

Distributed Receding Horizon Control of Coupled Nonlinear Oscillators: Theory and Application

William B. Dunbar

Abstract—This paper extends recent results on distributed receding horizon control (RHC) of dynamically coupled nonlinear systems subject to decoupled input constraints. Motivating examples of such systems include chains of coupled oscillators and supply chain management systems. Conditions for feasibility and stability of the distributed RHC algorithm are stated, with substantially less conservative requirements than previously derived. The conditions are shown to be satisfied for a set of coupled Van der Pol oscillators that model a walking robot experiment. Numerical experiments show good performance and demonstrate the computational savings over centralized RHC.

Keywords: receding horizon control, model predictive control, distributed control, decentralized control, large scale systems.

I. 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 [18] and supply chain management systems [2], [7]. 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. In this paper, subsystems that are dynamically coupled are referred to as *neighbors*. The work presented is an extension of a recent work [5]. As in [5], 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 employ decentralized decision making. In comparison to [5], the theoretical conditions which guarantee feasibility and stability are much

less conservative, and shown to be satisfied in the case of a set of coupled nonlinear oscillators.

Previous work on distributed RHC of dynamically coupled systems include Jia and Krogh [10] and Acar [1]. These papers address coupled linear time-invariant subsystem dynamics with quadratic separable cost functions. State and input constraints are not included, aside from a stability constraint in [10] that permits state information exchanged between the subsystems to be delayed by one update period. In another work, Jia and Krogh [11] 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.* [16], [17] 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 II begins by defining the nonlinear coupled subsystem dynamics and control objective. In Section III, 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 IV. 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 V, 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]. Finally, Section VI provides conclusions. To keep the paper conference length, all proofs are omitted and can be found in [6].

II. SYSTEM DESCRIPTION AND OBJECTIVE

In this section, the system dynamics and control objective are defined. 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

W. B. Dunbar is an Assistant Professor of Computer Engineering, University of California, Santa Cruz, 95064, USA. dunbar@soe.ucsc.edu.

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 $t \in \mathbb{R}$. The 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 $i \in \{1, \dots, N_a\}$, 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 $z = (z_1, \dots, z_{N_a})$ and $u = (u_1, \dots, u_{N_a})$.

The dynamic coupling between the agents is topologically identified by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N_a\}$ 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 $j \in \mathcal{V}$, it is said that j is an *upstream neighbor* of agent i , and $\mathcal{N}_i^u \subseteq \mathcal{V}$ denotes the set of upstream neighbors of any agent $i \in \mathcal{V}$. The set of all directed edges is defined as $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid j \in \mathcal{N}_i^u, \forall i \in \mathcal{V}\}$. For every $i \in \mathcal{V}$, it is assumed that z_i appears in the dynamic equation for i , and so $i \in \mathcal{N}_i^u$ 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 $j \in \mathcal{N}_i^u$ does not necessarily imply $i \in \mathcal{N}_j^u$.

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^d . The set of all directed edges can be equivalently defined as $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid i \in \mathcal{N}_j^d, \forall i, j \in \mathcal{V}\}$. Note that $j \in \mathcal{N}_i^u$ if and only if $i \in \mathcal{N}_j^d$, for any $i, j \in \mathcal{V}$. It is assumed in this paper that the graph \mathcal{G} is connected. Consequently, for every $i \in \mathcal{V}$, the set $(\mathcal{N}_i^d \cup \mathcal{N}_i^u) \setminus \{i\} \neq \emptyset$, 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 system dynamics for each agent $i \in \mathcal{V}$ are given by

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

where $z_{-i} = (z_{j_1}, \dots, z_{j_l})$, $l = |\mathcal{N}_i^u|$, denotes the concatenated vector of the states of the upstream neighbors of i . 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 $\mathcal{U} \times \dots \times \mathcal{U}$. In concatenated vector form, the system dynamics are

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

given $z(t_0)$, and $f = (f_1, \dots, f_{N_a})$.

