathrm{C}} - \theta_{i,\mathrm{D}}$ and the control deviation $\Delta u_i = u_{i,\mathrm{C}} - u_{i,\mathrm{D}}$, for each $i = 1, 2, 3$.

To compare the computational burden of the centralized problem and the distributed problems, the `cputime` function is used in MATLAB. The centralized optimal control problem has 81 variables to solve for, and the solution shown in Figure 4 took 328 seconds to generate. Each distributed optimal control problem has 27 variables to solve for, where each problem is solved in parallel and the average run time per receding horizon update is shown in Figure 7. From the figure, the average computation time per agent is 8.1 seconds at update time t_1. As time proceeds, the problems take less time to solve, as the system is approaching the fixed point solution. Clearly, for this example there is substantial savings in being able to solve the distributed problems in parallel.

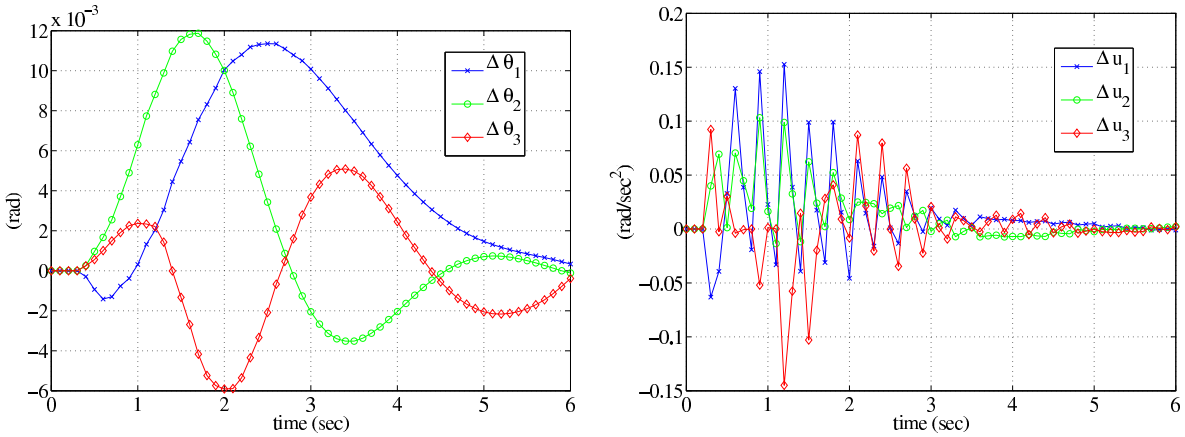


Figure 6: Deviation between centralized open-loop trajectories shown in Figure 4 and the closed-loop distributed RHC trajectories shown in Figure 5. The left plot shows the angular position deviation ($\Delta\theta_i = \theta_{i,\mathrm{C}} - \theta_{i,\mathrm{D}}$), and the right plot shows the control deviation ($\Delta u_i = u_{i,\mathrm{C}} - u_{i,\mathrm{D}}$).

Figure 7: Average run time versus receding horizon update number, for the three distributed optimal control problems solved in parallel.

6 Conclusions

In this paper, a distributed implementation of receding horizon control is developed for the case of dynamically coupled nonlinear systems subject to decoupled input constraints. A central element to the feasibility and stability analysis is that the actual and assumed responses of each agent are not too far from one another, as quantified by a consistency constraint. Parametric bounds on the receding horizon update period are identified. Also, conditions that bound the amount of dynamic coupling, parameterized by a Lipschitz constant, are also identified. While the theoretical results are conservative, relaxations of the proposed algorithm are shown to be applicable in a problem of distributed control of coupled nonlinear oscillators. In the numerical results, the time it takes to solve the (relaxed) distributed optimal control problems in parallel is *two orders of magnitude less* than the time it takes to solve a corresponding centralized optimal control problem, underlining the computational savings incurred by employing the distributed algorithm. Moreover, the closed-loop response generated by the distributed algorithm is quite close to an open-loop response generated by solving a single centralized optimal control problem. In addition to the oscillator example considered here, relaxations of the theory have been employed in the venue of supply chain management [6]. A theory more specific to the supply chain management case (coupled discrete-time dynamics with time delays) is currently under development.

References

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

[2] M. W. Braun, D. E. Rivera, W. M. Carlyle, and K. G. Kempf. Application of model predictive control to robust management of multiechelon demand networks in semiconductor manufacturing. *Simulation*, 79(3):139–156, 2003.

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

[4] S. Chopra and P. Meindl. *Supply Chain Management, Strategy, Planning, and Operations - Second Edition.* Prentice-Hall, 2004.

[5] J. P. Corfmat and A. S. Morse. Decentralized control of linear multivariable systems. *Automatica*, 12:479–495, 1976.

[6] W. B. Dunbar and S. Desa. Distributed nonlinear model predictive control for dynamic supply chain management. In *Proceedings of the International Workshop on Assessment and Future Directions of NMPC*, Freudenstadt-Lauterbad, Germany, August, 2005.

[7] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, April, 2006.

[8] M. S. Dutra, A. C. de Pina Filho, and V. F. Romano. Modeling of a bipedal locomotor using coupled nonlinear oscillators of Van der Pol. *Biol. Cybern.*, 88:286–292, 2003.

[9] D. Jia and B. H. Krogh. Distributed model predictive control. In *Proceedings of the IEEE American Control Conference*, 2001.

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

[11] H. K. Khalil. *Nonlinear Systems, Second Edition*. Prentice Hall, 1996.

[12] 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.

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

[14] M. B. Milam, K. Mushambi, and R. M. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the Conference on Decision and Control*, 2000.

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

[16] N. Petit, M. B. Milam, and R. M. Murray. Inversion based trajectory optimization. In *IFAC Symposium on Nonlinear Control Systems Design*, 2001.

[17] N. R. Sandell, P. Varaiya, M. Athans, and M. G. Safanov. Survey of decentralized control methods of large scale systems. *IEEE Trans. Auto. Contr.*, 23(2):108–128, 1978.

[18] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In *Proc. of the IEEE Conference on Decision and Control / IEE European Control Conference*, Seville, Spain, 2005.

[19] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Plant-wide optimal control with decentralized MPC. In *Proceedings of the IFAC Dynamics and Control of Process Systems Conference*, Boston, MA, July, 2004.

[20] G-Y. Zhu and M. A. Henson. Model predicitive control of interconnected linear anad nonlinear processes. *Ind. Eng. Chem. Res.*, 41:801–816, 2002.