Assumption 1: The following holds: **(a)** f is C^2 and $0 = f(0, 0)$; **(b)** system (2) has a unique solution for any $z(t_0)$ and any piecewise right-continuous control $u : [t_0, \infty) \rightarrow \mathcal{U}^{N_a}$; **(c)** $\mathcal{U} \subset \mathbb{R}^m$ is compact, containing the origin in its interior. \square

Consider now the linearization of (1) around the origin, denoting $A_{il} = \partial f_i / \partial z_l(0, 0)$ and $B_i = \partial f_i / \partial u_i(0, 0)$. As in many RHC formulations, a feedback controller for which the closed-loop system is asymptotically stabilized inside a neighborhood of the origin will be utilized.

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 $\dot{z} = A_c z$ is asymptotically stable, where $A_c \triangleq [f_z(0, 0) + f_u(0, 0)K]$ and $K = \text{diag}(K_1, \dots, K_{N_a})$. \square

The decoupled linear feedbacks above are referred to as *terminal controllers*. Associated with the closed-loop linearization, denote the block-diagonal Hurwitz matrix $A_d = \text{diag}(A_{d1}, \dots, A_{dN_a})$ and the off-diagonal matrix $A_o = A_c - A_d$. Assumption 2 inherently presumes decoupled stabilizability and that the coupling between subsystems in the linearization is sufficiently weak, as discussed and quantified in the survey paper [15]. The terminal controllers will be employed in a prescribed neighborhood of the origin.

III. DISTRIBUTED RECEDING HORIZON CONTROL

In this section, N_a separate optimal control problems are defined and the distributed RHC algorithm. 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, additional conditions on δ are required, as quantified in the next section. Denote the update time $t_k = t_0 + \delta k$, where $k \in \mathbb{N} = \{0, 1, 2, \dots\}$. 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^{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 on 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 $j \in \mathcal{N}_i^u$. Likewise, agent i transmits an assumed state trajectory \hat{z}_i to every downstream neighbor $j \in \mathcal{N}_i^d$, 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 the predicted control. This discrepancy is identified by using the following notation. Recall that $z_i(t)$ and $u_i(t)$ are the actual state and control, respectively, for each agent $i \in \mathcal{V}$ at any time $t \geq t_0$. Associated with update time t_k , 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 $\hat{z}_{-i}(\cdot; t_k)$ be the assumed open-loop state trajectories of the upstream neighbors of i . For any agent $i \in \mathcal{V}$, then, the predicted state trajectory satisfies

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

for all $t \in [t_k, t_k + T]$, 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

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

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

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

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 trajectory 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 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

$$\dot{\bar{z}}_i(t; t_k) = f_i(\bar{z}_i(t; t_k), \hat{z}_{-i}(t; t_k), \bar{u}_i(t; t_k)), \quad (6)$$

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

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

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. In each local optimal control problem, 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

$$\int_{t_k}^{t_k+T} \|z_i^p(s; t_k)\|_{Q_i}^2 + \|u_i^p(s; t_k)\|_{R_i}^2 ds + \|z_i^p(t_k + T; t_k)\|_{P_i}^2,$$

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 $P_i A_{di} + A_{di}^T P_i = -\hat{Q}_i$, where $\hat{Q}_i = Q_i + K_i^T R_i K_i$. Denoting $P = \text{diag}(P_1, \dots, P_{N_a})$ and $\hat{Q} = \text{diag}(\hat{Q}_1, \dots, \hat{Q}_{N_a})$, it follows that $PA_d + A_d^T P = -\hat{Q}$ and $\hat{Q} > 0$. Decoupled terminal state constraints will be included in each local optimal control problem. 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: $PA_o + A_o^T P \leq \hat{Q}/2$. \square

Lemma 1: Suppose that Assumptions 1–3 hold. There exists a positive constant $\varepsilon \in (0, \infty)$ such that the set $\Omega_\varepsilon \triangleq \{z \in \mathbb{R}^{n_{N_a}} \mid \|z\|_P \leq \varepsilon\}$, is a positively invariant region of attraction for both the closed-loop linearization $\dot{z}(t) = A_c z(t)$ and the closed-loop nonlinear system $\dot{z}(t) = f(z(t), Kz(t))$. Additionally, $Kz \in \mathcal{U}^{N_a}$ for all $z \in \Omega_\varepsilon$. \blacksquare The parameter $\varepsilon \in (0, \infty)$ that satisfied the conditions of the lemma can be found numerically, as described in [6], [13]. In each local optimal control problem, the terminal state constraint set for each $i \in \mathcal{V}$ is

$$\Omega_i(\varepsilon) \triangleq \left\{ z_i \in \mathbb{R}^n \mid \|z_i\|_{P_i} \leq \varepsilon / \sqrt{N_a} \right\}. \quad (8)$$

By construction, if $z \in \Omega_1(\varepsilon) \times \dots \times \Omega_{N_a}(\varepsilon)$, then the decoupled controllers can stabilize the system to the origin, since $\|z_i\|_{P_i}^2 \leq \frac{\varepsilon^2}{N_a}$, $\forall i \in \mathcal{V}$ implies $\sum_{i \in \mathcal{V}} \|z_i\|_{P_i}^2 \leq \varepsilon^2$. 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. 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 $q \in \{1, 2, 3, \dots\}$ be any positive integer. For each agent $i \in \mathcal{V}$ and at any update time t_k , $k \geq 1$:

Given: $z_i(t_k)$, $\bar{z}_{-i}(t; t_k)$ and $\hat{z}_{-i}(t; t_k)$, $t \in [t_k, t_k + T]$;
Find: the control trajectory $u_i^p(\cdot; t_k) : [t_k, t_k + T] \rightarrow \mathcal{U}$ that minimizes $J_i(z_i(t_k), u_i^p(\cdot; t_k))$, subject to the constraints

$$\|z_i^p(t; t_k)\|_{P_i}^2 \leq \|\bar{z}_i(t; t_k)\|_{P_i}^2 + a, \quad (9)$$

$$\|z_i^p(t; t_k) - \hat{z}_i(t; t_k)\|_{P_i}^2 \leq \|\bar{z}_i(t; t_k) - \hat{z}_i(t; t_k)\|_{P_i}^2 + b, \quad (10)$$

for all $t \in [t_k, t_k + T]$, where $a = [\delta\varepsilon/(8T\sqrt{N_a})]^2$, $b = [\varepsilon/(2(q+1)N_a)]^2$, $z_i^p(\cdot; t_k)$ 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 (8). \square

Equation (9) is utilized to prove that the distributed RHC algorithm is stabilizing. Equation (10) 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 (10) is a key equation in proving that \bar{z}_i is a feasible state trajectory at each update. The constant $q \in \{1, 2, 3, \dots\}$ is a design parameter, and the choice for q will be motivated in Section IV. Before stating the distributed RHC algorithm, an assumption (standard in centralized implementations) 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 $u_i^p(\cdot; t_0)$. \square

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 *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, $T \in (0, \infty)$, $\delta \in (0, T]$, $q \in \{1, 2, 3, \dots\}$.

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) and transmit it to every downstream neighbor $l \in \mathcal{N}_i^d$.
- c) Receive $\hat{z}_j(\cdot; t_{k+1})$ from every upstream neighbor $j \in \mathcal{N}_i^u$ and assemble $\hat{z}_{-i}(\cdot; t_{k+1})$.
- 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) Measure $z_i(t_{k+1})$.
- b) Compute $\bar{z}_i(\cdot; t_{k+1})$ according to (6).
- c) Solve Problem 1, yielding $u_i^p(\cdot; t_{k+1})$. \square

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 states the theoretical results showing that the distributed RHC algorithm is feasible at every update and stabilizing.

IV. THEORETICAL RESULTS

In this section, the feasibility and stability results are states, with all proofs provided in [6]. 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 $k \geq 1$. If Assumption 4 holds true, the first result of this section is that 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 $z_i^p(\cdot; t_k) = \bar{z}_i(\cdot; t_k)$. The feasibility result requires 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 $\dot{z}^p(t; t_k) = F(z^p(t; t_k), \hat{z}(t; t_k), u^p(t; t_k))$, $t \in [t_k, t_k + T]$, where $F: \mathbb{R}^{nN_a} \times \mathbb{R}^{nN_a} \times \mathbb{R}^{mN_a} \rightarrow \mathbb{R}^{nN_a}$, $z^p = (z_1^p, \dots, z_{N_a}^p)$, $\hat{z} = (\hat{z}_1, \dots, \hat{z}_{N_a})$ and $u^p = (u_1^p, \dots, u_{N_a}^p)$. 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

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

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

More generally, the Lipschitz bound would take the form $\|\hat{F}(z, z', u)\|_P \leq \hat{\alpha} \|z\|_P + \hat{\beta} \|z'\|_P + \hat{\gamma} \|u\|$ 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 feasibility result is now stated.

Theorem 1: Suppose that Assumptions 1–5 hold, $z(t_0) \in Z$ and the following parametric conditions hold:

$$\frac{e^{\delta(1+\gamma_K)} \ln \left[\frac{(q+1)^2/q^2}{\min_{i \in \mathcal{V}} \{\lambda_{\min}(P_i^{-1/2} \hat{Q}_i P_i^{-1/2})\}} \right]}{\leq \frac{r/(r+1)}{c(q+1)\sqrt{N_a}}}, \quad (11)$$

$$\beta 2T(r+1) \exp [T + \delta(1 + \beta + \gamma_K)] \leq 1, \quad (12)$$

where constants c and γ_K are defines as $c = 1/(10\lambda_{\max}(\hat{Q}^{-1/2} P \hat{Q}^{-1/2})) + \lambda_{\max}^{1/2}(P^{-1/2} A_0^T P A_0 P^{-1/2})$, $\gamma_K = \gamma \lambda_{\max}^{1/2}(P^{-1/2} K^T K P^{-1/2})$, and constants $q, r \in \{1, 2, 3, \dots\}$ are chosen design parameters. 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 $k \geq 1$. \blacksquare

The purpose of the design parameters $q, r \in \{1, 2, 3, \dots\}$ is to shift the bounds on δ and β , as necessary to find feasible values for a specific problem. The larger the chosen value of q , the smaller the lower and upper bounds on δ . for example. The ability to shift the feasible range for δ is useful for design purposes, as will be demonstrated in Section V. Also, larger values of q reduce the margin in the consistency constraint (10) that bounds how much the predicted state

can deviate from the assumed state. Equation (12) places an upper bound on the Lipschitz coupling constant β . 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 is also demonstrated in Section V. Parametric conditions which ensure the stability of the closed-loop system (5) are now stated.

Theorem 2: Suppose that Assumptions 1–5 hold, $z(t_0) \in Z$, conditions (11)–(12) are satisfied, and the following parametric conditions hold

$$T \geq 8\delta, \quad (q+1) \geq 2 \frac{T-\delta}{\delta}. \quad (13)$$

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

The feasibility and stability results in this paper are 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.

V. 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 experiment [8]. In the following, $\theta_1 \in [-\pi/2, \pi/2]$ is the relative angle between the two thighs, $\theta_2 \in [-\pi/2, \pi/2]$ is the right knee angle (relative to the right thigh), and $\theta_3 \in [-\pi/2, \pi/2]$ is the left knee angle (relative to left thigh). The controlled equations of motion in units of (rad/sec)² are

$$\begin{aligned} \ddot{\theta}_1(t) &= 0.1 [1 - 5.25\theta_1^2(t)] \dot{\theta}_1(t) - \theta_1(t) + u_1(t) \\ \ddot{\theta}_2(t) &= 0.01 [1 - p_2(\theta_2(t) - \theta_{2e})^2] \dot{\theta}_2(t) - 4(\theta_2(t) - \theta_{2e}) \\ &\quad + 0.057\theta_1(t)\dot{\theta}_1(t) + 0.1(\dot{\theta}_2(t) - \dot{\theta}_3(t)) + u_2(t) \\ \ddot{\theta}_3(t) &= 0.01 [1 - p_3(\theta_3(t) - \theta_{3e})^2] \dot{\theta}_3(t) - 4(\theta_3(t) - \theta_{3e}) \\ &\quad + 0.057\theta_1(t)\dot{\theta}_1(t) + 0.1(\dot{\theta}_3(t) - \dot{\theta}_2(t)) + u_3(t), \end{aligned}$$

subject to $|u_i(t)| \leq 1, \forall t \geq 0, i = 1, 2, 3$.

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 1 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 $\theta_1^c(t) = (50\pi/180) \cos(t)$, $\theta_2^c(t) = \theta_{2e} + (3\pi/180 - \theta_{2e}) \cos(2t)$, and $\theta_3^c(t) = \theta_{3e} + (3\pi/180 - \theta_{3e}) \cos(2t)$. The chosen initial condition demonstrates the attractivity of the stable

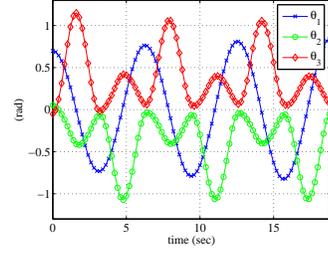


Fig. 1. Open-loop stable limit cycle, showing the angular positions starting from (40, 3, -3) degrees with zero initial angular velocity.

limit cycle. For example, note that the amplitude of $\theta_1(t)$ starts at 0.70 radians and approaches 0.87 radians, the amplitude of $\theta_1^c(t)$. 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 $(\theta_{2e}, \theta_{3e}, p_2, p_3)$ can 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 Assump. 1.

Denoting $z_i = (\theta_i - \theta_{ie}, \dot{\theta}_i)$, the dynamics are linearized around $(z_i, u_i) = (0, 0)$. The matrix A_{11} has unstable eigenvalues $0.05 \pm j$, and the matrices A_{22} and A_{33} are unstable with eigenvalues $0.055 \pm 2j$. 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: $K_1 = [3.6 \ 5.3]$, $K_2 = K_3 = [2.0 \ 5.0]$. The resulting closed-loop matrix A_c has eigenvalues $(-1.1, -4.1, -3, -2.4 \pm 0.5j, -2)$. For the cost function J_i , the chosen weights are $Q_i = \text{diag}(30, 30)$ and $R_i = 0.1, i = 1, 2, 3$. Then, each P_i is calculated according to the Lyapunov equation on page 3. Since the maximum eigenvalue of $PA_0 + A_0^T P - Q/2$ is -11 , Assumption 3 is satisfied. The constraint parameter $\varepsilon = 0.2$ satisfies the conditions of Lemma 1, with the calculated for ε described in [6]. In accordance with Assumption 4, a centralized optimal control problem is solved at initial time $t_0 = 0$. In this problem, and in the centralized RHC implementation, 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 1.

To solve the centralized optimal control problem, and

each of the distributed optimal control problems, the same approach is used. The computer with MATLAB 7.0 software has a 2.4 GHz Intel Pentium(R) 4 CPU, with 512 MB of RAM. 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, 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. With an update period of $\delta = 0.15$ seconds, the centralized RHC state and control response is shown in Figure 2. The position and control trajectories are

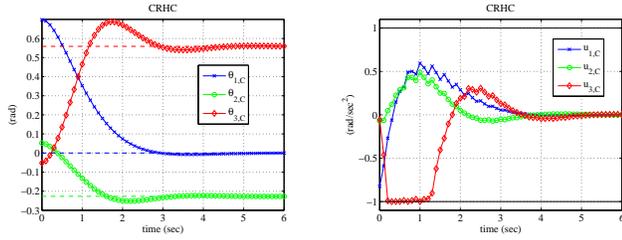


Fig. 2. The centralized RHC response, showing the angular position trajectories $\theta_{i,C}$ (left plot) and the control trajectories $u_{i,C}$ (right plot), for each $i = 1, 2, 3$.

denoted $\theta_{i,C}$ and $u_{i,C}$, respectively. 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 parametric equations that must be satisfied are (11)–(13). In accordance with Assumption 5, the Lipschitz parameters are identified as $\|\hat{F}(z, z', u)\|_P \leq 4\|z\|_P + 0.1\|z'\|_P + 1\|u\|$. To facilitate calculation of an update period δ that satisfies the parametric conditions, time scaling is introduced to normalize the horizon time from $T = 6$ seconds to 1 second. For the dynamics \hat{F} , let $\tau(t) = t/T \in [0, 1]$ such that $\frac{d}{d\tau}z(\tau) = T\hat{F}(z(\tau), z'(\tau), u(\tau))$ for all $\tau \in [0, 1]$. Now, the scaled dynamics satisfy $\|T\hat{F}(z, z', u)\|_P \leq 4T\|z\|_P + 0.1T\|z'\|_P + T\|u\|$. To get into the normalized form, the dynamics are scaled as $F = \hat{F}/(4T)$. 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 parameters $q = 90$ and $r = 6$, the lower bound in (11) is 0.025 and the upper bound is 0.028. So, the update period (for the time-scaled dynamics) is chosen to be $\delta = 0.025$ seconds. Also, the left hand side of (12) is 0.997, so the inequality is satisfied. Lastly, equation (13) is a sufficient

condition for stability, and it is satisfied for the values $T = 1$, $\delta = 0.025$ and $q = 90$. Therefore, the parametric 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 $\delta = 0.15$ seconds, and this is the update period used in the centralized and distributed RHC implementations.

Distributed RHC is implemented precisely according to Algorithm 1, with one modification to Problem 1. In the optimization code, the constants on the right-hand side of constraints (9) and (10) are set to $a = b = 0.1$. The actual constants in (9) and (10) are small enough ($\sim 10^{-7}$) to cause feasibility problems in each distributed optimization code. The value of 0.1, on the other hand, worked quite well. Of course, the constants defined in constraints (9) and (10) are derived based on the sufficient conditions of the theory, and are likely to be conservative. The closed-loop position and control trajectories generated by applying the distributed RHC algorithm are shown in Figure 3. The

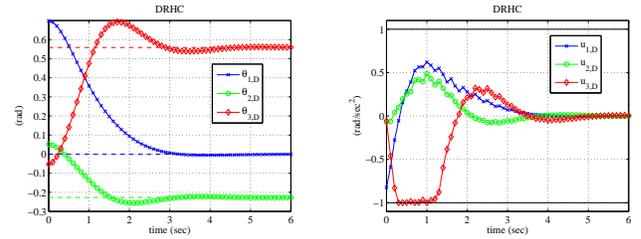


Fig. 3. The distributed RHC response, showing the angular position trajectories $\theta_{i,D}$ (left plot) and the control trajectories $u_{i,D}$ (right plot), for each $i = 1, 2, 3$. The response is quite close to the centralized RHC response shown in Figure 2.

position and control trajectories for this closed-loop solution are denoted $\theta_{i,D}$ and $u_{i,D}$, respectively. 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. To compute the actual closed-loop response between RHC updates requires numerical integration of the dynamic equations (see (5)). Also, to calculate each \tilde{z}_i , as required in part 2(b) of Algorithm 1, requires numerical integration of equation (6). In all cases, numerical integration was performed using the `ode23` function in MATLAB.

The centralized and distributed RHC responses are quite close, with the distributed position responses showing slightly more overshoot than the centralized counterparts, particularly for angles θ_2 and θ_3 . The closeness in the two responses can be attributed in part to the weak coupling in the dynamics as quantified by the coefficient $\beta = 0.025$. For weakly coupled dynamics, the error introduced by relying on \tilde{z}_j for neighbors has less of an impact on the closed-loop response, than for systems with dynamics that are strongly influenced by neighboring responses. Application of the theory to systems with stronger dynamic coupling would be useful in identifying difference between centralized RHC and the distributed RHC algorithm presented here. A hypothesis worth testing is that, even in the stronger coupling case,

if the update period δ is sufficiently small, the distributed RHC response is likely to be close to the centralized RHC response. The intuition behind this hypothesis is that the error introduced by relying on \hat{z}_j for neighbors is likely smaller for smaller update periods.

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 at each RHC update. The computational time for each RHC update, corresponding to the response shown in Figure 2, is shown in the top plot in Figure 4. Each distributed optimal control problem has 27 variables to solve for, where each problem is solved in parallel. The computational time for each RHC update per agent, corresponding to the responses shown in Figure 3, is shown in the bottom plot in Figure 4. From

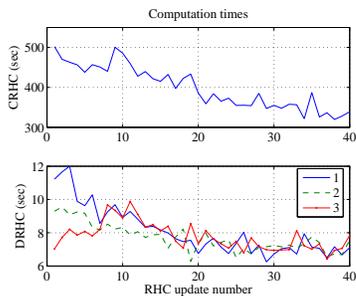


Fig. 4. Comparison of computation times, at each receding horizon update, to solve the centralized optimal control problem (top plot), and the distributed optimal control problems in parallel (bottom plot). The computation times correspond to the responses shown in Figures 2, 3.

the figure, the distributed optimal control problems were solved between 43 and 58 times faster than the centralized optimal control problem, over all updates. On average, each distributed problem was solved *50 times faster*, than the single centralized problem. Clearly, for this example, there is substantial savings in being able to solve the distributed problems in parallel. The savings are also consistent with the computational complexity comparison reported in [6]

VI. CONCLUSIONS

In this paper, a distributed implementation of RHC 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 sufficient, the proposed algorithm with minor relaxations is shown to be applicable to the problem of distributed control of coupled nonlinear oscillators. In the numerical results, the time it takes to solve the 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. The closed-loop response generated by the distributed algorithm is also quite close to a centralized RHC implementation. While it makes sense to compare centralized RHC with the distributed implementation for the academic coupled oscillator example considered here, centralized RHC is often not a viable option in applications [9] (for example, in supply chain management) where the distributed RHC algorithm may prove relevant.

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] W. B. Dunbar. A distributed receding horizon control algorithm for dynamically coupled nonlinear systems. In *Proc. of the IEEE Conference on Decision and Control / IEE European Control Conference*, Seville, Spain, 2005.
- [6] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. Technical Report ucsc-crl-06-03, Computer Engineering, University of California, Santa Cruz, 2006. Accepted to IEEE Trans. Aut. Contr., July, 2006.
- [7] 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.
- [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] Yu-Chi Ho. On centralized optimal control. *IEEE Trans. Auto. Contr.*, 50(4):537–538, 2005.
- [10] D. Jia and B. H. Krogh. Distributed model predictive control. In *Proceedings of the IEEE American Control Conference*, 2001.
- [11] 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.
- [12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- [13] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained 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. 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.
- [16] 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.
- [17] 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.
- [18] G-Y. Zhu and M. A. Henson. Model predictive control of interconnected linear and nonlinear processes. *Ind. Eng. Chem. Res.*, 41:801–816, 2002.